QR07

21st International Workshop on Qualitative Reasoning

June 26-28, 2007
Aberystwyth University, Wales, U.K.

Edited by Chris Price

Program Committee

Co-chairs QR07

Chris Price	Aberystwyth University, UK
Neal Snooke	Aberystwyth University, UK

Committee Members

Nuria Agell	ESADE, Spain
Chris Bailey-Kellogg	Dartmouth College, USA
Liz Bradley	Colorado University, USA
Ivan Bratko	University of Ljubljana, Slovenia
Bert Bredeweg	University of Amsterdam, Holland
George Coghill	University of Aberdeen, UK
Philippe Dague	CNRS-Université Paris-Sud, France
Teresa Escrig	Universitat Jaume I, Spain
Juan Flores	Universidad Michoacana, Mexico
Ken Forbus	Northwestern University, USA
Michael Hofbaur	Technische Universität Graz, Austria
Hidde de Jong	INRIA Rhône-Alpes, France
Johan de Kleer	PARC, USA
Tim Nuttle	University of Pittsburgh, USA
Paulo Salles	University of Brasilia, Brazil
Reinhard Stolle	BMW Car IT, Germany
Peter Struss	Technical University of Munich, Germany
Stefania Tentoni	IMATI-CNR, Pavia, Italy
Louise Trave-Massuyes	LAAS-CNRS, France

Contents

A formal theory of qualitative size and distance relations between regions

Thomas Bittner[1,2,3,4] **and Maureen Donnelly**[1,3]
[1] Department of Philosophy, [2] Department of Geography
[3] New York State Center of Excellence in Bioinformatics and Life Sciences
[4] National Center of Geographic Information and Analysis (NCGIA)
State University of New York at Buffalo
{bittner3,md63}@buffalo.edu

Abstract

We present a formal theory of qualitative distances between regions based on qualitative size relations. Using standard mereological relations, a sphere-predicate, and qualitative size relations such as roughly-the-same-size-as and negligible-in-size-with-respect-to, we define qualitative distance relations such as close-to, near-to, away-from, and far-away-from.

Relations such as roughly-the-same-size-as and negligible-in-size-with-respect-to are context-dependent and vague. The primary focus in the formal theory presented in this paper is on the *context-independent* logical properties of these sorts of qualitative size and distance relations. We are especially interested in how these relations interact with familiar mereological relations. In developing our formal theory, we draw upon work on order of magnitude reasoning in Artificial Intelligence.

Introduction

Qualitative distance relations such as close-to, near-to, and far-away are important in geography (Tobler 1970), in Artificial Intelligence (Hernandez, Clementini, & Di Felice 1995; Clementini, Di Felice, & Hernández 1997; Davis 1989; 1999), spatial cognition (Talmy 1983; Herskowitz 1986), and other disciplines. Most attempts to formalize qualitative distance relations are based on the order of magnitude reasoning pioneered in (Raiman 1988; 1991; Mavrovouniotis & Stephanopoulos 1988; Dague 1993a; 1993b). Order of magnitude reasoning deals with qualitative relations between quantities, such as roughly-the-same-size-as and negligible-with-respect-to.

In this paper we present a mereological theory for domains of spatial regions and extend this theory by adding qualitative size relations and a 'sphere' predicate. In the resulting theory we are able to define qualitative distance relations such as close-to, near-to and far-away-from, etc. It is important for characterizing qualitative *distance* relations between spatially extended regions to take the *size* of the regions into account. Whether, for example, the relation near-to holds between regions x and y which are a fixed quantitative distance apart depends in part on the sizes of x or y. For example, let x be negligible in size with respect to y and suppose that the least distance between points in x and y is very small with respect to the size of y but large with respect to the size of x. Then y may be near to x (on y's scale) but x might not be near y (on x's scale). As pointed out in (Worboys 2001), in many cases utterances involving qualitative distance relations between extended objects can be understood only if the size of the objects is taken into account.

The theory presented in this paper combines a version of region-based qualitative geometry (RBG) (Tarski 1956; Borgo, Guarino, & Masolo 1996; Bennett *et al.* 2000) with work from order of magnitude reasoning, especially (Dague 1993b). It gives a detailed account of the logical properties of qualitative size and distance relations. We show that qualitative size relations (essentially ordering relations) can be defined within a mereological framework extended by the primitive relations same-size-as and roughly-the-same-size-as, while qualitative distance relations need to be defined within the stronger framework of region-based geometry.[1]

Mereology

We present our formal theory of qualitative size and distance relations in a first-order predicate logic with identity. Variables range over regions of space. Spatial regions are here assumed to be parts of an independent background space in which all objects are located. On the intended interpretation, regions are the non-empty regular closed subsets of a three-dimensional Euclidean space.

We introduce the primitive binary predicate P, where Pxy is interpreted as: x is part of y. We define: x

[1] (Bennett 2002) sketches logical properties of region size measures within the framework of RBG by introducing the primitive 'sphere of insignificant size'.

Qualitative size relations are also discussed in (Gerevini & Renz 1998) in the context of a constraint based framework based on the RCC theory. The paper does not give an explicit axiomatization of relations such as roughly-the-same-size-as and negligible-with-respect-to. Neither does it consider qualitative distance relations.

overlaps y if and only if there is a z such that z is part of both x and y (D_O); x is a *proper part* of y if and only if x is a part of y and y is not a part of x (D_{PP}); z is the sum of x and y if and only if for all w, w overlaps z if and only if w overlaps x or w overlaps y (D_+); z is the *difference* of y in x if and only if any region w overlaps z if and only if w overlaps some part of x that does not overlap y (D_-).

$$
\begin{array}{ll}
D_O & O\ xy \equiv (\exists z)(P\ zx \wedge P\ zy) \\
D_{PP} & PP\ xy \equiv P\ xy \wedge \neg P\ yx \\
D_+ & +xyz \equiv (w)(Owz \leftrightarrow (O\ wx \vee O\ wy)) \\
D_- & -\ xyz \equiv (w)(O\ wz \leftrightarrow \\
 & \qquad (\exists w_1)(P\ w_1x \wedge \neg O\ w_1y \wedge O\ w_1w))
\end{array}
$$

We add the usual axioms of reflexivity (A1), antisymmetry (A2), and transitivity (A3). We also assume the following existence axioms: if x is not a part of y then there is a z such that z is a difference of y in x (A4), for any regions x and y there is a region z that is the sum of x and y (A5).

$$
\begin{array}{ll}
A1 & P\ xx \\
A2 & P\ xy \wedge P\ yx \rightarrow x = y \\
A3 & P\ xy \wedge P\ yz \rightarrow P\ xz \\
A4 & \neg P\ xy \rightarrow (\exists z)(-\ xyz) \\
A5 & (\exists z)(+xyz)
\end{array}
$$

We can prove: x and y are identical if and only if they overlap exactly the same regions (T1). We can also prove that sums and differences are unique whenever they exist (T2-T3). Together, A4 and T2 ensure that summation is a functional operator.

$$
\begin{array}{ll}
T1 & x = y \leftrightarrow (z)(O\ zx \leftrightarrow O\ zy) \\
T2 & +\ xyz_1 \wedge +\ xyz_2 \rightarrow z_1 = z_2 \\
T3 & -\ xyz_1 \wedge -\ xyz_2 \rightarrow z_1 = z_2
\end{array}
$$

EMR, extensional mereology for regions, is the theory axiomatized A1-A5 (Simons 1987; Varzi 1996).

Ordering based on the exact size

In the next two sections, we present a modified version of our theory of granular parthood and qualitative cardinalities (Bittner & Donnelly 2006).[2]

We use $\|x\|$ in the meta-language to refer to the exact volume size of region x. In the formal theory we introduce the *same size* relation $\sim$ where, on the intended interpretation, $x \sim y$ holds if and only if $\|x\| = \|y\|$. We then define that the size of x is *less than or equal* to the size of y if and only if there is a region z that is a part of y and has the same size as x ($D_\leq$).

$$D_\leq \quad x \leq y \equiv (\exists z)(z \sim x \wedge P\ zy)$$

On the intended interpretation, $x \leq y$ holds if and only if $\|x\|$ is less than or equal to $\|y\|$.

We require: $\sim$ is reflexive (A6); $\sim$ is symmetric (A7); $\sim$ is transitive (A8); if x is part of y and x and y have the same size then y is part of x (A9); for any x and y, the size of x is less than or equal to the size of y or the size of y is less than or equal to the size of x (A10); if the size of x is less than or equal to the size of y and the size of y is less than or equal to the size of x, then x and y have the same size (A11).

$$
\begin{array}{ll}
A6 & x \sim x \\
A7 & x \sim y \rightarrow y \sim x \\
A8 & x \sim y \wedge y \sim z \rightarrow x \sim z \\
A9 & P\ xy \wedge x \sim y \rightarrow P\ yx \\
A10 & x \leq y \vee y \leq x \\
A11 & x \leq y \wedge y \leq x \rightarrow x \sim y
\end{array}
$$

We can prove: if x is identical to y, then x and y are of the same size (T4); if x is part of y and y is part of x, then x and y have the same size (T5); if x is part of y and x and y have the same size then x and y are identical (T6); if x is a part of y, then the size of x is less than or equal to the size of y (T7); $\leq$ is reflexive (T8); $\leq$ is transitive (T9); if the size of x is less than or equal to the size of y and y and z have the same size, then the size of x is less than or equal to the size of z (T10); if z and x have the same size and the size of z is less than or equal to the size of y then the size of z is less than or equal to the size of x (T11).

$$
\begin{array}{ll}
T4 & x = y \rightarrow x \sim y \\
T5 & P\ xy \wedge P\ yx \rightarrow x \sim y \\
T6 & P\ xy \wedge x \sim y \rightarrow x = y \\
T7 & P\ xy \rightarrow x \leq y \\
T8 & x \leq x \\
T9 & x \leq y \wedge y \leq z \rightarrow x \leq z \\
T10 & x \leq y \wedge y \sim z \rightarrow x \leq z \\
T11 & z \sim x \wedge x \leq y \rightarrow z \leq y
\end{array}
$$

Thus, $\sim$ is an equivalence relation, $\leq$ is reflexive and transitive, and $\sim$, $\leq$, P, and $=$ are logically interrelated in the expected ways.

Roughly the same size, negligible in size

We introduce the relations *roughly the same size* ($\approx$) and *negligible in size* ($\ll$) as in (Bittner & Donnelly 2006). Let ω be a parameter such that $0 < \omega < 0.5$. On one possible class of interpretations, x has *roughly same size as* y if and only if $1/(1+\omega) \leq \|x\|/\|y\| \leq 1+\omega$. x is a *negligible in size* with respect to y if and only if $\|x\|/\|y\|$ is less than $\omega/(1+\omega)$.

Consider Figure 1. Values for the size of x range along the positive horizontal axis and values for the size of y

[2] Axioms A6-A11 correspond to AC7-AC10,AC12,AC13 in (Bittner & Donnelly 2006) (BD06). The axioms are similar in structure. However in this paper we work in a mereological framework formalizing size relations between regions while in BD06 we worked in the framework of finite non-empty collections formalizing relations on cardinalities of collections. Similarly, A12-A16 correspond to AC14,AC15,AC17,AC18 in BD06. The axiom A14 of this paper is a theorem in BD06. Unlike in GP06 we do not include an axiom constraining the relationships between size relations the and summation operation. In BD06 granular parthood for material objects is formalized. The size relations discussed in the present paper hold between arbitrary regions and not only between parts.

range along the positive vertical axis. If x and y have the same size then $(\|x\|, \|y\|)$ represents a point on the dotted line. If $1/(1+\omega) \leq \|x\|/\|y\| \leq 1+\omega$ (i.e., x has roughly the same size as y), then $(\|x\|, \|y\|)$ represents a point lying within the area delimited by the dashed lines. If $\|x\|/\|y\|$ is smaller than $\omega/(1+\omega)$ (i.e., x is negligible with respect to y), then $(\|x\|, \|y\|)$ represents a point lying between the positive vertical axis and the solid diagonal line.

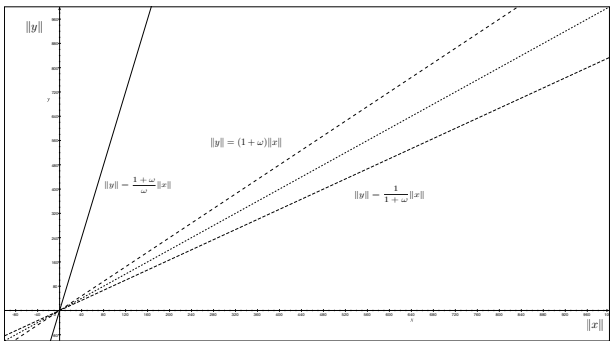


Figure 1: Graph for $\omega = 0.2$

Now consider a fixed region y and imagine that different values of ω are appropriate for different contexts. The smaller the value of ω, the smaller the value of $|\|x\| - \|y\||$ must be for x to count as close in size to y and the smaller $\|x\|$ must be for x to count as negligible in size with respect to y. To picture this situation graphically: the smaller the value of ω, the narrower the corridor between the dashed diagonal lines in Figure 1 and also the narrower the corridor between the solid diagonal line and the positive vertical axis.

We require: $\approx$ is reflexive (A12); $\approx$ is symmetric (A13); if x and y have roughly the same size and y and z have the same size, then x and z have roughly the same size (A14); if x and y have roughly the same size and x is a part of z and z is a part of y, then z and x, as well as z and y, have roughly the same size (A15).

$$\begin{array}{ll} A12 & x \approx x \\ A13 & x \approx y \rightarrow y \approx x \\ A14 & x \approx y \wedge y \sim z \rightarrow x \approx z \\ A15 & x \approx y \wedge P\ xz \wedge P\ zy \rightarrow (z \approx x \wedge z \approx y) \end{array}$$

Notice that unlike (Raiman 1991) and (Dague 1993b) we do not require $\approx$ to be transitive. In many of the intended models of our theory, it is possible to find regions $z_1, \ldots, z_n$ such that $x \approx z_1$, $z_1 \approx z_2$, ... and $z_n \approx y$ and but NOT $x \approx y$. Hence, adding a transitivity axiom for $\approx$ would give rise to a version of the Sorites paradox (Hyde 1996; van Deemter 1995).

We can prove: if x and y have the same size and y and z have roughly the same size, then x and z have roughly the same size (T12); if x and y have the same size, then x and y have roughly the same size (T13).

$$\begin{array}{ll} T12 & x \sim y \wedge y \approx z \rightarrow x \approx z \\ T13 & x \sim y \rightarrow x \approx y \end{array}$$

Region x is *negligible in size with respect to* region y if and only if there are regions z_1 and z_2 such that (i) x and z_1 have the same size, (ii) z_1 is a part of y, (iii) z_2 is the difference of z_1 in y and (iii) z_2 and y have roughly the same size ($D_{\ll}$).

$$D_{\ll} \quad x \ll y \equiv (\exists z_1)(\exists z_2)(z_1 \sim x \wedge P\ z_1 y \wedge -y z_1 z_2 \wedge z_2 \approx y)$$

As pointed out above, when $\approx$ is interpreted so that $z \approx y$ holds if and only if $1/(1+\omega) \leq \|z\|/\|y\| \leq 1+\omega$, then $x \ll y$ holds if and only if $\|x\|/\|y\|$ is smaller than $\omega/(1+\omega)$.

We require that if x is negligible with respect to y and the size of y is less than or equal to the size of z, then x is negligible with respect to z (A16).

$$A16 \quad x \ll y \wedge y \leq z \rightarrow x \ll z$$

We can prove: if x is negligible with respect to y, then x is smaller than y (T14); if the size of x is less than or equal to the size of y and y is negligible with respect to z, then x is negligible with respect to z (T15); if x is a part of y and y is negligible with respect to z, then x is negligible with respect to z (T16); if x is negligible with respect to y and y is part of z, then x is negligible with respect to z (T17); $\ll$ is transitive (T18).

$$\begin{array}{ll} T14 & x \ll y \rightarrow (x \leq y \wedge x \not\sim y) \\ T15 & x \leq y \wedge y \ll z \rightarrow x \ll z \\ T16 & P\ xy \wedge y \ll z \rightarrow x \ll z \\ T17 & x \ll y \wedge P\ yz \rightarrow x \ll z \\ T18 & x \ll y \wedge y \ll z \rightarrow x \ll z \end{array}$$

Thus, the relation negligible-in-size-with-respect-to has the expected logical properties. We call the theory, which extends EMR by axioms A6-A16, QSizeR.

Spheres and connectedness

We introduce the primitive predicate S where $S\ x$ is interpreted as x is a sphere. We define: x is *maximal with respect to y in z* if and only if (i) x, y, and z are spheres, (ii) x and y are non-overlapping parts of z, and (iii) every sphere u that has x as a part either is identical to x, overlaps y, or is not a part of z (D_{Mx}). x is a *concentric* proper part of y if and only if (i) x and y are spheres, (ii) x is a proper part y and (iii) all spheres that are maximal with respect to x in y have the same size (D_{CoPP}).

$$\begin{array}{ll} D_{Mx} & Mx\ xyz \equiv S\ x \wedge S\ y \wedge S\ z \wedge \\ & \quad P\ xz \wedge P\ yz \wedge \neg O\ xy \wedge \\ & \quad (u)(S\ u \wedge P\ xu \rightarrow (x = u \vee O\ uy \vee \neg Puz) \\ D_{CoPP} & CoPP\ xy \equiv S\ x \wedge S\ y \wedge PP\ xy \wedge \\ & \quad (u)(v)(Mx\ uxy \wedge Mx\ vxy \rightarrow u \sim v)) \end{array}$$

We require that the following spheres exist: Every region has a sphere as a part (A17). Every sphere has a concentric proper part (A18). If sphere x is a proper part of sphere y then there is a sphere z that is maximal with respect to x in y (A19).

$$\begin{array}{ll} A17 & (\exists z)(S\ z \wedge P\ zx) \\ A18 & S\ x \rightarrow (\exists y)(S\ y \wedge CoPP\ yx) \\ A19 & S\ x \wedge S\ y \wedge PP\ xy \rightarrow (\exists z)(Mx\ zxy) \end{array}$$

Similar to (Bennett *et al.* 2000) we then define that two regions x and y are *connected* if and only if there is a sphere z that overlap x and y and all spheres that are concentric proper parts of z also overlap x and y (D_C).

$$D_C \quad C\ xy \equiv (\exists z)(S\ z \wedge O\ zx \wedge O\ zy \wedge (u)(CoPP\ uz \rightarrow (O\ ux \wedge O\ uy))$$

On the intended interpretation, the connection relation C holds between regions x and y if and only if the distance between them is zero (where the distance between regions is here understood as the greatest lower bound of the distance between any point of the first region and any point of the second region).

We can prove that C is reflexive (T18a), symmetric (T18b), and that if x is part of y, then everything connected to x is connected to y (T18c).

$$T18a \quad C\ xx \qquad \begin{array}{ll} T18b & C\ xy \rightarrow C\ yx \\ T18c & P\ xy \rightarrow (z)(Czx \rightarrow Czy) \end{array}$$

We call the theory formed by axioms A1-A11 and A17-A19 *region-connection geometry* RCG.

Qualitative distance relations

We now use the sphere primitive, the connectedness relation, and the qualitative *size* relations of QSizeR to define qualitative *distance* relations such as close-to, near-to, and away-from.

Region x is *close* to region y if and only if either x and y are connected or there is a sphere z such that z is connected to both x and y and z is negligible in size with respect to x (D_{Cl}). x is *strictly close* to y if and only if x is close to y but not connected to y (D_{SN}). x is *near* to y if and only if either x and y are connected or there is a sphere z such that z is connected to x and y and the size of z is less than or equal to the size of x (D_N). x is *strictly near* to y if and only if x is near to y but not close to y (D_{SN}). x is *away* from y if and only if x is not near to y (D_A). x is *far away* from y if and only if (i) x and y are not connected and (ii) there is a sphere z such that z is connected to x and y, and (iii) x is negligible in size with respect to all spheres w that are connected to x and y (D_{FA}). x is *moderately away* from y if and only if x is away from y but not far away from y (D_{MA}).[3]

$$\begin{array}{ll} D_{Cl} & Cl\ xy \equiv C\ xy \vee (\exists z)(S\ z \wedge C\ zx \wedge C\ zy \wedge z \ll x) \\ D_{SCl} & SCl\ xy \equiv Cl\ xy \wedge \neg C\ xy \\ D_N & N\ xy \equiv C\ xy \vee (\exists z)(S\ z \wedge C\ zx \wedge C\ zy \wedge z \leq x) \\ D_{SN} & SN\ xy \equiv N\ xy \wedge \neg Cl\ xy \\ D_A & A\ xy \equiv \neg N\ xy \\ D_{FA} & FA\ xy \equiv \neg C\ xy \wedge (\exists z)(S\ z \wedge C\ zx \wedge C\ zy) \wedge (w)(S\ w \wedge C\ wx \wedge C\ wy \rightarrow x \ll w) \\ D_{MA} & MA\ xy \equiv A\ xy \wedge \neg FA\ xy \end{array}$$

[3]Notice that, unlike the other distance relations, N and A are crisp, i.e., their interpretations do not depend on ω (See also Table 1). A possible definition that takes the vagueness of 'near' better into account may be $N'\ xy \equiv C\ xy \vee (\exists z)(S\ z \wedge C\ zx \wedge C\ zy \wedge z \approx x)$.

Let $d(x, y)$ be the greatest lower bound of the distance between any point of x and any point of y and let $d_{\|x\|}$ be the diameter of a sphere of size $\|x\|$. When $\approx$ is interpreted so that $z \approx y$ holds if and only if $1/(1+\omega) \leq \|z\|/\|y\| \leq 1+\omega$, then the distance relations defined above hold for the distance ranges specified in Table 1.

Relation	holds for distance ranges
$Cl\ xy$	$0 \leq d(x,y) \leq (\omega * d_{\|x\|})/(1+\omega)$
$SCl\ xy$	$0 < d(x,y) \leq (\omega * d_{\|x\|})/(1+\omega)$
$N\ xy$	$0 \leq d(x,y) \leq d_{\|x\|}$
$SN\ xy$	$(\omega * d_{\|x\|})/(1+\omega) < d(x,y) \leq d_{\|x\|}$
$A\ xy$	$d_{\|x\|} < d(x,y)$
$FA\ xy$	$(d_{\|x\|} * (1+\omega))/\omega < d(x,y)$
$MA\ xy$	$d_{\|x\|} < d(x,y) \leq (d_{\|x\|} * (1+\omega))/\omega$

Table 1: Distance ranges for which the qualitative distance relations hold on the intended interpretation in context ω. (In this table $<$ and $\leq$ refer to the total (strict) ordering on the real numbers.)

Consider Figure 2. In the center of the concentric circles there is the circle-shaped region x of size $\|x\|$ and radius $r(x)$. In the center of x is the origin of our coordinate system. Using our qualitative distance relations we can identify the following nested ring structure around x for every context ω: The relation strictly-close holds between x and any region y which has points in the *SCl*-ring (the ring between $r(x)$ and $r(x)+(\omega * d_{\|x\|})/(1+\omega)$ excluding the boundary $r(x)$).

The relation strictly-near holds between x and any region y which has points in the *SN*-ring (in the ring between $r(x) + (\omega * d_{\|x\|})/(1+\omega)$ and $r(x) + d_{\|x\|}$, excluding the boundary $r(x) + (\omega * d_{\|x\|})/(1+\omega)$).

The relation moderately-far-away holds between x and any region y which has points in the *MA*-ring (the ring between $r(x) + d_{\|x\|}$ and $r(x) + (d_{\|x\|} * (1+\omega))/\omega$, excluding the boundary $r(x) + d_{\|x\|}$).

The relation far-away holds between x and any region y which has all points in the *FA*-ring (outside the circle with radius $r(x) + (d_{\|x\|} * (1+\omega))/\omega$).

The following theorems are immediate consequences of our definitions: x is close to y if and only if x is connected to y or x is strictly close to y (T19); if x and y are connected then x and y are not strictly close (T20); x is near to y if and only if x is close to y or x is strictly near to y (T21); if x and y are close then x and y are not strictly near (T22); x is away from y if and only if x moderately away from y or x is far away from y (T23); if x and y are moderately away then x and y are not far away (T24).

$$\begin{array}{ll} T19 & Cl\ xy \leftrightarrow (C\ xy \vee SCl\ xy) \\ T20 & C\ xy \rightarrow \neg SCl\ xy \\ T21 & N\ xy \leftrightarrow (Cl\ xy \vee SN\ xy) \\ T22 & Cl\ xy \rightarrow \neg SN\ xy \\ T23 & A\ xy \leftrightarrow (MA\ xy \vee FA\ xy) \\ T24 & MA\ xy \rightarrow \neg FA\ xy \end{array}$$

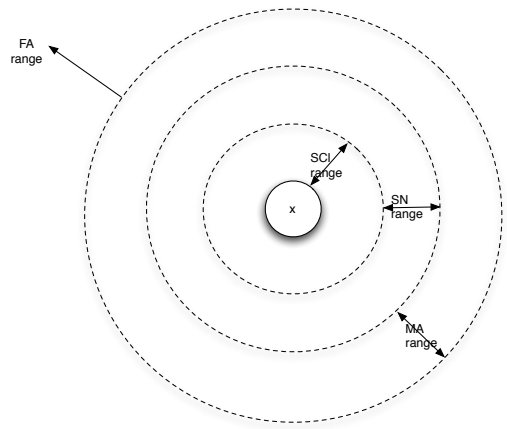


Figure 2: Qualitative distance diagram for a circular region x.

The implication hierarchy and the sets of jointly exhaustive and pair-wise disjoint relations which follow from these theorems are pictured graphically in Figure 3.

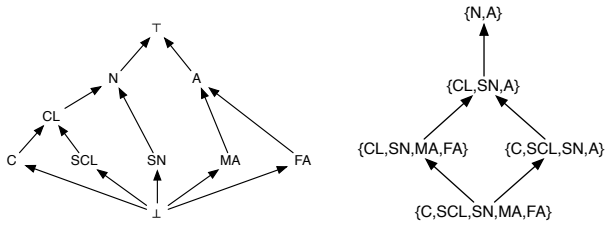


Figure 3: Implication hierarchy of the qualitative distance relations (left). Sets of JEPD qualitative distance relations (right).

We can also prove that Cl and N are reflexive and that SCl, SN, A, MA, and FA are irreflexive.

Notice that NONE of the defined distance relations is symmetric. For example, a road-sized region may be (on the scale of the road) close to a pebble-sized region in an adjacent ditch, even if the pebble-sized region is not (on the scale of the pebble) close to the road-sized region. However we can prove: if the size of x is less than or equal to y and x is close to y then y is also close to x (T25); if the size of x is less than or equal to y and x is strictly close to y then y is strictly close to x (T26); if the size of x is less than or equal to y and x is near to y then y is near to x (T27); if the size of x is less than or equal to y and x is strictly near to y then y is near to x (T28); if the size of x is equal to y and x is strictly near to y then y is strictly near to x (T29); if the size of y is less than or equal to x and x and y are away from one another then y and x are away from one another (T30); if the size of y is less than or equal to x and x and y are far away then y and x are far away (T31); if the size of y is less than or equal to x and x and y are moderately away then y and x are away (T32); if the size of y is equal to the size of x and x and y are moderately away then y and x are moderately away (T33).

$$
\begin{array}{ll}
T25 & x \leq y \wedge Cl\ xy \rightarrow Cl\ yx \\
T26 & x \leq y \wedge SCl\ xy \rightarrow SCl\ yx \\
T27 & x \leq y \wedge N\ xy \rightarrow N\ yx \\
T28 & x \leq y \wedge SN\ xy \rightarrow N\ yx \\
T29 & x \sim y \wedge SN\ xy \rightarrow SN\ yx \\
T30 & y \leq x \wedge A\ xy \rightarrow A\ yx \\
T31 & y \leq x \wedge FA\ xy \rightarrow FA\ yx \\
T32 & y \leq x \wedge MA\ xy \rightarrow A\ yx \\
T33 & x \sim y \wedge MA\ xy \rightarrow MA\ yx
\end{array}
$$

Theorems T25-T33 reflect the logical interrelationships between the qualitative distance relations and the relative size of the regions involved.

We can also prove the following theorems about logical interrelationships between parthood and the various qualitative distance relations: if x and y are close and z has y as a part then x and z are close (T34); if x and y are near and z has y as part then x and z are near (T35); if x is a part of y and y and z are away then x and z are away (T36).

$$
\begin{array}{ll}
T34 & Cl\ xy \wedge P\ yz \rightarrow Cl\ xz \\
T35 & N\ xy \wedge P\ yz \rightarrow N\ xz \\
T36 & P\ xy \wedge A\ yz \rightarrow A\ xz
\end{array}
$$

We call the theory which extends QSizeR and RCG by the definitions for qualitative distance relations QDistR.

Conclusions

We have presented an axiomatic theory of qualitative size and distance relations between regions. The theory is based on the formal characterization of the primitive predicates and relations: part-of (P), sphere (S), exactly-the-same-size ($\sim$), and roughly-the-same-size ($\approx$). In our theory, we are able to formally distinguish: i) regions that are negligible in size with respect to one another, ii) regions that are close, near, far away, etc. We thereby extend existing work on mereo-geometries and order of magnitude reasoning.

The axiomatic theory presented in this paper is part of the top-level ontology 'Basic Formal Ontology' (BFO). BFO is developed using *Isabelle*, a computational system for implementing logical formalisms (Nipkow, Paulson, & Wenzel 2002). The computational representation of BFO consists of several hierarchically organized sub-theories. An automatically generated WEB presentation of the theory containing all axioms, definitions, theorems, and the computer-verified proofs can be accessed at `http://www.ifomis.org/bfo/fol`.

Relations such as roughly-the-same-size-as, negligible-in-size-with-respect-to, close-to, far-away, etc, are *context-dependent* and *vague*. Context is represented abstractly in numerical parameters which determine the canonical interpretations of the qualitative size and distance relations of the formal theory.

Although the canonical models use precise numerical parameters for fixing the interpretations, it is not expected that precise numerical parameters are fixed in actual practical contexts. Since the qualitative size and distance relations are vague, in many cases (at best) we can associate contexts demanding high precision with a different range of numerical parameters than contexts requiring only loose precision.

Since the logical properties of the relations of our theory are valid over a range of numerical parameters, the formal theory can be used for reasoning even where qualitative size and distance relations lack precise numerical definitions. Thus the primary focus in the formal theory presented in this paper is on the *context-independent logical* properties of these sorts of qualitative size and distance relations and the logical interrelations among one another and the mereotopological relations.

References

Bennett, B.; Cohn, A. G.; Torrini, P.; and Hazarika, S. M. 2000. A foundation for region-based qualitative geometry. In Horn, W., ed., *Proceedings of ECAI-2000*, 204–208.

Bennett, B. 2002. Physical objects, identity and vagueness. In Fensel, D.; McGuinness, D.; and Williams, M.-A., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Eighth International Conference (KR2002)*. San Francisco, CA: Morgan Kaufmann.

Bittner, T., and Donnelly, M. 2006. A theory of granular parthood based on qualitative cardinality and size measures. In Bennett, B., and Fellbaum, C., eds., *Proc. 4th Int. Conference on Formal Ontology in Information Systems*, 65–76.

Borgo, S.; Guarino, N.; and Masolo, C. 1996. A pointless theory of space based on strong connection and congruence. In Aiello, L. C.; Doyle, J.; and Shapiro, S., eds., *Principles of Knowledge Representation and Reasoning (KR96)*, 220–229. Morgan Kaufmann.

Clementini, E.; Di Felice, P.; and Hernández, D. 1997. Qualitative representation of positional information. *Artificial Intelligence* 95(2):317–356.

Dague, P. 1993a. Numeric reasoning with relative orders of magnitude. In *Proceedings of the National Conference on Artificial Intelligence*, 541–547.

Dague, P. 1993b. Symbolic reasoning with relative orders of magnitude. In *Proc. 13th Intl. Joint Conference on Artificial Intelligence*, 1509–1515. Morgan Kaufmann.

Davis, E. 1989. Order of magnitude reasoning in qualitative differential equations. In de Kleer, J., and Weld, D., eds., *Readings in Qualitative Physical Reasoning*. Morgan Kaufmann. 424–434.

Davis, E. 1999. Order of magnitude comparisons of distance. *Journal of AI Research* 10:1–38.

Gerevini, A., and Renz, J. 1998. Combining topological and qualitative size constraints for spatial reasoning. In *Proceedings of the 4th international conference on principles and practice of constraint programming (CP'98)*. Berlin: Springer-Verlag.

Hernandez, D.; Clementini, E.; and Di Felice, P. 1995. Qualitative distances. In Frank, A. U., and Kuhn, W., eds., *Spatial Information Theory, A Theoretical Basis for GIS*, LNCS. Semmering, Austria: Springer-Verlag.

Herskowitz, A. 1986. *Language and Spatial Cognition - An Interdisciplinary Study of the Propositions in English*. Studies in natural language processing. Cambridge University Press.

Hyde, D. 1996. Sorites paradox. In *Stanford Encyclopedia of Philosophy*.

Mavrovouniotis, M., and Stephanopoulos, G. 1988. Formal order-of-magnitude reasoning in process engineering. *Computers and Chemical Engineering* 12:867–881.

Nipkow, T.; Paulson, L. C.; and Wenzel, M. 2002. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer.

Raiman, O. 1988. Order of magnitude reasoning. In Weld, D., and de Kleer, J., eds., *Readings in Qualitative Reasoning about Physical Systems*. San Mateo, Calefornia: Morgan Kaufmann Publishers, INC.

Raiman, O. 1991. Order of magnitude reasoning. *Artificial Intelligence* 51:11–38.

Simons, P. 1987. *Parts, A Study in Ontology*. Oxford: Clarendon Press.

Talmy, L. 1983. How language structures space. In Pick, H., and Acredolo, L., eds., *Spatial Orientation: Theory, Research, and Application*. New York, NY: Plenum Press.

Tarski, A. 1956. Foundations of the geometry of solids. In *Logic, Semantic, Metamathematics*. Oxford Clarendon Press.

Tobler, W. 1970. A computer movie simulating urban growth in the detroit region. *Economic Geography* 46(2):234–240.

van Deemter, K. 1995. The sorites fallacy and the context-dependence of vague predicates. In Kanazawa, M.; Pinon, C.; and de Swart, H., eds., *Quantifiers, Deduction, and Context*. CSLI Publications, Stanford, CA. 59–86.

Varzi, A. 1996. Parts, wholes, and part-whole relations: The prospects of mereotopology. *Data and Knowledge Engineering* 20(3):259–86.

Worboys, M. F. 2001. Nearness relations in environmental space. *International Journal of Geographical Information Science* 15(7):633–651.

A Garp3 Model of Environmental Sustainability in the Danube Delta Biosphere Reserve based on Qualitative Reasoning Concept

Eugenia Cioaca*, Bert Bredeweg **, Paulo Salles***, Tim Nuttle ****

*Danube Delta National Institute, Tulcea, Romania; eugenia@indd.tim.ro
** Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands; bredeweg@science.uva.nl
*** University of Brasília, Institute of Biological Sciences, paulo.bretas@uol.com.br
**** Institute of Ecology, Friedrich Schiller University, Jena, Germany; Tim.Nuttle@uni-jena.de

Abstract

The paper presents a qualitative reasoning (QR) model of sustainable development issues of the Danube Delta Biosphere Reserve (DDBR, Romania) environmental system. This model contributes to NaturNet-Redime projects goal to assist the implementation of the EU's Strategy of Sustainable Development. The DDBR QR model emphasizes the main causes that hamper achievement of sustainable development in the DDBR. Specifically, following a standardized framework for conceptual description of QR case studies, we have organized our expert knowledge about negative effects of water pollution from the Danube catchment area on aquatic biota and human health in and around the DDBR. We present essential background about the model system, and describe how available knowledge was encapsulated into QR knowledge structures including model fragments and scenarios. Finally, we present simulation output based on this knowledge and discuss how this output contributes to understanding factors affecting sustainability of the DDBR.

Introduction

Qualitative Reasoning is of great importance for developing, strengthening and further improving education and training on topics dealing with systems and their behaviors (Bredeweg and Forbus 2003). To meet the objectives of the European Union's Strategy for Sustainable Development (SSD) that call for increasing participation in the process of making decisions that affect sustainable development (SD), stakeholders, decision makers, and citizens must gain a better understanding the factors that affect SD (European Commission 2001). SD is broadly defined as "a real increase in well-being and standard of life for the average person that can be maintained over the long-term without degrading the environment or compromising the ability of future generations to meet their own needs" (Brundtland and the World Commission on Environment and Development 1987, Cunningham and Cunningham 2005).

Part of the FP6 NaturNet-Redime project involves developing qualitative reasoning (QR) models of five case studies that explore different SD issues and scenarios, in order to support these objectives of the SSD. The goal is to represent SD problems from different systems and perspectives and build an online curriculum about SD that focuses on user interaction with QR models.

Both to support the model building effort as well as to facilitate integration of the different models, Bredeweg et al. (2006) developed a "structured approach to qualitative modeling". Researchers from five case studies (the Danube Delta Biosphere Reserve, Romania; Riacho Fundo, Brasil; River Mesta, Bulgaria; Salmon restoration in England, and river catchment restoration in Austria) have been following this methodology.

The goal of this paper is to present a description of the DDBR model system, main model goals, the system global behavior, the main model ingredients as they are implemented into the Garp3 workbench (representative **scenarios** and **model fragments**) and the scenario simulation results. The QR models will be used by end users for learning about specific conditions to be fulfilled by the modeled system (either social, economic, or environmental) in order to contribute to increased public involvement as called for in the Strategy for Sustainable Development.

Model System

The DDBR - located at the mouth of the Danube River before it reaches the Black Sea - was declared as a World Heritage Site and Wetland of International Importance since 1990 (according to The Ramsar Convention on Wetlands, signed in Ramsar, Iran, in 1971). Its area of 5,800 sq. km, making it one of the greatest wetlands in the world, contains 30 types of terrestrial and aquatic ecosystems, of which 23 are natural or artificially modified and 7 are man-made ecosystems, including human settlements (Oosterberg et al. 2000). The DDBR's status as a biosphere reserve dictates that all social and economic actions must fall in line with **biodiversity conservation and protection measures**. Thus, the most appropriate concept of sustainable development for DDBR can be expressed by development through biodiversity, where most flora and fauna species are protected both to meet obligations of international conventions, but also to serve as natural resources for social and economic development of the region.

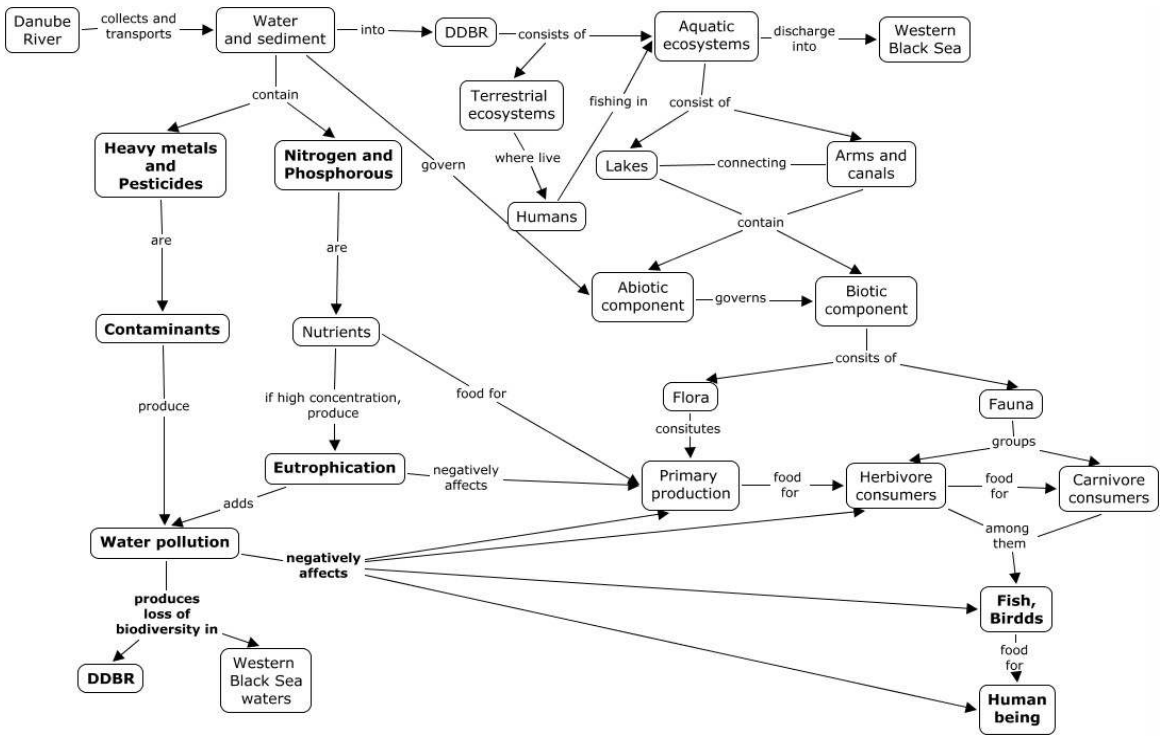


Figure 1. Concept map for Danube Delta Biosphere Reserve water pollution – the negative effect on DDBR biodiversity and human health.

Stakeholder Issues

Scientists from DDBR met with local stakeholders to determine threats to conserve and develop these resources within the DDBR. The stakeholders involved in DDBR management include: nature conservation and protection bodies (DDBRA, NGOs), fishery and fishing companies, tourism companies, fluvial and marine transport companies, and recreational hunting groups.

Stakeholders identified the following threats:

- Decline in biodiversity (number of species) over the last several decades
- Contamination of water and fish from pollutants
- Concern about contamination in humans
- Reduction of fish diversity and abundance.

Decline in biodiversity (Otel and Ciocarlan 2000) is most likely a direct result of loss of wetlands through embankment works for different types of land use (agricultural polders, fishponds, and forest plantations), summing 15% of the whole DDBR surface. This has reduced habitat for migratory waterfowl, an important draw for ecotourism. Contamination from water pollutants is also an important potential mechanism for threatening biodiversity in the DDBR.

Contamination of water and fish from pollutants also contributes to health problems in humans. Contaminants come in, basically, two forms: heavy nutrient loads from agricultural fertilizers and heavy metals from industry. In both cases, most of the pollutants originate from far upstream in the vast Danube River catchments. Heavy nutrient loads lead to algal blooms, which can result in toxic by-products form algae as well as depletion of oxygen in the water when algae die and are degraded by bacteria. This can cause die-offs in fish. Heavy metals in the DDBR waters threaten human populations in two ways, first from direct consumption because many people drink untreated water directly form the DDBR waterways, and second from consumption of fish which bioaccumulate heavy metals (Otchere 2003; Wachs 2000) in their muscle tissues.

Fishing has been the main occupation of the Danube Delta inhabitants since ancient times and although nowadays the supply of fish has diminished and changed in quality, it continues to be basic trade. Contamination is one of the causes of reduced fish diversity and population sizes, but also over-fishing (even poaching) has been recording especially within the last decades.

Model Specification

Before implementing the model in the Garp3 modeling workbench, we identified the main model goals, created a concept map to organize our thinking about processes, entities, and relations, and describe the kind of behavior we want the model to produce. These steps are described in the following subsections.

Main Model Goals

Contamination by pollutants is at the root of most of DDBR's threats to SD. Furthermore, in order to understand indirect as well as direct effects of pollutants

on humans, their effects on other ecosystem components, like fish, must also be understood. Thus, the DDBR model will describe the aquatic ecosystems behavior governed by water pollution rate and the ways it propagates to aquatic organisms and to humans living in or around the DDBR. The main goal of the DDBR model is:

- Understand and emphasize connections between water pollution in the Danube River catchment basin and health of human population living in and around the DDBR.

The model will be used to explain and educate the environment agency representatives, decision makers and stakeholders about the working of processes within the Danube River and their influence on these processes. Also the model will be used for argumentation purposes to convince decision makers what kind of actions they should take in order to improve (or stop) the Danube River water pollution process.

DDBR Concept Map

The concept map helps identify, clarify, and focus our knowledge about the system of interest (Figure 1). The model for the DDBR case study should capture the most relevant problems mentioned by the stakeholders, as reflected in the model goals. Hence, the concept map stresses effects of water pollution process on the aquatic biological components and human health for people living inside or around the DDBR.

System Selection and Structural Model

The DDBR full structural model (Figure 2) contains both terrestrial and aquatic ecosystems. It depicts a broader perspective on the entities and relations between them in the DDBR. The subset of entities that is relevant to the model goal specified above are shown in bold in Figure 2. The main system entities to be included in QR are thus model ***Water***, ***Fish***, and ***Human***. They can relate to each other by the following configurations:

› Fish ***lives in*** Water
› Human ***eats*** Fish
› Human ***drinks*** Water

Global Behavior

The main physical, physical-chemical and biological processes, influencing aquatic organism group behavior, in the framework of their Functional Feeding Group relationship, and humans (living in or around DDBR) are:

› water flow
› water eutrophycation – as result of Nutrients (mainly Nitrogen and Phosphorous compounds) increase
› phytoplankton bloom - overgrowth of algae and cyanobacteria (most of them are poisoning species)
› water pollution - mainly with nutrients, heavy metals, and cyanobacteria
› fish growth
› human being health.

Changes (increase/decline) in some groups influence other groups behavior. These cause-effect dependencies (Influence: I+/I- or Proportionality: P+/P-) for the aquatic ecosystems of the DDBR are presented in Figure 3).

In total, 12 processes are active in the DDBR aquatic environment that influences the abundance of each organism group. Changes in these abundances propagate to other quantities that affect other organism groups. Additionally, there are two agents (external influences) in

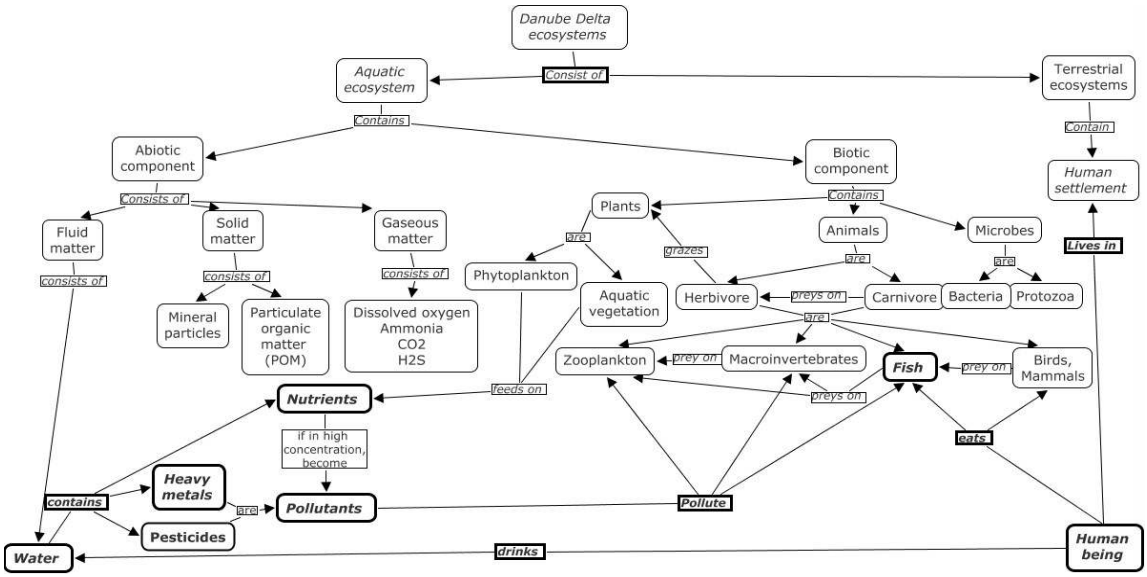


Figure 2. Structural model of the Danube Delta Biosphere Reserve aquatic ecosystems (Note: The structural entity hierarchy related directly to water pollution is represented in **bold**).

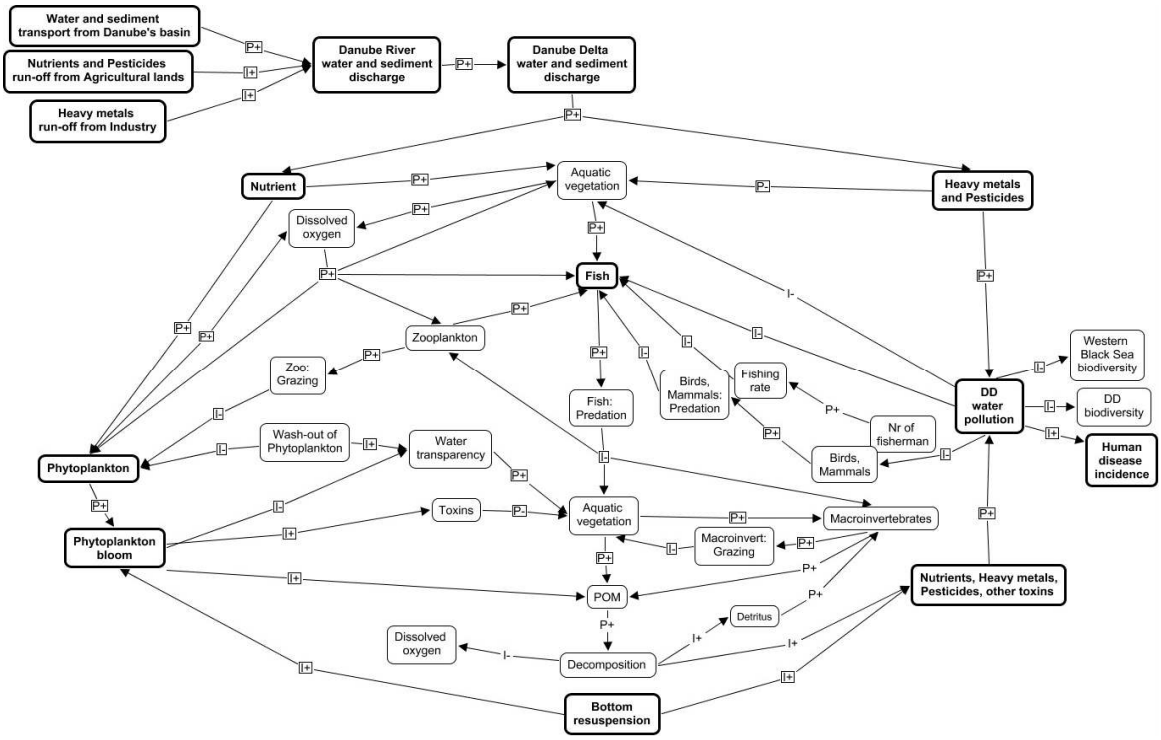


Figure 3. Global Causal model for Danube Delta Biosphere Reserve aquatic ecosystems water pollution.

the system: runoff from agriculture in the form of nutrients and runoff from industry in the form of heavy metals. These are considered external influences because they impact the DDBR system, but are located far upstream in the Danube catchment area. The implementation of these processes and agents can be seen in the section below on model fragments, and the context in which they operate in a simulation can be seen in the section on simulation results (below).

Space precludes a full description of details concerning the quantities that characterize each entity, as well as the quantity spaces that depict their qualitative values. These are discussed as they arise in the next sections describing scenarios, model fragments, and simulation results (see Cioaca et al. 2007 for full documentation).

Implementation details

QR model ingredients **implementation details** contain the detailed description of the modeled system: *Entities*, *Attributes*, *Configurations* (structural relationships between Entities), *Quantities* associated to each *Entity*, Quantity Spaces associated to *Quantities*, Scenarios, *Model Fragments*, *Agents* (External influences), and *Assumptions*. The main DDBR QR model ingredients are: 18 entities, 17 Scenarios, and 57 Model Fragments.

Figure 4 gives an overview of the entities involved in the model, and their hierarchical organization.

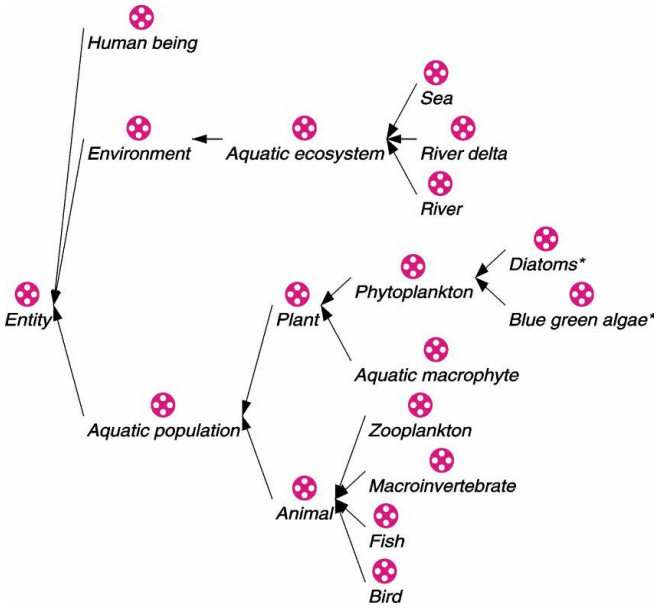


Figure 4. *Entity* hierarchy of the DDBR system.

Table 1 describes each of these entities and Table 2 describes configurations that are possible between entities.

Table 1. DDBR Entity summary.

Entity	Description
Human being	Human population living in/around the DDBR.
Environment	Physical space where aquatic ecosystems (***River***, ***River Delta***, and ***Sea***) belong to.
Aquatic population	Any biological entity living in water.
Aquatic ecosystem	A type of ecosystem where aquatic populations live.
River	Aquatic ecosystem where water flows from a catchment area to the sea.
River Delta	Aquatic ecosystem near the mouth of a river, consisting of branches, canals, and lakes.
Sea	Aquatic ecosystem at the end of a river and river delta.
Plant	This group is made of green plants (***Aquatic macrophytes, Phytoplankton***), organisms able to produce their own energy using sunlight to convert carbon dioxide and water into sugars by photosynthesis. ***Nutrients*** are their main food resource. Plants are the primary producers in all food chains since the materials they synthesize and store are the energy sources for all other organisms.
Animal	This group is made of all animals: ***Zooplankton***, ***Macroinvertebrates***, ***Fish***, ***Birds,*** and *Mammals*. They can be either herbivores or carnivores, and all are heterotrophic organisms (consumers) because they obtain their energy from other organisms (either plants or other animals).
Phytoplankton	Microscopic plant species (algae and bacteria) free-floating in the upper layer of water surface since sunlight is vital for their growth. Phytoplankton is the basis of most aquatic food chains, and also release oxygen into the water.
Aquatic macrophyte	Larger aquatic plant species; food resource for large animal species.
Diatoms	Predominant and harmless algae species division of Phytoplankton. Diatoms are a significant source of food for higher trophic levels, especially for *Zooplankton*.
Blue-green algae	Bacteria species (not algae), actually named as Cyanobacteria. Like other phytoplankton, they photosynthesize Most of species contain cyanotoxins in their cells. These toxins contribute to pollution and mortality of other organisms if concentrations are high..
Zooplankton	Microscopic species of animals inhabiting entire water column; food resource for larger animals, especially for fish.
Macro-invertebrate	Macroscopic animal species inhabiting both the water column and the bottom sediment (benthos).
Fish	Vertebrate species inhabiting almost any type of aquatic ecosystem.
Bird	Vertebrate species inhabiting aquatic ecosystems or the very near areas.

Table 2. DDBR Configuration summary

Configuration	Entity (from)	Entity (to)	Description
Drinks water from	Human being	River Delta	Specifies the link between people living inside the study area which provides their water source.
Eats	Human being	Fish	Specifies the link between people living inside the study area which provides their main food source.
Feeds on	Zoo-plankton	Diatoms	Specifies the feeding relationship between two aquatic species. One of them is consumer (predator) feeding on the other one (the prey).
	Macro-invertebrates	Aquatic macro-phyte	
	Fish	Zooplankton	
	Bird	Fish	
Flows in	River	River Delta	Specifies direction of water flow.
	River Delta	Western Black Sea	
In catchment area of	Agriculture (Run-off of Nutrients) Agent	Danube River	Specifies the ways the Agents exert their influence on River.
	Industry (Run-off of Heavy metals) Agent	Danube River	
Lives in	Diatoms	River Delta	Specifies that these species are aquatic species.
	Blue-green algae	River Delta	

Scenarios

A scenario describes the scope of a system to be modeled. It includes the *Entities*/*Agents* involved in modeled process, *Configurations* between *Entities*/*Agents*, *Entity*/*Agent Quantities* with initial values, and *Assumptions* (if necessary). This structure shows a possible start situation of the modeled process from which changes in the quantity values can be triggered, describing certain behaviors of the system. Here we present two scenarios that provide an overview of this model as related to the model goals.

Scenarios Concerning the Water Pollution Process

This Scenario models the DDBR water pollution process, its negative effects and the ways it propagates to aquatic

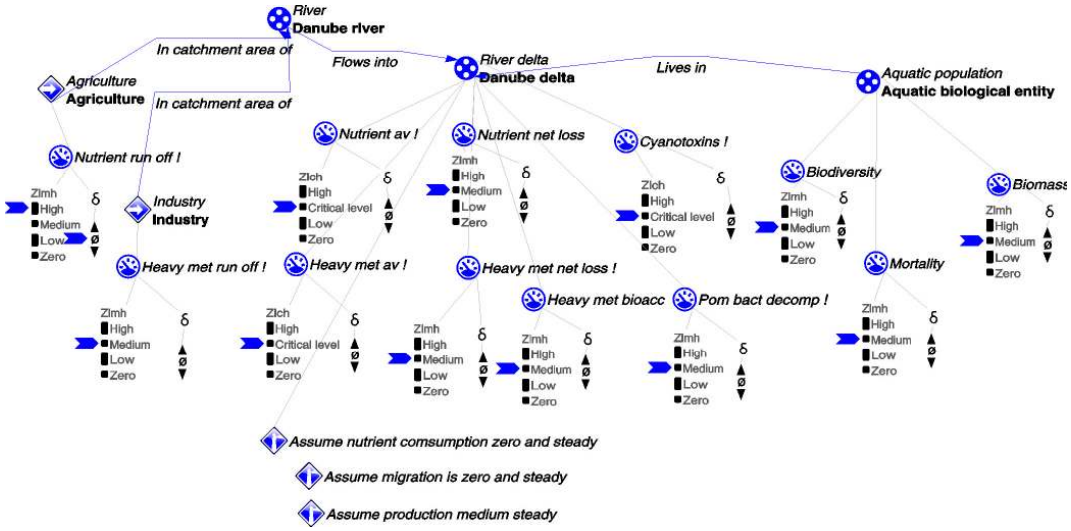


Figure 5. DD Water pollution and DD aquatic population biodiversity Scenario.

biotic component (Aquatic biological entities: Flora and Fauna populations) living in aquatic ecosystems of the modeled system. (Figure 5).

1. The modeled system's external influences (*Agents*) participating in this process are:
   - *Agriculture*: *Nutrient run-off* which participates in the water pollution process only if in high content. For values equal or smaller than Medium it participates in Plant growth process, as main food resource for any Plant species;
   - *Industry*: *Heavy metals*, which have the property of bioaccumulation in any aquatic biological entity, leading to that entity pollution, even Mortality if in Medium/High concentration in water.
2. The third water pollution component is given, mainly by *Cyanotoxins*. They are produced in water if there is a content of some poisoning species of Blue-green algae (Cyanobacteria), which contain Cyanotoxins in their cells.
3. To reduce the simulation complexity, Assumptions are introduced in the Scenario construction: "*Assume nutrient consumption is zero and steady*", "*Assume Migration is zero and steady*", and "*Assume Production is medium and steady*".
   - ¤ It concerns the modelled system's components, Entities and their associated quantities, participating in the chemical process of Water pollution and the Configurations among Entities, as follows:
   - ¤ The two system's Agents (External influences): Agriculture and Industry, developed "*In catchment area of*" the River, and participating in the system Water pollution with Nutrient run-off, and Heavy metals run-off, respectively;
   - ¤ The River, that "*Flows into*" its own River Delta, after collecting and transporting the pollutants, mainly Nutrients and Heavy metals, from its catchmnet area;
   - ¤ The River Delta system's inner components contributing to water pollution process: Nutrients, Heavy metals, Cyanotoxins, and POM bacterial decomposition.
   - ¤ Aquatic population that "*Lives in*" the River Delta.

Scenario Concerning Human Health and Water Quality

This scenario models the effects of increasing heavy metal and nutrient concentrations in the DDBR on health of humans living in and drinking water from the DDBR waters (Figure 6). Initial conditions can be seen in this

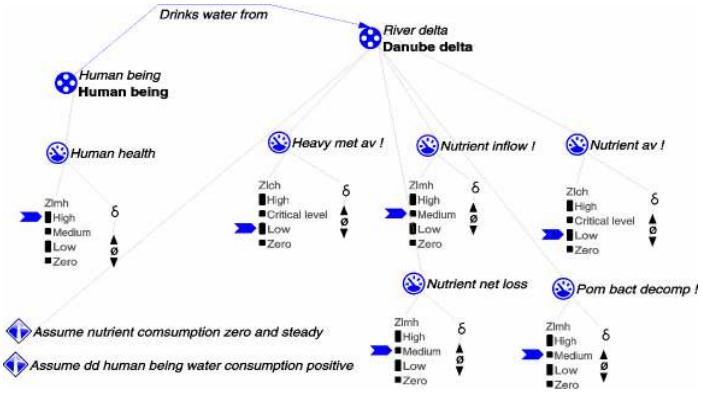


Figure 6. DD Human health influenced by DD water quality Scenario.

figure. Quantities related to nutrients and particulate organic matter are included only to satisfy certain static model fragments that require starting values for them, and are not of central focus for this scenario. Exclamation marks next to quantity names indicate exogenous behavior has been implemented (see Bredeweg et al. 2007). Hence, the amount of heavy metals in the environment (Heavy metals av) is set to low and exogenously increasing. This is meant to demonstrate the effects of increasing heavy metals on human health, after the factors contributing to increasing heavy metals (via runoff from industry) have been explored (see previous scenario). Other exogenous quantities set the respective quantity to remain steady (derivative = zero). Assumption labels implement behavior that is self-explanatory (Figure 6).

Model Fragments

There are three types of MFs: Static, Process, and Agent. Static MFs capture behavioral knowledge about the system. For DDBR aquatic ecosystems components there are 16 static MFs. A Process Model fragment defines the system behavioral characteristics related to a process. For DDBR system, there are 39 Process Model fragments. Two examples of Process Model Fragments are presented in Figure 7 and 8.

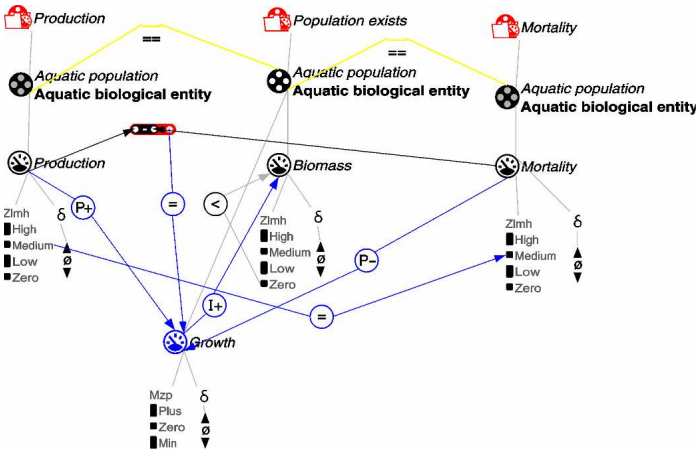


Figure 7. Aquatic Plant Growth - Process MF.

This MF (Figure 7) implements the causal dependencies (positive Influence I+, or positive/negative proportionality P+/P-) and the mathematical calculus (Minus, meaning difference between *Production* and *Mortality*) within any Aquatic population Growth process.

The MF shown in Figure 8 implements the structural and behavioral relationships between River delta and the Aquatic population, related to River delta: Water pollution influence on Aquatic population components behavior, as follows: positive influence (I+) on *Aquatic population*: *Mortality* and negative proportionality (P-) of *Aquatic population*:*Mortality* on *Aquatic population*: *Biodiversity*.

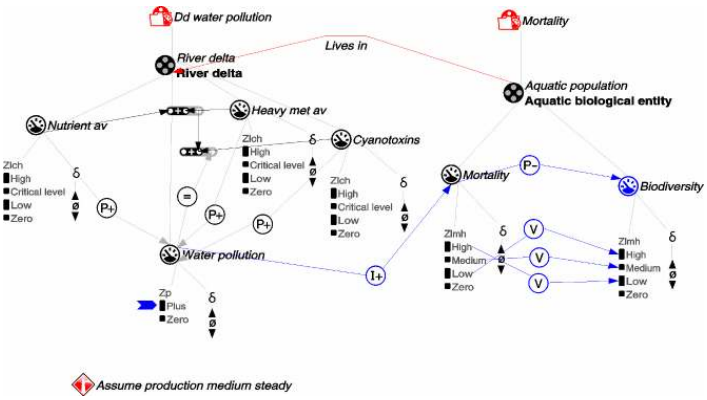


Figure 8. DD Water pollution and DD aquatic population biodiversity - Process MF.

Simulation Results

Scenarios' simulation results constitute the model output. This helps end users to understanding both the modeled system functional components causal relationships and the relevant factors affecting sustainability.

Simulation Results of Scenario Concerning the Water Pollution Process

There are presented simulation results for "DD Water pollution and DD aquatic population biodiversity Scenario" (see Figure 5). The most relevant results presented here are: Dependency diagram (Figure 9), Global State-graphs (Figure 10), and Global State-graphs and value history (Figure 11).

Dependency diagram

As the dependency diagram of any state in this simulation is very large, we present this diagram without quantity spaces (Figure 9). The diagram provides information on structure (entities, quantities, and configurations), causality (Influence I, or Proportionality P), and correspondence (Q, dQ) and in/equality (=, >, <) among the system's water pollutants (*Nutrients*, *Heavy metals* and *Cyanotoxins*), and any *Aquatic biological entity*:

Danube River: Nutrient inflow and Heavy Metals inflow main resources are the two system's external influences (Agents): Agriculture: Nutrient run-off and Industry: Heavy metals run -off, respectively, localized "In catchment area of" the River. There is a close relationship (P+, Q) between Nutrient run-off from Agriculture lands and Nutrient that enters the Danube River. The same relationship occurs between Heavy metals run-off from Industrial zones and Heavy metals that enter the Danube River. From the River, these two main water pollutants reach the Danube Delta aquatic ecosystems.

A part of *Danube Delta*: *Nutrient inflow* stays in the system and contributes to *Nutrient available for Plant*

species growth while another part is lost (*Nutrient net loss*), either through *Nutrient outflow* or *Nutrient consumption* (by aquatic *Plant* species only).

The same happens with *Danube Delta*: *Heavy metals inflow*. The only difference is that a part of the *Heavy metals inflow* is lost (*Heavy metals net loss*) as they are bioaccumulated within any *Aquatic biological entity* body both of *Plant* and *Animal* species.

A part of *Nutrient net loss* and *Heavy metals net loss* is recycled from dead organic matter as result of Particulate Organic matter bacterial decomposition (*Pom bact decomp*) process.

Danube delta: Water pollution rate is the result of three main water pollutants: *Danube delta: Nutrient available*, *Heavy metals available* and *Cyanotoxins*;

Danube delta: *Water pollution rate* has a direct positive influence (I+) on any *Aquatic biological entity*: *Mortality*. That signifies that a positive rate of Water pollution process induces an increase of *Mortality* for any Aquatic population.

Aquatic biological entity: Mortality has an indirect negative influence (P-) on any *Aquatic biological entity: Biomass*.

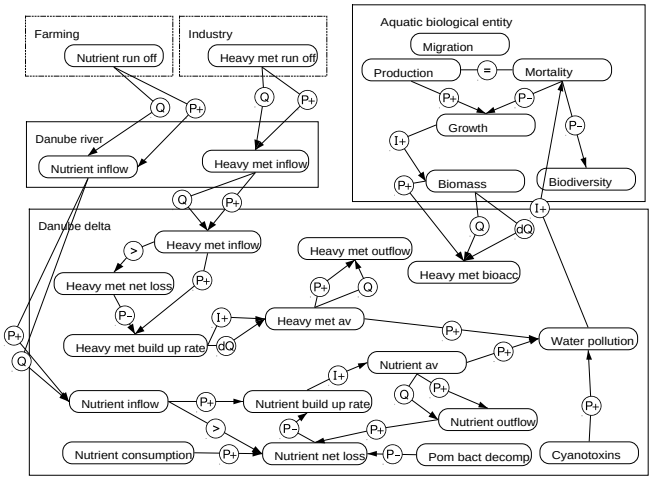


Figure 9. Dependency diagram (causal model) of the Water pollution process.

Global State-graphs

All states generated by full simulation, starting from the three initial states 1, 2, and 3 are presented in Figure 10. By full simulation of the initial states a total of 20 states are generated. Five of them are end-states: 12, 14, 15, 17, and 20.

Value history diagrams

Despite the many possible pathways resulting from the scenario, all of them show the same basic behavior with respect to heavy metals and biodiversity: heavy metals increase and biodiversity decreases (see right two value histories, Figure 11). Other paths differ only in behavior of nutrients, which may increase, decrease, remain constant, or exihibit a combination of these behaviors.

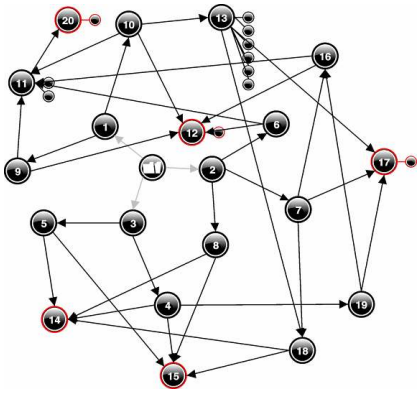


Figure 10. DD Water pollution and DD aquatic population biodiversity Global State-graph.

This is due to ambiguity in the relative magnitudes of inflow and outflow of nutrients (see end states, Figure 10 center). For example, states 15 and 20 define the end of the process, for two extreme conditions of the system as result of the *Danube Delta*: *Nutrient net loss*, High, + and Zero, 0, respectively.

Within the water pollution process related to *Aquatic biological entity* behavior, the *Aquatic biological entity*: *Biomass* never reaches the value Zero, because the *Growth* rate never reaches this value, as both the assumption and MF "*Growth on Migration only*", when Migration is assumed Zero/Steady, is not considered in this process. In these conditions, within most states, the following tendency happens:

1. Both *Aquatic biological entity*: *Biomass* and *Biodiversity* are Low, -;
2. *Aquatic biological entity*: *Growth* is Minus, -;
3. *Aquatic biological entity*: *Mortality* is High, +.

Simulation Results of Scenarios Concerning the Human Being Behavior from the Human Health Point of View

This simulation produces one possible beginning state, which gives rize to a total of 7 possible states and three behavioral paths (Figure 12). Each of these paths shows that as heavy metal and nutrient concentrations increase, human health decreases. Heavy metals increase because they were specified to increase due to an exogenous influence, whereas nutrients increase because nutrient inflow is greater than nutrient net loss (see Figure 13). The difference between the three paths arizes because of different relative rates of increase between heavy metals and nutrients.

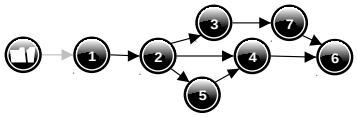
Dependency diagram

The Dependency diagram for each of the states in the simulation is similar (Figure 13). The diagram shows how

the model fragments have been composed into a complete causal model of the system specified by the scenario.

Figure 11. Value histories for beginning states (far left), end states (center left), and two behavioral paths (right) for the scenario concerning DDBR Water pollution and aquatic population biodiversity. →

It provides information on structural and causality (Influence I, or Proportionality P), correspondence (Q is a quantity space correspondence, dQ is a derivative correspondence, and V is a value correspondence) and In/equality (=, >, <) dependency relationships among the system's water pollutants (*Nutrient available* and *Heavy metals available*) and *Human being*: *Human health*.



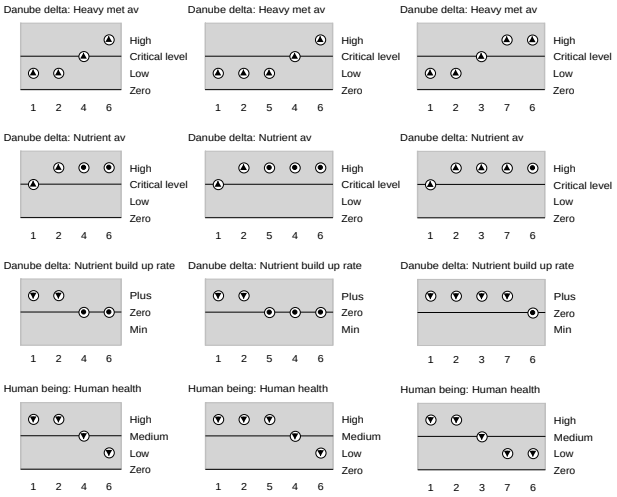
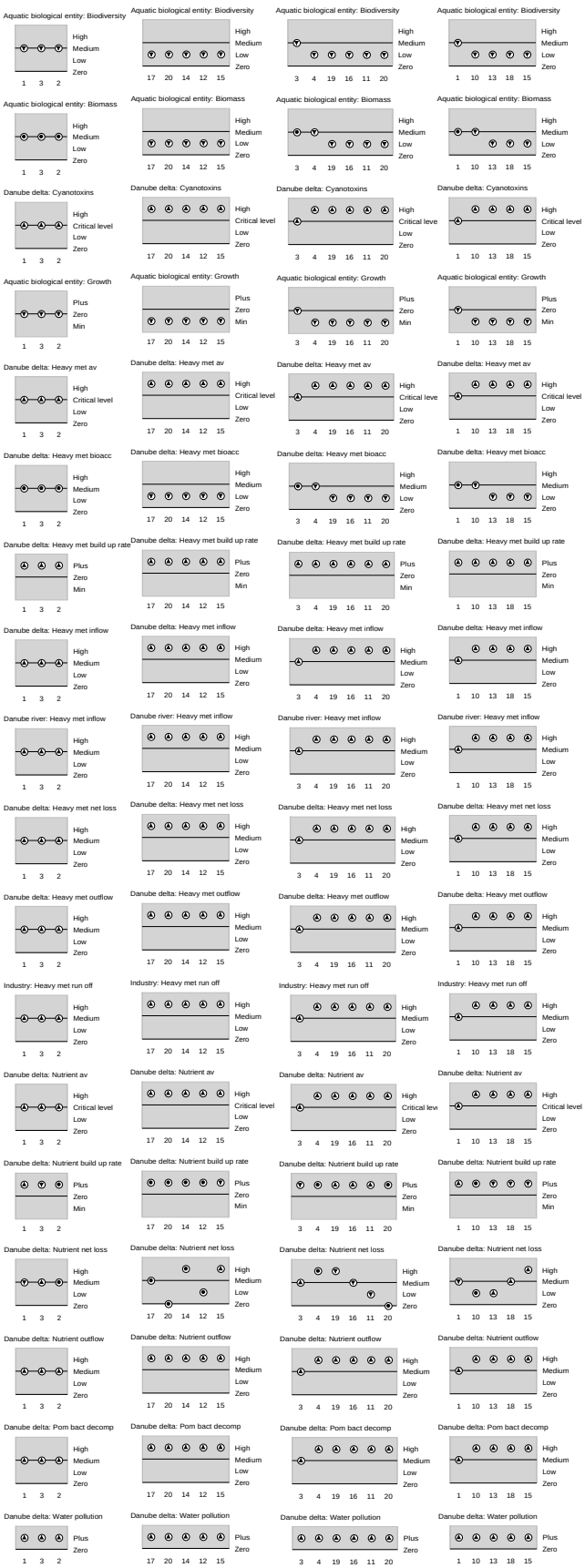


Figure 12. State graph and value histories for the three behavioral paths for scenario concerning human health and water quality. Quantities that were set to steady can be inferred from the scenario diagram (Figure 6).



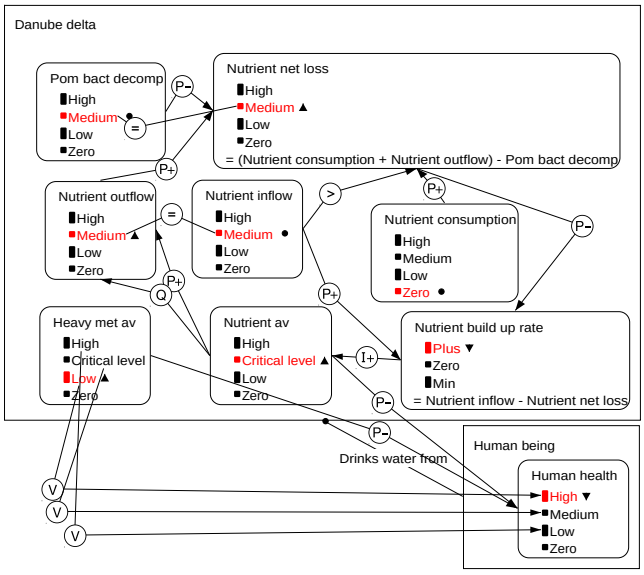


Figure 13. Dependency diagram for state 1 of the scenario concerning human health and water quality.

Discussion and Conclusions

This paper contains a description of some of the aspects of the DDBR Garp3 model, emphasizing the main causes and their effects that challenge achievement of Sustainable Development within the DDBR. The DDBR Qualitative Reasoning Model Fragments emphasize the causality conditions, which have been generating loss of DDBR biodiversity, aiming to delimit those objectives for a sustainable use of natural resources and a Sustainable Development Strategy addressing the aquatic ecosystems. Conservation and protection of biodiversity is one of the main objectives in achieving the Sustainable Development Strategy for the DDBR. These must be based on the best current understanding of the phenomena, which occur within and beyond the DDBR, including the whole hydrographic basin of the Danube River and the Western Black Sea coastal waters. Toward this aim, knowledge about the aquatic ecosystems behavior within DDBR system, as it is presented in the DDBR QR Model, serves for making decisions for biodiversity conservation and protection measures.

Ongoing work with this model serves to optimize the model-fragment representations to make the representations most insightful and capitalize on inheritance in the entity hierarchy more effectively. This will reduce the need to include extraneous quantities in some scenarios, while making the knowledge representation more transportable. Also, we are working to manage ambiguity through development of appropriate simplifying assumptions.

This model and accompanying documentation aimed at producing educational materials to teach about concepts of SD.

References

Bredeweg, B., Bouwer, A., Liem, J., 2006 – Garp3- *Workbench for capturing conceptual knowledge*. NaturNet-Redime NEWSLETTER 2: 2-4.

Brundtland, G.H., and the World Commission on Environment and Development. 1987. Our Common Future. Oxford University Press, Oxford, UK.

Cioaca, E., Nuttle, T. and Bredeweg, B. 2006 – The Danube Delta Biosphere Reserve: a case study for qualitative modelling of sustainable development. In *Proceedings of the QR 2006: 20th International Workshop on Qualitative Reasoning,* 151-155. Hanover, New Hampshire, USA.

Cioaca, E., Bredeweg, B. and Salles, P. 2007 – *Danube Delta Biosphere Reserve Case Study Qualitative Reasoning Model - an Environmental System Sustainable Development Decision Making Support Tool*. NaturNet-Redime NEWSLETTER 5: 9-12.

Cunningham, W. P., and Cunningham. M.A. 2005. *Principles of Environmental Science. in*. McGraw-Hill online.http://highered.mcgraw-hill.com/sites/0072452706/student_view0/glossary.html

European Commission. 2001. *Communication from the European Commission: a Sustainable Europe for a Better World: A European Union Strategy for Sustainable Development*. Brussels, Belgium.

Nuttle, T., Salles, P., Bouwer, A., Bredeweg, B., Caldas, A. L., Cioaca, E., Covaliov, S., Liem, J., Nakova, E., M. Neumann, R. Noble, E. Varadinova, Y. Uzunov, and A. Zitek. 2006 – Navigating a landscape of sustainability concepts: towards progressive learning routes within and among qualitative reasoning models. *In Proceedings of the 5th ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, part of the 17th European Conference on Artificial Intelligence*, 2-1 – 2-10. Riva del Garda, Italy.

Oosterberg, W., Buijse, A.D., Coops, H., Ibelings, B.W., Menting, G.A.M., Staras, M., Bogdan, L., Constantinescu, A., Hanganu, J., Navodaru, I., and Török, L. 2000. *Ecological Gradients in the Danube Delta Lakes - Present State and Man-induced Changes,* Technical Report, RIZA rapport 2000.015, ISBN 0.369.5309x, Institute for Inland Water Management and Waste Water Treatment, the Netherlands, and the Danube Delta National Institute, Romania.

Otel, V., and Ciocarlan, V. (eds.) 2000. *The Red List of Plant and Animal species from the Danube Delta Biosphere Reserve.* Reading, Mass.: Aves Foundation, Romania.

Behaviour Prioritisation in Fuzzy Qualitative Simulation.

George Macleod Coghill
Department of Computing Science, Unviersity of Aberdeen AB24 3UE, Scotland, UK.
e-mail: gcoghill@csd.abdn.ac.uk

Abstract

Fuzzy qualitative simulation combines the features of qualitative simulation and fuzzy reasoning in order to gain advantages from both. However, the output of a fuzzy qualitative simulation process is a behaviour tree which for complex systems will be large. In order to overcome this and permit focussing on preferred behaviours *priortisation* was developed. In this paper a new prioritisation scheme is presented that makes use of both constraint and temporal information to perform the prioritisation.

Keywords: Fuzzy Qualitative Reasoning, Prioritisation

1 Introduction

One of the original motivations for the development of Qualitative Reasoning (QR) systems was a research programme to enable expert systems to reason from first principles, in order to overcome perceived weaknesses inherent in the first generation, rule-based, expert systems [7]. QR gives a broad picture of the way in which a system can behave and it was not long before the engineering community became interested in, and contributed to the field, because it was seen as a useful tool for simulating the behaviour of complex but incompletely specified plant. These influences have contributed to the utilisation of semi-quantitative information [1].

On the other hand Fuzzy systems were also developed to overcome some limitations in rule-based systems, by extending them to handle approximate knowledge. However, whereas QR deals with incomplete structural models, Fuzzy systems have tended to deal with input/output type problems. This has not been exclusively the case though, and fuzzy sets have been combined with interval simulators to carry out fuzzy interval simulations [2]. However, as with normal interval simulation the goal has been to generate narrowly focussed *unique* behaviours.

This situation led to the development of systems that combined the features of qualitative reasoning with those of fuzzy systems [9, 3]. There are at least three advantages which ensue from the combination of fuzzy and qualitative approaches [9]:

- the fact that the meaning of a qualitative value and its support set (the real number line here) are captured in a single representation,
- the ability to incorporate empirical knowledge into a model (which is also finer grained than the $M^{+/-}$ constraint utilised in QSIM [7]), and
- being able to include more detailed knowledge of the temporal behaviour of the variables in a model than the total ordering available within QSIM, which is useful for use in such applications as model-based diagnosis and control.

This was the motivation behind the development of FuSim [9], which is the system which was the major influence on the development of *Morven*. However, QR systems, regardless of complexion, generate behaviours that are not unique; and in the case of complex systems may be prohibitively large. In order to ameliorate this steps have been taken to assign a priority to the behaviours in a fuzzy behaviour tree [8]. It is the analsysis and development of such prioritisation schemes that is the subject of this paper.

The structure of this paper is as follows. In the next section the *Morven*[1] toolset is summarised, in order to put the subject of this paper in its overall context. In section 2 the details of the original prioritisation scheme are presented

1. This system was previously known as *Mycroft*, but I discovered that this name was far from unique so I changed it to the name of my elder daughter.

and criticised; and this is followed (in section 4) by the description of an alternative and extended approach to behaviour prioritisation. In section 5 an illustrative example is described and analysed, and from this some relevant conclusions drawn.

2 The *Morven* Toolset

The *Morven* toolset is a constraint-based fuzzy qualitative reasoning system containing a number of simulation and envisionment algorithms. The development of this toolset has permitted the suitability of different techniques to be examined in a number of contexts; and the comparison of different approaches to constraint-based fuzzy qualitative simulation to be made [3].

2.1 Representation and Inference

Fuzzy sets extend the ideas of traditional set theory to include the concept of partial (or graded) membership. It is assumed here that the ideas underlying fuzzy sets are known to the reader; however, a description of the domain and explanation of the concepts may be found in [6]. In FuSim, for reasons of computational efficiency, trapezoidal fuzzy numbers and intervals are used. An example of such a fuzzy number, a parameterised four tuple, is shown in Figure 1. This fuzzy number meets the criteria for being a fuzzy set in that it has a graded membership. The set is described by its membership function, $\mu_a(x)$, which is described by the four tuple [a, b, α, β] and is defined as:

$$\mu_a(x) = \begin{cases} 0 & x < a - \alpha \\ \alpha^{-1}(x - a + \alpha) & x \in [a - \alpha \quad a] \\ 1 & x \in [a \quad b] \\ \beta^{-1}(b + \beta - x) & x \in [b \quad b + \beta] \\ 0 & x > b + \beta \end{cases}$$

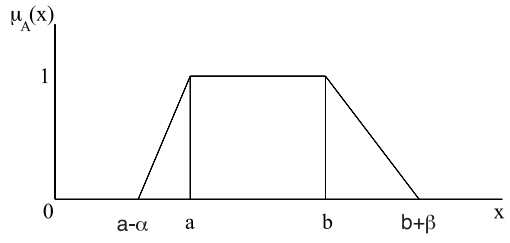


Figure 1: A Trapezoidal Fuzzy Number

The quantity space which is built from fuzzy numbers must be closed, continuous, finite and cover all values which a variable can take. An example of such a quantity space is shown in Figure 2. It is also possible to turn a trapeziodal fuzzy interval into a crisp interval by means of α-cuts [9]. Here a particular membership value (α) is chosen as representing "typicality" with respect to the fuzzy quantity, or quantity space, of interest; and then these 'typical' values can be used in the reasoning process as was done in FuSim, and carried over into *Morven*. Of course, what constitutes 'typical' can be altered by selecting different values for α. In fuzzy qualitative simulation, unlike QSIM, the quantity space for the derivatives of a variable may also be dense (that is, can have any number of divisions consistent with the definition of a quantity space).

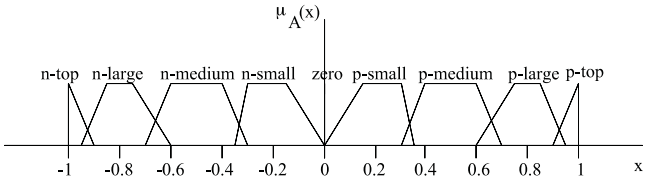


Figure 2: A Fuzzy Quantity Space

The *Morven* toolset is a qualitative reasoning system within the so called constraint based ontology. The models used in the toolset consist of sets of variables and the constraints that relate them. In fuzzy qualitative reasoning the operators utilised are the same as for its symbolic counterpart, though by the nature of the case there is a difference in the way they are implemented. All the variables of the system take their values from a predefined fuzzy quantity space.

In *Morven* the model constraints are causally ordered [5] and distributed over a number of differential planes [10]. That is, the qualitative differential equation (*qde*) model is developed on plane-0 (the zeroth differential plane), and the relationships between the higher derivatives of the system are obtained by differentiating the *qde* and representing the results as a *qde* on the so called higher differential planes.

Finally, as with QSIM and FuSim, *Morven* represents variables as a vector consisting of the fuzzy qualitative value for the magnitude and derivatives of the variable as a function of time. However, whereas QSIM and FuSim only make use of vectors of length 2 (representing the magnitude and direction of change of the variable), *Morven* can use vectors of any required length. For practical purposes three is the most that is generally required for exogenous variables, per-

mitting a description of the *curvature* of the function. For any model of a state system there is a relationship between the number of differential planes in the model and the length of the vectors describing the state variables: for a model containing n differential planes, the state variable vector will contain $n+1$ elements. For example, in the single tank case above, the state variable, h, will require a vector of length 3:

$$h = [d^0 d^1 d^2]$$

where d^0, d^1 and d^2 are the zeroth, first and second derivatives of the variable respectively (or, if preferred: the magnitude, direction and curvature). When applied to an actual system variable, these elements are referred to as variable-vector elements

As stated previously, the *Morven* toolset consists of a number of simulation and envisionment algorithms [3, 4]. However, all these algorithms, in common with qualitative reasoning approaches in general, divide the inference process into two phases: Qualitative Analysis (QA) and Transition Analysis (TA).

In the QA phase the equations of the system model are solved and qualitative states generated. In the TA phase the values of the magnitudes and derivatives of, at least, the state variables of the system are known, and this information is used (along with transition rules) to decide which values these variables may take in the succeeding time interval (or time point).

2.2 Temporal Calculations

One of the motivations for combining qualitative reasoning with fuzzy reasoning was the provision of fuller temporal information to be included in the qualitative reasoning process. There are four temporal calculations that are important in *Morven*: the *persistence time*, the *relative arrival time*, the *absolute arrival time*, and the *absolute departure time*.

Persistence time, ΔT_p: This is a time interval representing the endpoint of the interval during which an element of a variable vector remains in the same state, assuming that the calculation is made from the time the variable entered that state, and the derivative used in its calculation does not change during the time interval. The persistence time for an element of a variable vector, d^n, is defined as:

$$\text{If } 0 \notin d_\alpha^{n+1}, \text{ then } \Delta T_p \in \frac{W(d^n)}{|d^{n+1}|_\alpha}$$

where d^n and d^{n+1} are the nth and $(n+1)$th derivatives of the variable (for the purposes of these calculations the magnitude of the variable is considered as the 0th derivative). $W(d^n)$ is the α-width of the fuzzy interval of the nth derivative. This formula is the same as that used in FuSim.

Relative Arrival time, ΔT_a: This is the time interval representing the length of time it takes for a variable-vector element to transit from one state to another. Consider the case where d_j^n and d_{j+1}^n are jth and $(j+1)$th quantities from the quantity spaces of the nth derivative of the variable under consideration (likewise, d_j^n and d_{j+1}^{n+1} are jth and $(j+1)$th quantities from the quantity spaces of the $(n+1)$th derivative of that variable.). Then the relative arrival time for an element of a variable vector, d^n, used in a *Morven* simulation is given as:

$$\text{If } 0 \notin d_\alpha^{n+1}, \text{ then } \Delta T_a \in \frac{L[d_{j+1}^n]_\alpha - U[d_j^n]_\alpha}{|d^{n+1}|_\alpha}$$

where, the $L[\cdot]_\alpha$ and $U[\cdot]_\alpha$ are the lower and upper bounds of the respective α-intervals. If the nth element does not transit then the relative arrival time is zero. However, depending on whether the $(n+1)$th element transits, the value of $|d^{n+1}|_\alpha$ will be different. On the assumption that the present value of the $(n+1)$th element is d_j^{n+1}, if this element does not transit then $|d^{n+1}|_\alpha = |d_j^{n+1}|_\alpha$. However, if there is a transition then:

$$|d^{n+1}|_\alpha = |d_{j+1}^{n+1}|_\alpha - |d_j^{n+1}|_\alpha$$

Absolute Arrival time, T_A: This is the time interval representing the length of time it takes a variable vector element, d^n, to arrive at a particular state from the initial time of the simulation ($t = 0$). The formula for this is:

$$T_{A_n} = \sum_{i=0}^{i-1} \Delta T_{p_i}(d^n) + \sum_{j=1}^{n} \Delta T_{a_j}(d^n)$$

where n is the nth absolute time index.

Absolute Departure time, T_D: This is the time interval representing the length of time it takes a variable vector element to depart from a particular state, with the initial time of the simulation as the datum. The formula for the absolute departure time is:

$$T_{D_n} = T_{A_n} + \Delta T_{p_n}$$

There is an exception to this formula if the absolute time contains a transition of the (n+1)th element of the variable-vector. The expression utilised in such cases is:

$$T_{D_{n+1}} = T_{D_n} \Xi \Delta T_{p_n} + \Delta T_{p_{n,n+1}}$$

where, Ξ is a retraction operator, and $\Delta T_{p_{n,n+1}}$ is a pseudo-persistence time in which the $(n+1)$th element used in the calculation is the difference between the two quantities appearing in the transition of the $(n+1)$th element.

3 Prioritisation

Qualitative simulation does not result in a unique behaviour; rather it generates a tree of behaviours. In symbollic qualitative simulation the maximum branching factor is 4, whereas in fuzzy qualitative simulation it is 6. This means that for a complex system, even ignoring the problem of spurious behaviour generation, the behaviour tree may be large.

In response to these results Leitch and Shen [8] developed a scheme for prioritising the states and behaviours generated in a simulation, on the basis of a distance metric, in order to find the best plausible approximate behaviour. In this section the details of the particular distance metric utilised by FuSim is given. This is followed by a description of the method used to prioritise the states generated by FuSim. Finally the technique for prioritising the behaviours in a FuSim behaviour tree is assessed.

3.1 A Distance Metric

To understand the distance metric introduced by Leitch & Shen (which is also utilised by *Morven*) and how it is used, it is necessary to give a more detailed example of two kinds of value which are used in a simulation: the *predicted value* and the *propagated value*. The former is the value (or set of values) which a variable is assigned on the basis of the transition rules; that is they are the values predicted by the integration phase of the simulation. Each system constraint consists of a constrained variable, which is the variable appearing on the left hand side of a constraint, and one or more constraining variables which appear on the right hand side of a constraint and may be used to calculate a value for the constrained variable. The propagated value of a variable is the value thus calculated.

Consider a system consisting of the following three place constraint:

$$a = b + c$$

If the constraining variables b and c have the α-cut interval values [1 4] and [5 8] respectively, then the propagated value for a will be [6 12], as shown in Figure 3.

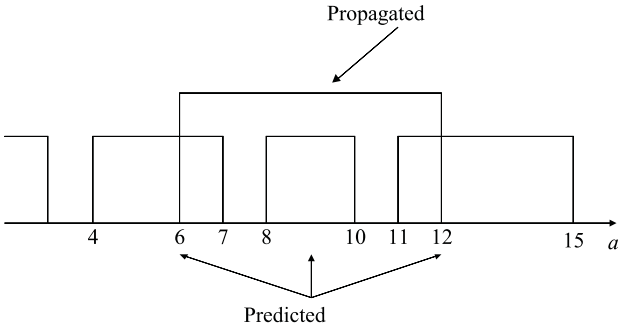


Figure 3: Propagated and predicted values

It is possible that the predicted and propagated values are identical; however, it is usually the case that a propagated value will intersect with several predicted values, as shown in Figure 3. One can treat each predicted value that intersects with the propagated value as equally possible. However, since not all the members of a variables quantity space will intersect the propagated value to the same degree the method of prioritisation was developed to reflect this fact and give those quantities which are a closer approximation to the propagated value a higher priority. The measure of which quantity is the best approximation is gained by means of the distance metric.

This distance metric (given the symbol d) is really a measure of similarity between two fuzzy numbers. Let the propagated value be depicted by a normal capital letter and the predicted value by a capital letter with a circumflex above it; then the formula for the distance between the two values given in [8] is:

$$d(A, \hat{A}) = [(power(A) - power(\hat{A}))^2 + (centre(A) - centre(\hat{A}))^2]^{\frac{1}{2}}$$

where, for 4-tuple parametric fuzzy numbers:

$power([a, b, \alpha, \beta]) = \frac{1}{2}[2(b - a) + \alpha + \beta]$

$centre([a, b, \alpha, \beta]) = \frac{1}{2}[b + a]$

These formulae represent the area of the fuzzy interval and the centre of the nucleus of the fuzzy interval respectively. If two fuzzy intervals are identical, then according to the above expressions they will have a distance of zero. Thus, the smaller the distance between a propagated and predicted value, the better the approximation.

As an example, consider again the situation

depicted in Figure 3. Call the propagated value $\hat{a}$, and the predicted values $a1, a2$ and $a3$ with values [4 7], [8 10] and [11 15] respectively. Then
$d(a1, \hat{a}) = [(3-6)^2 + (5.5-9)^2]^{0.5} = 4.6$
$d(a2, \hat{a}) = [(2-6)^2 + (9-9)^2]^{0.5} = 2$
$d(a3, \hat{a}) = [(4-6)^2 + (13-9)^2]^{0.5} = 5.7$
from which it can be seen that the value which must be assigned the top priority is $a2$, (followed by $a1$ and finally $a3$).

3.2 State Prioritisation

A system would not normally consist of only one constraint; therefore the states which are to be prioritised for any step ahead in the simulation will be made up of values for a number of different variables which are consistent with several different constraints in the system. Thus a method of prioritisation is required which will order complete system states. The approach suggested by Shen and Leitch deals with this task in two stages. The first stage provides the distance for a complete predicted variable value consisting of a magnitude and derivative $< A, B >$ from the equivalent propagated state $< \hat{A}, \hat{B} >$. Each element of the variable will have an associated distance $d(.)$; the distance for the complete variables, $D(.)$, is:

$$D(< A, B >< \hat{A}, \hat{B} >) = max\{d(A, \hat{A}), d(B, \hat{B})\}$$

The second stage is to find a distance for each complete system state from the distances for each complete variable. In their paper, Shen and Leitch suggest the following formula:

Prioritise the states such that $\rho((A_i, B_i)) = j$, $i = 1, 2, \ldots, M$, if

$$D_i = min\{\{D_k | k = 1, 2, \ldots, M\} - \{D_k | k < j\}\}$$

That is: for each constrained variable in the constraints, the constraint is applied to both the magnitude and derivative of the variable, and hence a distance between propagated and predicted values can be found for both the magnitude and derivative of the variable. The overall distance for the variable is then taken to be the maximum of these distances. Having obtained distances for each variable value in the state, the distance for the particular state is taken to be the minimum of these distances. Then the states are prioritised in accordance with these distances, from minimum to maximum.

The above approach provides an ordering of the states at each step in the simulation. However, this method effectively makes a single constraint responsible for the priority assigned to the state. This is because the magnitude and derivative distances calculated for a single pair are assigned from the application of a single constraint, and the maximum chosen. Then the ordering is performed by selecting the minimum distance for these maxima, which is the same as selecting the constraint that produced this *maxmin* value. This approach then, while providing an excellent start in the application of prioritisation to fuzzy qualitative simulation, does not utilise most of the information available about the model structure. Therefore an alternative approach to the prioritisation of the system behaviours is discussed in Section 4.

3.3 Behaviour Prioritisation

The prioritisation dealt with in the preceding section is associated with the QA phase of the simulation. This is the analysis which deals with the values of states at instants in time. However, what is important about simulations are the behaviours; therefore it is essential that the behaviours generated by the qualitative simulator be prioritised, rather than the states. Leitch and Shen also address this problem. However, their solution is to examine the distances and priorities of the states at each step of the simulation and base the estimate of which behaviour should be top priority on the combination of the distances calculated at each step. This amounts to being a depth first search through the tree, seeking to find a path that minimises some cost. Again, this solution can be criticised methodologically. It implicitly assumes that the QA phase is the only important part of the simulation and that there is no cost (or at least equal cost) in transiting from one state to another. In contrast to this it can be argued that since simulation comprises a TA phase as well as a QA phase, behaviour prioritisation should be based on a combination of the constraint prioritisation and a temporal prioritiser which can estimate the most likely transition.

4 An Alternative Approach

On the basis of the distances calculated between the predicted and propagated values, the predicted values can each be assigned a priority. For every arithmetic constraint there will therefore be (at least) one predicted value which has the top priority. Since the model consists of a conjunction of constraints the top priority state will be the one consisting of the conjunction of the top priority predicted value for each constraint.

4.1 Constraint Prioritisation

In *Morven* variables and models have complementary representations: variables are represented as vectors that are qualitative representations of Taylor series and the models exist in differential planes. In a Taylor series the higher derivatives have less weight in the calculation of the variables value at a future time; likewise in the present situation the higher derivatives carry less weight in the calculation of the priority of a state. Also, since *Morven* models tend to be causally ordered for a simulation the values predicted from constraints later in the list are dependant on the predicted values of variable-vector elements calculated earlier in the constraints list. Thus, the state priority decreases as the priorities of the predicted values of the variable-vector elements decrease, from the lowest derivative upward to the magnitude and from variable-vector elements calculated later to those calculated earlier in the constraints. For example, the second priority state will be the one with every predicted value having top priority, except the constrained variable of the final constraint of the system which will have second top priority. Obviously then, the lowest priority state is the one in which all the predicted values are of the lowest priority. To clarify this, consider the following pseudo example. A causally ordered model consists of three constraints:

$a = f(x)$, $b = f(a)$, $c = f(b)$

x is given and a, b and c are the variables which need to be calculated. After application of the constraints the variables have the following values:

$$a = \{qa1{:}1,\ qa2{:}2\}$$
$$b = \{qb1{:}2,\ qb2{:}1\}$$
$$c = \{qc1{:}2,\ qc2{:}1\}$$

where the qs are quantities from the appropriate quantity space and the numerals after the colons are the priorities assigned to the quantities by the constraint. Thus the eight states created with these values will be prioritised as in the order of the following list (from highest priority to lowest):
$\{(qa1, qb2, qc2), (qa1, qb2, qc1), (qa1, qb1, qc2),$
$(qa1, qb1, qc1), (qa2, qb2, qc2), (qa2, qb2, qc1),$
$(qa2, qb1, qc2), (qa2, qb1, qc1)\}$
which if looked at as a sets of conjunctions of priorities would have the following form:
$\{(1 \wedge 1 \wedge 1), (1 \wedge 1 \wedge 2), (1 \wedge 2 \wedge 1), (1 \wedge 2 \wedge 2),$
$(2 \wedge 1 \wedge 1), (2 \wedge 1 \wedge 2), (2 \wedge 2 \wedge 1), (2 \wedge 2 \wedge 2)\}$
which gives a total ordering of the priorities for a given model.

4.2 Temporal Prioritisation

In Section 2.2 the different time calculations performed by *Morven* were described, and two of them - the *absolute departure time* and the *absolute arrival time* are relevant here since it is the ongoing simulation that is being dealt with. These times give a measure of the time elapsed since the beginning of the simulation till the variable-vector element either departs from its present state, or arrives in its next state. These times are both intervals representing the earliest and latest times a departure or arrival could take place. The possible transitions that take place may be ordered, and thus prioritised, on the basis of these absolute times, from fastest to slowest.

By combining constraint and temporal prioritisation one obtains an ordering of the behaviours in the behaviour tree: the behavioural prioritisation

5 An Illustrative Example

In this section an example is presented to illustrate the concepts discussed in the previous sections. The coupled tanks system is chosen because it is complex enough to explicate the concepts and simple enough to be understood and analysed. A schematic diagram of the coupled tanks system is shown in Figure 4.

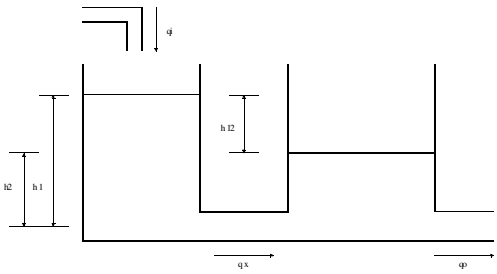


Figure 4: A Coupled Tanks System

Fuzzy qualitative reasoning permits the incorporation of empirical knowledge in a model in the form of fuzzy rules. The exact form in this example was chosen to emphasise the advantages of fuzzy qualitative reasoning in this respect, whilst keeping the problem tractable for ease of analsysis and explanation. The constitutive relationships of the coupled tanks system are fuzzy rule-bases of the following form:

$q_x = F(\Delta h)$	nt	nl	nm	ns	z	ps	pm	pl	pt
nt	1	0	0	0	0	0	0	0	0
nl	0	1	0	0	0	0	0	0	0
nm	0	0	1	0	0	0	0	0	0
ns	0	0	0	1	0	0	0	0	0
z	0	0	0	0	1	0	0	0	0
ps	0	0	0	0	0	1	0	0	0
pm	0	0	0	0	0	0	1	0	0
pl	0	0	0	0	0	0	0	1	0
pt	0	0	0	0	0	0	0	0	1

$q_o = F(h_2)$	nt	nl	nm	ns	z	ps	pm	pl	pt
nt	1	0	0	0	0	0	0	0	0
nl	0	1	0	0	0	0	0	0	0
nm	0	0	1	0	0	0	0	0	0
ns	0	0	0	1	0	0	0	0	0
z	0	0	0	0	1	0	0	0	0
ps	0	0	0	0	0	1	0	0	0
pm	0	0	0	0	0	0	1	0	0
pl	0	0	0	0	0	0	0	1	0
pt	0	0	0	0	0	0	0	0	1

and the structural relations are:

$$\Delta h = h_1 - h_2 \quad h_1' = q_i - q_x \quad h_2' = q_x - q_o$$

where h_1 and h_2 are the head of fluid in the tanks, Δh is the head difference, q_i is the rate fluid flow into the system, q_o is the flow of fluid from the system and q_x is the crossflow of fluid between the tanks.

These constraints are causally ordered for use with *Morven*. They also constitute the constraints of the zeroth differential plane; the constraints for the higher differential planes are obtained by differentiating these constraints. Of course, empirically derived rule bases cannot be differentiated and so this will only be achievable if empirical information has been obtained for the first differential plane as well. For the purpose of this example it is assumed that this has been done.[2]

5.1 Results and Discussion

In order to run a simulation, two pieces of information are required: the initial values of the states variables (h_1 and h_2), and a complete specification of the input (or exogenous) variable, q_i in this case. Consider the situation where there is a continuous steady flow into the tanks (which are initially empty) with value $p - medium$; this gives the following input description and initial values.

Input: $q_i =< p - medium\ zero >$

Initial Values: $\begin{cases} h_1 = zero \\ h_2 = zero \end{cases}$

The result of this experiment is a behaviour tree containing 550 states. From this, one can select a number of paths to steady state that serve to illustrate the features of behaviour prioritisation. Three paths will be examined. A behaviour tree is also known as a partial envisionemnt because it constitutes that part of a complete envisionment graph reachable from a given initial starting state. In a behaviour tree the same state can appear in a number of different branches with a different state number and this can make it harder to discern what is going on when analysing the effects of prioritisation. In order to make things clearer, the same states for the three paths have been given the same label and the part envisionment graph is shown in Figure 5.

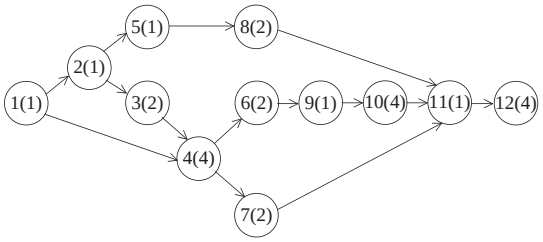


Figure 5: Partial behaviour graph with constraint prioritisation applied

This part envisionment starts with both tanks empty (state 1) and ends at an equilibrium state (state 12). Three states are common to all three paths (states 1, 11 and 12). The numbers in parentheses on the graph refer to the constraint priority of the states.

In order to demonstrate the effectiveness of the prioritisation scheme, the three paths chosen are: the fastest path (containing states: 1, 2, 5, 8, 11 and 12) taking a maximum time of 12.17 time units (tu); the shortest path (containing states: 1, 4, 7, 11, 12) taking a maximum time of 12.83tu; and a long path (containing states: 1, 2 3, 4, 6, 9, 10, 11 and 12) with a maximum time of 13.83tu.

Several things emerge perusal of Figure 5:

- The fastest path is also the one with the highest overall constraint priority. The exception to this is priority of state 8. However the top priority states at that level did not form any path to equilibrium and may therefore part of a spurious behaviour.
- The shortest path in terms of the number of qualitative states traversed is not necessarilly the fastest. This can occur because some states persist longer than others; therefore, paths including them will be temporally longer than the other paths.
- Temporal prioritisation must be done globally rather than locally. This arises from

2. *Morven* can operate using only the zeroth differential plane, but for purposes of comparison two planes are used for this example.

the fact that the transition from state 2 to state 3 is faster than the transition state 2 to state 5; yet the former is part of the longest path and the latter is part of the fastest path overall. This is related to the problem of spurious behaviour generation, where it is well recognised that one of the sources of this problem is the local nature of the TA phase.

- The existence of different rational paths to equilibrium indicate that prioritisation is a means of behaviour selection. The constitutive relations in a fuzzy system contain vagueness and depending on the degree may allow the incorporation of a wide range of underlying parameter values. Here prioritisation acts as an analogue to sampling, enabling one to select either faster or slower changing behaviours as required.

Finally, on the assumption that an appropriate prioritisation scheme can be developed, the question arises as to how it should be integrated with existing filters. The process of prioritisation orders a set according to a metric. It does not actually allow one to state that any of the states should be eliminated. Implicit in this is the assumption that all the members of the set being prioritised are real. This leads to the conclusion that the existing methods for eliminating spurious behaviours [7] should still be used to eliminate as many behaviours as possible, and then (on the assumption that the remaining states are valid) prioritisation should be applied to select the most likely.

6 Conclusions

The theme of this paper has been the exploration of issues relating to prioritisation of the behaviours generated by fuzzy qualitative simulation. The possibility of selecting behaviours in this way was first suggested by Leitch and Shen [8], and they provided a useful initial method. However, their approach was based on *max* and *min* operators and did not utilize all the information available in making the decisions regarding which behaviour should be considered to have top priority. Therefore a new approach to prioritisation is suggested based on the recognition that constraint based models are conjunctions of constraints and the vectors representing the values of variables and qualitative versions of Taylor series. The resulting behavioiur prioritisation utilises both constraint and temporal prioritisation and is therefore a more versatile and informative version of prioritisation.

The conclusions from the experimental analysis of using this new prioritisation scheme are:

- The fastest path is also the one with the highest overall constraint priority.
- The shortest path may not be the fastest.
- Temporal prioritisation must be done globally rather than locally.
- The existence of different rational paths to equilibrium indicate that prioritisation is a qualitative analogue to sampling.

References

[1] Berleant D. & Kuipers B. J. 1997. "Qualitative and Quantitative Simulation: Bridging the Gap." *Artificial Intelligence* 75: 215–255.

[2] Bonarini A. & Bontempi G., "A qualitative simulation approach for fuzzy dynamical models", ACM Transactions on Modeling and Computer Simulation, 4(4), 285-313, 1994.

[3] Coghill G. M. "*Mycroft:* a Framework for Constraint Based Fuzzy Qualitative Reasoning", Ph.D. Thesis, Heriot-Watt University, September 1996.

[4] Coghill G. M. "On the development of Fuzzy Envisionment" Proceedings of the 3rd International Workshop on Hybrid Methods for Adaptive Systems, Oulu Finland, July 10-11, 2003.

[5] Iwasaki Y., "Model based reasoning of device behavior with causal ordering", Ph.D. Thesis, Carnegie Mellon University, August 1988.

[6] Kosko B., *Fuzzy Engineering*, Prentice-Hall International, 1997.

[7] Kuipers B. J., *Qualitative Reasoning: Modelling and Simulation with Incomplete Knowledge*, MIT Press, London, 1994.

[8] Leitch R. R. & Shen Q., "Prioritising behaviours in qualitative simulation", in Proceedings of the 13th International Joint Conference on Artificial Intelligence, Pp 1523-1528, Chambercy, France, 28th August - 3rd September, 1993.

[9] Shen Q. & Leitch R. R., "Fuzzy qualitative simulation", IEEE Transactions on Systems, Man and Cybernetics, 23, 1038 - 1061, 1993.

[10] Wiegand M. E., "Constructive Qualitative Simulation of Continuous Dynamic Systems", Ph.D. Thesis, Heriot-Watt University, May 1991.

Capturing and Categorizing Mental Models of Food Webs using QCM

Morteza Dehghani[1], Sara Unsworth[2], Andrew Lovett[1], Kenneth D. Forbus[1]

{morteza, s-unsworth, andrew-lovett, forbus}@northwestern.edu
[1]Qualitative Reasoning Group, Northwestern University
[2]Doug Medin's Research Group, Northwestern University

Abstract

This paper examines the use of qualitative representations in modeling the similarities and differences in causal reasoning for biological kinds between Menominee Native Americans and US majority culture. Qualitative Concept Maps are used for modeling and analyzing transcripts of interviews conducted with these groups. The individual models are used to construct generalizations for the groups, which are tested both by inspection and by creating a classifier to distinguish models from these two cultures.

Introduction

Qualitative modeling could become an important tool for cognitive science, by providing formal languages for expressing human mental models. Formalization provides two benefits: First, we should be able to make predictions about what someone believes, based on what we have been able to glean of their models. Second, we should be able to use machine learning techniques to construct generalizations across particular people over time, or across people from particular groups, to concisely capture common properties of the models of people and groups. In this paper, we examine the relationship between culture, expertise, and causal reasoning in the domain of biology. Culture is defined here as the causally distributed patterns of mental representations, their public expressions, and the resultant behaviors in given ecological contexts (Atran, Medin & Ross, 2005; Sperber, 1985; 1996). People's mental representations interact to the extent that those representations can be physically transmitted in a public medium (language, dance, signs, artifacts, etc.). These public representations, in turn, are sequenced and channeled by ecological features of the environment (including the social environment) that constrain interactions between individuals.

The cultural communities involved in the present work include rural Menominee Native Americans, rural European Americans and Northwestern undergraduate students. The Menominee live on 234,000 acres of heavily forested land along the Wolf river in Northeast Wisconsin. The European Americans involved in this research live in the neighboring town of Shawano. Menominee individuals are more likely to engage in culture-specific ceremonial practices outdoors and are also more likely to simply engage in 'observing' practices (e.g., walks in the forest), whereas rural European Americans are more likely to engage in outdoor sporting activities (e.g., fishing competitions) and outdoor work-related activities (e.g., landscaping; Bang, 2007). One of our goals is to examine the similarities and differences in causal reasoning for biological kinds between these two cultures. By automatically constructing generalizations from field data, we should get a more objective perspective on these differences. One important way to test this hypothesis is to train classifiers, to automatically recognize which culture a causal model belongs to.

First, we discuss the role that culture plays in causal reasoning. Next, we describe our Qualitative Concept Map (QCM) system, used here to construct models of food webs from interview transcripts. We then describe how we use cognitive simulations of analogical matching and generalization to automatically construct generalizations that are used for classification. Experimental results are discussed, followed by related and future work.

The Role of Culture and Expertise in Reasoning about Biological Kinds

There are many reasons to believe that there might be similarities in individuals' causal understanding of relationships in nature. Medin, Atran, and their colleagues (see Atran et al., 2005; Medin & Atran, 2004), building on decades of important work in ethnobiology (see Berlin, 1992 for one summary), have found that, in spite of highly varying input, a few key principles guide the recognition and organization of biological information in extraordinarily similar ways. For instance, there is marked cross-cultural agreement on the hierarchical classification of living things, such that plants and animals are grouped according to a ranked taxonomy with mutually exclusive groupings of entities at each level (Atran, 1990; Berlin, Breedlove, & Raven, 1973; 1974; Brown, 1984; Hays, 1983; Hunn, 1977). The highest level of taxonomic organization includes the most general categories, such as the folk kingdom rank (which includes groupings such as plants and animals), and lower levels distinguish between increasingly greater degrees of specificity (e.g., life forms such as tree or bird; generic species level such as oak or

blue jay). The generic species level appears to be consistently privileged for inductive inference when generalizing properties across plants and animals, as it is the lowest level for which inductive power is the greatest, and only minimal inductive advantage is gained at more subordinate levels (Coley, Medin, and Atran, 1997). There is cross-cultural agreement that the appearance and behavior of every species is caused by an internal biological (and usually unspecified) essence that is inherited from the birth parents and is responsible for identity persistence in the face of physical and developmental transformation (Atran, 1998; Atran, Estin, Coley, & Medin, 1997; Gelman, 2003; Gelman & Wellman, 1991; Medin & Atran, 2004; Sousa, Atran, & Medin, 2002).

However, there is also evidence suggesting considerable variability within these universal constraints in folk biological concept formation as a function of both experience with the natural world and cultural salience (two highly related factors). For instance, Rosch and Mervis (1975) have found that the life form level is the level for which urban undergraduates possess the greatest knowledge (i.e., basic level), but Berlin (1992) found that among traditional societies in which individuals have more direct experience with the natural environment, the basic level corresponds to the generic-species level, and these differences have been attributed to differences in expertise (Medin & Atran, 2004). Other findings implicate cultural differences above and beyond expertise. For instance, Menominee Native Americans are more likely than rural European Americans to see themselves as a part of nature rather than apart from nature and to say that every creature has a role to play on Mother Earth (Bang, Unsworth, Townsend, & Medin, 2005).

When asked to sort biological kinds into categories, individuals from different communities vary not only in their taxonomic sorting but also in the degree to which they spontaneously sort along ecological dimensions, and this difference is not as predictable on the basis of expertise or experience alone. Specifically, Medin, Ross, Atran, Burnett, and Blok (2002) found that Menominee Native American fisherman and European American fishermen, who both have similar levels of expertise about fish and fish habitats, exhibit differences in ecological sorting of fish during a regular sorting task. Menominee fishermen are significantly more likely to sort in terms of ecological relationships. This pattern was found for both expert fishermen and for nonexperts in the two communities. Furthermore, in a subsequent task involving questions about fish-fish interactions, Menominee fishermen were significantly more likely to report positive and reciprocal relations, although both groups were equally likely to report negative relations.

Similar differences in ecological reasoning were found for children from these communities, such that Menominee children were more likely to reason about shared properties between living things on the basis of ecological relations, relative to rural European American children (Ross, Medin, Coley, & Atran, 2003). Differences in ecological reasoning appear to be the result of both culture and expertise, as rural European American children were more likely to engage in ecological-based reasoning than were urban European American children who had comparatively less experience with the natural world.

Do you think that the disappearance in the bears would affect other plants and animals in the forest?
-Probably just like shrubs and stuff that these animals the basic food sources like berry plants and stuff. And then maybe larger trees, too, because bears climb trees.
…
Because of there's a more competing for water in the soils. There's more shade, because I'm assuming it's a taller tree. So there's more shade so the ground growth couldn't grow as well. It would provide more nesting areas for the animals that use it for nesting. So they might benefit from it but they'd have less food.
-Right. And so do you think that other trees would be affected?
-Yes, because there's a competing for space. So the underbrush and that wouldn't grow as well, or any tree that's smaller.
-Yeah. And what about larger animals like bears? Do you think that they would be affected?
- If their food source was decreased because of the lack of undergrowth.

Figure 1: Excerpts from a transcript

Although prior research suggests that there are cross-cultural differences in causal models, little research has focused on directly assessing such differences. Consequently, we interviewed experts (i.e., hunters and fishermen) and novices (individuals who do not hunt or fish) from Menominee Native American and from European American cultural communities. Participants were presented a scenario in nature and were asked open-ended questions about the scenarios. Transcriptions of three scenarios were modeled in the present study. In each scenario, participants were told about a perturbation in an ecological system and were asked to speculate about the effects of such an event on other plants and animals in the forest. In one scenario, the perturbation involved the disappearance of all of the bears in a nearby forest. In another scenario, the perturbation involved a doubling of the bear population in a nearby forest. In a third scenario, the perturbation involved the disappearance of all of the poplar trees in a nearby forest. Each participant was presented with all three scenarios, and after each scenario participants were first allowed to openly discuss any consequences that came to mind before being probed with an exemplar (e.g., eagle) that represented a particular trophic type with respect to the perturbation species (e.g., competitor). Given the open-ended nature of the interviews, the number of probes presented to participants varied across individuals depending on the depth of initial responses and the degree to which they responded to subsequent probes.

The verbal explanations of the subjects were transcribed (see Figure 1 for example), and used as data to construct

formal qualitative models expressing their beliefs. Based on previous research cited above, we predicted that Menominee Native Americans' causal mental models of nature would be more inclusive and would include more interconnections, relative to rural European Americans.

Qualitative Concept Maps

We use the Qualitative Concept Maps (QCM) system to create formal models based on transcripts of the interviews. QCM provides a friendly interface for

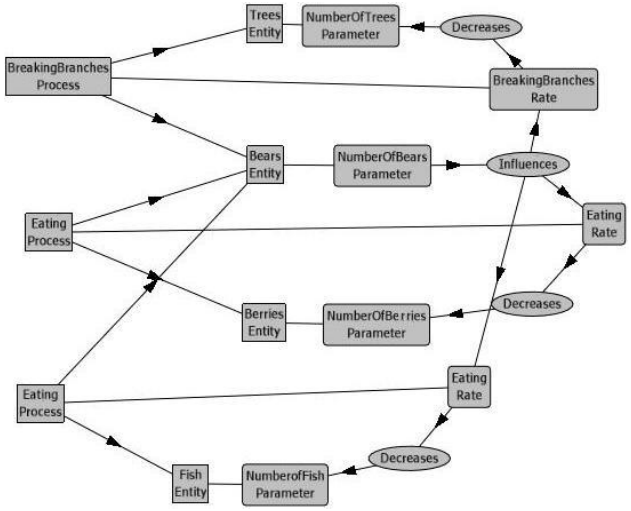


Figure 2: A QCM model

experimenters to explore causal models using Qualitative Process (QP) Theory semantics (Forbus, 1984). QP theory as a representation language for physical phenomena includes:

- Continuous parameters (*quantities*)
- Causal relationships between them (*influences*)
- Mechanisms underlying physical causality (*physical processes*)

QCM uses a concept map interface (Novak & Gowin, 1984). QCM automatically checks for any modeling errors which violate the laws of QP theory, providing detailed error messages.

QCM uses multiple panes to represent distinct qualitative states. This is important for capturing changes over time. For example, in the scenarios outlined above, participants would often discuss immediate effects of a change followed by long-term effects of changes. Figure 2 illustrates one pane from a model for the Bears Disappearing scenario. The *meta-pane* (Figure 3) allows modelers to see all the states at once. Modelers can easily extend the vocabulary of specific processes and quantities used in the models, to expedite model creation.

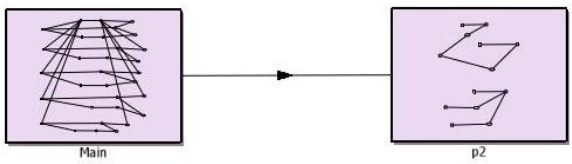


Figure 3: The meta-pane provides an overview of the qualitative states in the model

QCM can import and export models via GraphML (Brandes et al., 2002), allowing graphs drawn in QCM to be easily viewed in other graph drawing programs. This facilitates collaboration between modelers. More importantly, for cognitive simulation purposes, models can be exported as predicate calculus statements. This enables QCM models to be used in a variety of reasoning systems. We are also working on directly importing propositional statements into QCM, to visualize models constructed via other systems. In this paper, we use the propositional statements produced by QCM to automatically construct generalizations, testing them via learning a classifier.

Computational Experiments

Here we describe a method for building generalizations from transcripts modeled in QCM. These generalizations make explicit the common structure found in the models. They can also be used to automatically categorize subsequent models, based on the culture they belong to. The learning technique that we use in this experiment has previously been used in automatic sketch recognition (Lovett, Dehghani and Forbus 2007), automatic music genre classification (Dehghani and Lovett 2006) and classifying terrorist activities by perpetrator (Halstead and Forbus 2007**)**. The major benefit of this technique is that, although it only requires very small training sets, utilizing qualitative representations it can achieve the performance of machine learning algorithms which require orders of magnitude larger data sets.

Comparison and Generalization

We compare representations using the Structure-Mapping Engine (SME) (Falkenhainer, Forbus and Gentner, 1989). SME is a computational model of similarity and analogy based on Gentner's (1983) structure mapping theory of analogy in humans. It works on structured representations, consisting of entities, attributes of entities and relations. There are both first-order relations between entities and higher-order relations between other relations. Given two representations in this form, a *base case* and a *target case*, SME aligns their common structure to form a mapping between the cases. This mapping consists of a set of correspondences between entities and expressions in the two cases. SME tries to find mappings that maximize

systematicity; that is, it prefers mappings with higher-order relations and relationally connected structure.

Our system learns categories of objects using SEQL (Kuehne et al, 2000), a model of generalization built on SME. SEQL is based on the idea that when humans are exposed to multiple exemplars of a category, they construct generalizations by comparing the exemplars and abstracting out the common structure. SEQL does this by comparing individual cases with SME. For each category, SEQL maintains a list of generalizations and exemplars. Each new incoming exemplar is compared against the existing generalizations, and if it is sufficiently similar, the generalization is refined based on their common structure. Otherwise, the exemplar is compared against other, unassimilated exemplars. If sufficiently close to one of them, a new generalization is formed from their common structure. Originally non-overlapping structure was simply thrown away. Now, SEQL associates a probability with every expression in a generalization which is updated with each new exemplar, and only gets rid of very low-probability structure (Halstead and Forbus 2005). SEQL can be forced to construct a single generalization for a category by simply setting the assimilation threshold to be extremely low.

Results

81 transcripts, generated in response to three food web scenarios, were modeled using QCM. The transcripts for two additional scenarios were excluded because participants rarely responded to these scenarios in detail. These two scenarios were structurally similar to the other scenarios presented and were always presented at the end of the interview, and so it is speculated that participants perceived repetition as they progressed through the interview and reduced responding as a result.

We randomly divided the models into a test and a training set 1,000 different times. In each run, we used SEQL to produce two generalizations, one for Menominee and one for non-Menominee, from the models in the training set. These generalizations were then used to classify models in the test set by using SME to compare each model with the two generalizations. We calculated the percentage of the model's expressions that aligned with each generalization, and the percentage of the generalization's expressions that aligned with the model, and classified test models based on which generalization it had more in common with. We tabulated successful classification by cultural group and averaged the results over all 1,000 trials.

Table 1 shows the results of our experiment. In the first two columns the percentage of Menominee models being correctly classified as Menominee and non-Menominee being classified as non-Menominee are shown. The last column shows the overall accuracy of the system. The average accuracy across the three scenarios was 64%.

Our system was able to automatically compute generalizations which differentiated between the two culture models. Our system was also able to find similarities in causal models from the same culture. By examining the system's results, we can gain insights into the differences and similarities between the models. Specifically, we found that the number of facts that were consistent across individuals was higher in Menominee models. We examined the generalizations from a single test run for each scenario, in which the system achieved 70% accuracy. For this test run, there were 24 facts found consistently across all Menominee models vs. 16 facts for non- Menominee. Also, the number of consistent causal relations was higher among Menominee. Menominee models contained 4 causal relations found consistently across all models, whereas non-Menominee models only contained 2. We can conclude from this result that causal understanding of relationships in nature is more homogeneous among Menominee than among non-Menominee.

	Menominee	Non-Menominee	Overall Accuracy
Bears Disappearing	65%	57%	61%
Bears Doubling	82%	52%	67%
Poplar Disappearing	64%	64%	64%

Table 1: Performance of the classification system

As per our prediction, the generalizations that were made from Menominee models were more detailed, larger and therefore subsumed other smaller generalizations. This had the unfortunate side-effect of biasing models towards being classified as Menominee. However, as mentioned above, the open-ended nature of the interviews led to variation in the number of probes presented to participants across individuals, and the resultant variability in responses can introduce some difficulty when attempting to evaluate similarities in causal maps. Open-ended interviews are useful for exploratory investigations of the ways in which participants are likely to respond to hypothetical scenarios, and future research can build on the knowledge gained here. Specifically, the present results can now be used as a basis for designing a more structured survey in which participants are presented with a larger, more comprehensive list of animals and plants that represent all of the trophic levels and ecological considerations mentioned by Menominee and non-Menominee adults of varying hunting and fishing expertise. This should help provide the most systematic probing of their knowledge.

Related Work

QCM can be thought of as the second generation VModel (Forbus, Carney, Harris and Sherin, 2001). VModel was developed to help middle-school students learn science. Like QCM, it uses a subset of QP theory to provide strong semantics. However, VModel was limited to single-state reasoning, whereas QCM can be used to model physical

causal phenomena with multiple states. Similar differences hold with Betty's Brain (Biswas *et al* 2001), which provides a concept-map interface for single-state qualitative reasoning designed for middle-school students.

The closest other qualitative modeling tools are MOBUM (Machado & Bredeweg, 2001) and VISIGARP (Bouwer & Bredeweg 2001), which have lead to Garp3.[1] Like QCM, these environments are aimed at researchers, but their focus is on constructing models for qualitative simulation, using generic, first-principles domain theories. QCM focuses instead on helping capture concrete, situation-specific qualitative explanations of phenomena. Thus it provides a useful tool for scientists working with interview data.

Discussion

We have shown that cultural differences in causal reasoning about food webs can be captured to some degree in terms of similarities and differences in qualitative models extracted from transcript data. Although previous manual analysis of the transcripts have shown to be very difficult and time consuming, by using SEQL and SME we were able to find similarities and differences and automatically cluster causal models built from the transcripts. While the results are significant, the accuracy could be improved, and we plan to use a more stringent interview protocol to test this. We also plan to use more than one expert for modeling the results. Also, we are investigating how causal models of hunters (experts) are different from non-hunters (novices).

More generally, we are encouraged by the success of QCM in providing a scientist friendly environment, where QP theory can be used to model interview data. We plan to extend QCM in several ways. First, we plan to use similarity-based qualitative simulation (Yan & Forbus, 2005) to support creating predictions based on learned generalizations from transcript models. Second, we plan to integrate our qualitative simulator (Gizmo), to provide a complementary first-principles simulation engine. Finally, we plan to provide a more comprehensive interface, to provide a unified platform for representing, clustering, and reasoning about qualitative models derived from data.

Acknowledgments

This research was supported in part by a grant from the US Air Force Office of Scientific Research. We thank Doug Medin for many useful insights.

[1] http://hcs.science.uva.nl/QRM/index.html

References

Atran, S. (1990). *Cognitive Foundations of Natural History*. Cambridge, England: Cambridge University Press.

Atran, S. (1998). Folk biology and the anthropology of science: Cognitive universals and cultural particulars. *Behavioral and Brain Sciences, 21*(4), 547-609.

Atran, S., Estin, P., Coley, J., & Medin, D. L. (1997). Generic species and basic levels: Essence and appearance in folkbiology. *Journal of Ethnobiology, 17*, 22-45.

Atran, S., Medin, D. L., Ross, N., Lynch, E. B. Vapnarsky, V., Ucan Ek', E., et al. (2002). Folkecology, cultural epidemiology, and the spirit of the commons: A garden experiment in the Maya Lowlands, 1991-2001. *Current Anthropology, 43*, 421-450.

Atran, S., Medin, D. L., and Ross, N. O. (2005). The cultural mind: Environmental decision making and cultural modeling within and across populations. *Psychological Review, 112*, 744-776.

Bang, M., and Medin, D. L. (in preparation). The cultural context of learning: Native-American science education.

Bang, M., Townsend, J., Unsworth, S. J., and Medin, D. L. (2005). *Cultural models of nature and their relevance to science education.* Paper presented at the 2005 Annual Meeting of the American Education Research Association, Montreal, Quebec.

Berlin, B. (1992). *Ethnobiological Classification*. Princeton, NJ: Princeton University Press.

Berlin, B., Breedlove, D. E., and Raven, P. H. (1973). General Principles of Classification and Nomenclature in Folk Biology. *American Anthropologist, 75*, 214-242.

Berlin, B., Breedlove, D. E., and Raven, P. H. (1974). *Principles of Tzeltal plant classification; an introduction to the botanical ethnography of a Mayan-speaking people of highland Chiapas*. New York: Academic Press.

Biswas, G., Schwartz, D., Bransford, J. & The Teachable Agents Group at Vanderbilt. Technology Support for Complex Problem Solving: From SAD Environments to AI. In Forbus, K. and Feltovich, P. (eds) 2001 *Smart Machines in Education: The coming revolution in educational technology*. AAAI Press.

Brandes U., Eiglsperger M., Herman I., Himsolt M., and Marshall M. S. (2001): GraphML Progress Report: Structural Layer Proposal. Proc. 9th International Symposium. Graph Drawing (GD '01), LNCS 2265, pp. 501-512.

Brown, C. (1984). *Language and Living Things: Uniformities in Folk Classification and Naming*. New Brunswick, NJ: Rutgers University Press.

Bouwer A. and Bredeweg B. (2001): VISIGARP: graphical representation of qualitative simulation models, in: *Artificial Intelligence in Education – AI-ED in the Wired and Wireless Future*, J.D. Moore, G. Luckhardt Redfield and J.L. Johnson, eds, IOS Press/Ohmsha, Osaka, Japan, 2001, pp. 294–305.

Coley, J., Medin, D. L., and Atran, S. (1997). Does rank have its privilege? Inductive inferences in folkbiological taxonomies. *Cognition, 63*(73-112).

Dehghani, M., and Lovett, A. (2006). Efficient Genre Classification using Qualitative Representations. In the Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06). Victoria Canada.

Falkenhainer, B., Forbus, K., and Gentner, D. 1989. The Structure-Mapping Engine: Algorithms and Examples. Artificial Intelligence 41: 1-63

Forbus, K. (1984). Qualitative process theory. Artificial Intelligence,24, 85-168.

Forbus, K., Carney, K., Harris, R., and Sherin, B. (2001) Modeling Environment for Middle-School Students: a Progress Report. In G. Biswas ed. *Papers from the 2001 Qualitative Reasoning Workshop* Stoughton, WI, The Printing House.

Gelman, S. A. (2003). *The essential child : origins of essentialism in everyday thought*. New York: Oxford University Press.

Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. Cognitive Science 7: 155-170.

Gelman, R., and Wellman, H. M. (1991). Insides and essences. *Cognition, 38*, 214-244.

Halstead, D. and Forbus, K. (2007). Some Effects of a Reduced Relational Vocabulary on the Whodunit Problem. In *Proceedings of the International Joint Conference on Artificial Intelligence.*

Halstead, D., and Forbus, K. (2005). Transforming between Propositions and Features: Bridging the Gap. In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05). Pittsburgh, PA: AAAI Press.

Hays, T. (1983). Ndumba folkbiology and general principles of ethnobotanical classification and nomenclature. *American Anthropologist, 85*, 592-611.

Hunn, E. (1977). *Tzeltal Folk Zoology: The Classification of Discontinuities in Nature*. New York: Academic Press.

Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. (2000). SEQL: Category Learning as Progressive Abstraction Using Structure Mapping. In Proceedings of the 22nd Annual Conference of the Cognitive Science Society, 770-775. Philadelphia, PA.

Lopez, A., Atran, S., Coley, J. D., & Medin, D. L. (1997). The tree of life: Universal and cultural features of folkbiological taxonomies and inductions. *Cognitive Psychology, 32*(3), 251-295.

Lovett, M., Dehghani, M., Forbus K. (2007). Incremental Learning of Perceptual Categories for Open-Domain Sketch Recognition. In the Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07). Hyderabad, India: AAAI Press.

Machado, V. and Bredeweg, B. 2001. Towards Interactive Tools for Constructing Articulate Simulations. *Proceedings of QR01*

Medin, D. L., and Atran, S. (2004). The Native Mind: Biological Categorization and Reasoning in Development and Across Cultures. *Psychological Review, 111*(4), 960-983.

Medin, D. L., Ross, N. O., Atran, S., Burnett, R., and Blok, S. (2002). Categorization and Reasoning in Relation to Culture and Expertise. On *Psychology of Learning and Motivation, 41*. New York: Academic Press.

J.D. Novak and D.B. Gowin, Learning How To Learn, New York: Cambridge University Press. 1984.

Proffitt, J. B., Coley, J. D., and Medin, D. L. (2000). Expertise and category-based induction. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*(4), 811-828.

Rosch, E., and Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology, 7*(4), 573-605.

Ross, N., Medin, D., Coley, J. D., and Atran, S. (2003). Cultural and experimental differences in the development of folkbiological induction. *Cognitive Development, 18*(1), 25-47.

Sousa, P., Atran, S., and Medin, D. (2002). Essentialism and folkbiology: Evidence from Brazil. *Journal of Cognition and Culture, 2*(3), 195-223.

Sperber, D. (1985). Anthropology and Psychology: Towards an epidemiology of representations. *Man, 20*, 73-89.

Sperber, D. (1996). *Explaining Culture: A Naturalistic Approach*. Oxford: Blackwell.

Yan, J. and Forbus, K. 2005. Similarity-based qualitative simulation. *Proceedings of the 27th Annual Meeting of the Cognitive Science Society.*

Dynamic domain abstraction through meta-diagnosis

Johan de Kleer
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
dekleer@parc.com

Abstract

One of the most powerful tools designers have at their disposal is abstraction. By abstracting from the detailed properties of a system, the complexity of the overall design task becomes manageable. Unfortunately, faults in a system need not obey the neat abstraction levels of the designer. This paper presents an approach for identifying the abstraction level which is as simple as possible yet sufficient to address the task at hand. The approach chooses the desired abstraction level through applying model-based diagnosis at the meta-level, i.e., to the abstraction assumptions themselves.

1 Introduction

Of the many tools designers have at their disposal, abstraction is one of the most powerful. By abstracting from the detailed properties of a system, the complexity of the overall design task becomes manageable. For example, a computer engineer can focus on the logic level without concern for the properties of the individual transistors which make up a particular gate, and a chip designer can layout a chip without being concerned with the fabrication steps needed to construct it. Abstraction allow designers to partition concerns into independent black boxes and is one of the most important ideas underlying the design of modern technology.

Unfortunately, faults in a system need not obey the neat abstraction levels of the designer. A fault in a few transistors can cause an Intel Pentium processor to generate an ocasional incorrect floating point result. To understand this fault requires transcending the many abstraction levels between software and hardware. A PC designer can focus on functional layout without being concerned about the physical layout and its thermal properties. However, a technician must determine that the processor crashed because dust sucked into the processor fan clogged the heatsink and allowed the processor temperature to rise to such a dangerous level that the PC automatically shut down. As a consequence diagnostic reasoning is inherently messy and complex, as it involves crossing abstraction boundaries never contemplated by the designers.

Existing model-based reasoning has addressed a number of types of abstraction.

- Range abstraction. The ranges of variables are abstracted, e.g., instead of a continuous quanity it might be represented by the qualitative values of $-$, 0 or $+$. [Struss, 1991b; 1991a; Sachenbacher and Struss, 2005; Torta and Torasso, 2003]
- Structural abstraction. Groups of components are abstracted to form hierarchies [Chittaro and Ranon, 2004; Hamscher, 1990].
- Model selection. Approaches to choosing among a collection of hand-constructed models [Addanki *et al.*, 1989; Falkenhainer and Forbus, 1991]

In this paper we present a new type of abstraction (*domain* abstraction): changing the physical principles which underlie models, such as moving from the $0/1$ level to currents and voltages and providing a systematic approach to choosing the appropriate domain for the diagnostic task.

Choosing the right domain abstraction level requires balancing two opposing desiderata. Reasoning at the highest abstraction level is the simplest. Unfortunately, it may be inadequate to analyze or troubleshoot the system. Instead, the system needs to be analyzed at a more detailed level. On the other hand, reasoning at too low of a level can require enormously more computational resources and difficult to obtain parameters, and it generates more complicated analyses. As Albert Einstein reportedly said: "Make everything as simple as possible, but not simpler."

Technicians expect that systems are non-intermittent and that the schematic is an accurate description of the physical system. Consider the simple analog circuit of Figure 1. Suppose a technician measures the current to be 0 ampere (1 is expected), which leads to an inference that the resistor is faulty, but repeating the measurement gives 1 ampere. Either the resistor is intermittent or there is a fault in the connections. The technician must change abstraction level to diagnose this system further by, for example, checking the connections or further tests on resistor R itself.

Consider a circuit of three logic inverters in sequence, with its output fedback to its input (Figure 2). At the usual gate level of analysis, an inverter simply complements its input. This circuit has no inputs, so we need to consider the possible values at the connecting nodes. Assume the input to inverter A is 0. Its output must be 1. The output of B must be 0. The output of D must be 1. This is impossible, as we assumed

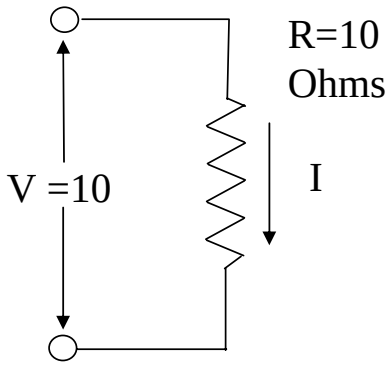


Figure 1: Simple analog diagnosis problem.

it was 0. Conversely, assume the input to inverter A is 1. Its output must be 0. The output of B must be 1. The output of D must be 0. This is impossible, as we assumed it was 1. Therefore, the input of inverter A can neither be 0 or 1. Also, the inputs of inverters B or D cannot be 0 or 1. Somehow the circuit is contradictory when modeled as logic gates. Thus, one of the components A, B or D must be faulted. Suppose the technician chooses to systematically remove and check each of these three components for proper functioning. If each component is confirmed to be correct, the technician has encountered an impasse.

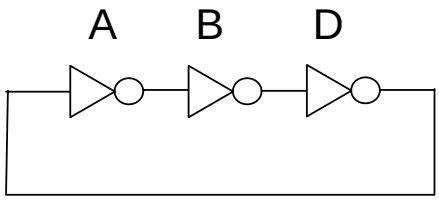


Figure 2: A very simple circuit which yields a contradiction when analyzed as combinational logic; yet its a perfectly reasonable fault-free circuit with well-defined behavior.

Analyses that result in contradictions are the most important indicator that the level of abstraction used is too simplistic. In this paper we present a general reasoner which automatically descends abstraction layers to perform needed analyses, and which does not descend abstraction levels needlessly. This approach is broadly applicable, but we explore these ideas in the context of digital circuits with messier models that include failures in connections, intermittents, and temporal behaviors.

2 Meta-Diagnosis

Figure 3 characterizes the basic architecture of a typical model-based, component-based diagnosis engine. Given the component topology (e.g., the schematic for analog circuits), component models (e.g., resistors obey ohm's law) and observations (e.g., the voltage across resistor $R6$ is 4 volts), it computes diagnoses which explain all the observations. Observations inconsistent with expectations guide the discovery of diagnoses. When the MBD engine can find no diagnoses it signals a failure.

Suppose we need to troubleshoot the circuit of Figure 2. Most diagnosis systems would immediately conclude that some subset of the components $\{A, B, D\}$ is faulted. However, testing each inverter individually demonstrates that all the components are good. As a consequence, most algorithms would report an unresolvable contradiction.

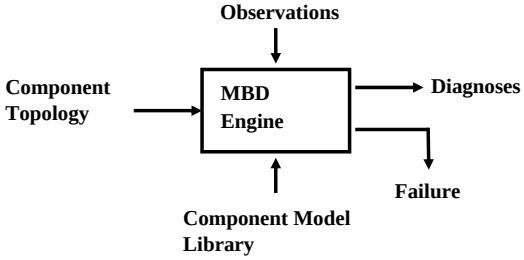


Figure 3: Basic architecture of a model-based diagnosis engine.

The architecture of Figure 4 includes two model-based diagnosis engines. The top model is used to identify the best abstraction level, and the bottom model performs the actual system diagnosis. This composite architecture has the same inputs as the basic architecture with one additional input: the abstraction library. The 'Applicable Models' module identifies all the applicable abstractions for the component topology. The 'Modeler' module uses the preferred meta-diagnosis to construct conventional model-based diagnosis models using the 'Component Model Library.'

Consider the example of Figure 2. The component topology is simply the circuit schematic as before. The system observations are as before (e.g., the output of A is 1). The component model library is will contain different models for gate behavior (e.g., boolean, analog, thermal, temporal, etc.). The new input, the abstraction library, is the set of all possible abstractions. Instead of a usual component topology, the abstraction MBD engine is provided width a set of possible abstractions applicable to the given system to be diagnosed. Initially, there are no meta-observations, so the preferred diagnosis is the one at the most abstract level (analogous to all components working). Therefore, the domain MBD engine will perform diagnosis in the usual way with the most abstract models. Suppose each gate is physically checked, leading to the observations that A, B and D are working correctly. The domain model-based engine now fails as it has found an unresolvable contradiction. This invokes the abstraction MBD engine as an observation. As analysis proceeds, the preferred meta-diagnosis will descend abstraction levels. For the purposes of this paper, the preferred meta-diagnosis is one minimal cardinality meta-diagnosis.

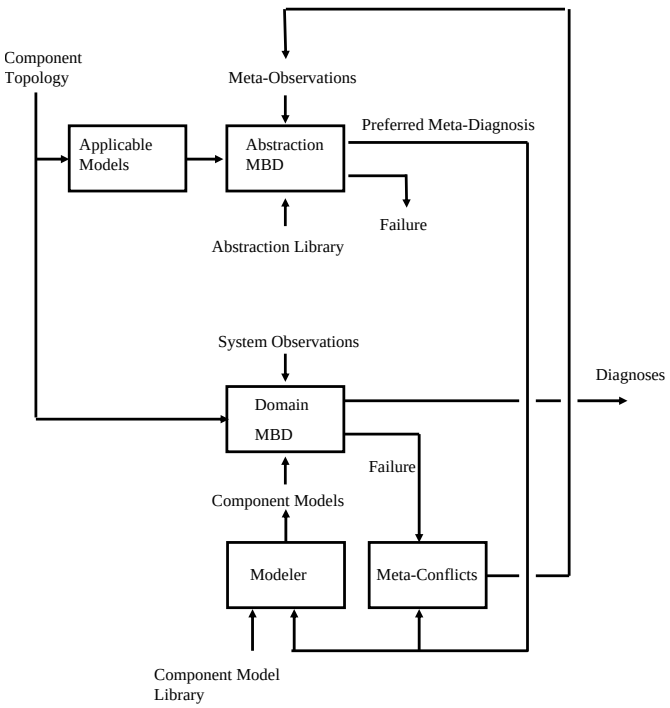


Figure 4: Architecture of an abstraction-based, model-based diagnosis engine.

3 Formalization

This section summarizes the formal framework for model-based diagnosis we use in the rest of the paper [de Kleer and Williams, 1987; de Kleer *et al.*, 1992]. In order to distinguish between domain and abstraction AB literals, we state the usual framework in terms of domain AB_d literals.

Definition 1 *A system is a triple (SD,COMPS, OBS) where:*

1. *SD, the system description, is a set of first-order sentences.*
2. *COMPS, the system components, is a finite set of constants.*
3. *OBS, a set of observations, is a set of first-order sentences.*

In Figure 3 *SD* is the component topology and component model library, and *COMPS* is the set of components in the component topology.

Definition 2 *Given two sets of components Cp and Cn define $\mathcal{D}_d(Cp, Cn)$ to be the conjunction:*

$$\Big[\bigwedge_{c \in Cp} AB_d(c)\Big] \wedge \Big[\bigwedge_{c \in Cn} \neg AB_d(c)\Big].$$

Where $AB_d(x)$ represents that the component x is ABnormal (faulted).

A diagnosis is a sentence describing one possible state of the system, where this state is an assignment of the status normal or abnormal to each system component.

Definition 3 *Let $\Delta \subseteq$COMPS. A diagnosis for (SD,COMPS,OBS) is $\mathcal{D}_d(\Delta, COMPS - \Delta)$ such that the following is satisfiable:*

$$SD \cup OBS \cup \{\mathcal{D}_d(\Delta, COMPS - \Delta)\}$$

Definition 4 *An AB_d-literal is $AB_d(c)$ or $\neg AB_d(c)$ for some c ∈ COMPS.*

Definition 5 *An AB_d-clause is a disjunction of AB_d-literals containing no complementary pair of AB_d-literals.*

Definition 6 *A conflict of (SD,COMPS,OBS) is an AB_d-clause entailed by SD ∪ OBS.*

3.1 Formalizing abstraction

The abstraction MBD is defined analogously:

Definition 7 *An abstraction system is a triple (SD,ABS, OBS) where:*

1. *SD, constraints among possible abstractions, is a set of first-order sentences.*
2. *ABS, the applicable abstractions, is a finite set of constants.*
3. *OBS, a set of meta-observations, is a set of first-order sentences.*

Definition 8 *Given two sets of abstractions Cp and Cn define $\mathcal{D}_a(Cp, Cn)$ to be the conjunction:*

$$\Big[\bigwedge_{c \in Cp} AB_a(c)\Big] \wedge \Big[\bigwedge_{c \in Cn} \neg AB_a(c)\Big].$$

Where $AB_a(x)$ represents that the abstraction x is ABnormal (cannot be used).

A meta-diagnosis is a sentence describing one possible state of the system, where this state is an assignment of the status normal or abnormal to each system component.

Definition 9 *Let $\Delta \subseteq$ABS. A meta-diagnosis for (SD,ABS,OBS) is $\mathcal{D}_a(\Delta, ABS - \Delta)$ such that the following is satisfiable:*

$$SD \cup OBS \cup \{\mathcal{D}_a(\Delta, ABS - \Delta)\}$$

Definition 10 *An AB_a-literal is $AB_a(c)$ or $\neg AB_a(c)$ for some c ∈ ABS.*

Definition 11 *An AB_a-clause is a disjunction of AB_a-literals containing no complementary pair of AB_a-literals.*

Definition 12 *A meta-conflict of (SD,ABS,OBS) is an AB_a-clause entailed by SD ∪ OBS.*

4 Example of a lattice of models

To illustrate these ideas we use 3 axes of abstraction:

- Model of connections as in [de Kleer, 2007b] which is an improvement over [Böttcher, 1995; Böttcher *et al.*, 1996].
- Model of non-intermittency [Raiman *et al.*, 1991] or intermittency [de Kleer, 2007a]
- Model of time [de Kleer, 2007c].

The corresponding AB_a literals are:

- $\neg AB_a(C)$ represents the abstraction that connections need not be modeled.
- $\neg AB_a(I)$ represents the abstraction that the system is non-intermittent.
- $\neg AB_a(T)$ represents the abstraction that temporal behavior need not be modeled.

Figure 5 shows a portion of the abstraction space for digital circuits along three of the axes of abstraction. This lattice is identical in structure to the ones used in conventional model-based diagnosis for system diagnoses. In conventional model-based diagnosis, each node represents a candidate diagnosis which explains the observations. Each node in Figure 5 represents a candidate meta-diagnosis. The bottom node in the figure represents the meta-diagnosis in which connections, time, and intermittency are not relevant:

$$\neg AB_a(T) \wedge \neg AB_a(C) \wedge \neg AB_a(I).$$

Under the principle that we want to find the simplest meta-diagnosis which explains the observations (and no simpler), we are primarily interested in the minimal diagnoses. For brevity sake, we name meta-diagnoses with the letters corresponding to the abstractions which are AB_a. For example, the meta-diagnosis $\neg AB_a(T) \wedge \neg AB_a(C) \wedge AB_a(I)$ is named by I.

For the example in Figure 2, analysis immediately detects a contradiction and the meta-conflict:

$$AB_a(T) \vee AB_a(C)$$

(this contradiction cannot depend on $AB_a(I)$ as there is only one observation time so far). Figure 6 illustrates the resulting meta-diagnosis lattice. Every meta-diagnosis below the curve is eliminated. The minimal meta-diagnoses are T and C.

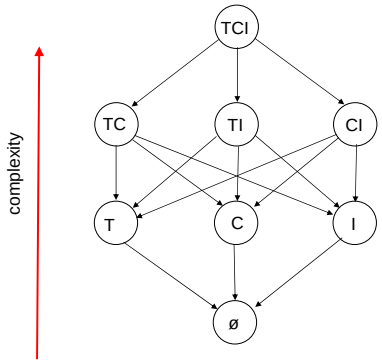


Figure 5: Meta-Diagnosis lattice for digital gates. T indicates temporal models; C indicates connection models; I indicates intermittent models.

5 Modeling components

The conventional MBD model for an inverter is (presuming the usual background axioms define the appropriate functions, domains, and ranges):

$$INVERTER(x) \rightarrow$$

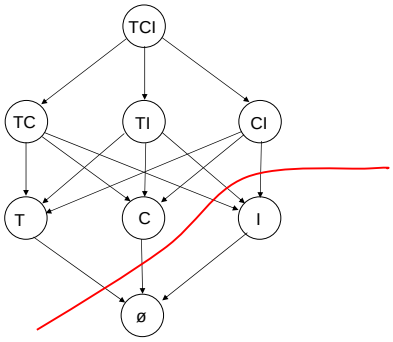


Figure 6: Meta-Diagnosis lattice for digital gates. The meta-conflict $AB_a(T) \vee AB_a(C)$ rules out all meta-diagnoses below the curved line. The minimal meta-diagnoses are T and C.

$$\Big[\neg AB_d(x) \rightarrow [in(x,t) = 0 \equiv out(x,t) = 1]\Big].$$

As this model presumes connections and temporal behavior need not be modeled, in our new framework it is written as:

$$\neg AB_a(T) \wedge \neg AB_a(C) \rightarrow \Big[INVERTER(x) \rightarrow \Big[\neg AB_d(x) \rightarrow [in(x,t) = 0 \equiv out(x,t) = 1]\Big]\Big].$$

Figure 7: Model of inverter under T and C abstractions.

When modeling an inverter as having a delay Δ, the model changes to (labeled T in Figure 5):

$$AB_a(T) \wedge \neg AB_a(C) \rightarrow \Big[INVERTER(x) \rightarrow \Big[\neg AB_d(x) \rightarrow [in(x,t) = 0 \equiv out(x,t+\Delta) = 1]\Big]\Big].$$

5.1 Connection models

To model the inverter to accommodate faults in connections, including bridge faults, requires the introduction of new formalisms. What follows is a brief summary of the formalism presented in [de Kleer, 2007b]. Each terminal of a component is modeled with two variables, one which models how the component is attempting to influence its output (roughly analogous to current), and the other which characterizes the result (roughly analogous to voltage). There are 5, mutually inconsistent, qualitative values for the influence of a component on a node (we refer to these as "drivers").

- $d(-\infty)$ indicates a direct short to ground.
- $d(0)$ pull towards ground (i.e., 0).
- $d(R)$ presents a high (i.e., draws little current) passive resistive load.

- $d(1)$ pull towards power (i.e., 1).
- $d(+\infty)$ indicates a direct short to power.

There are three possible qualitative values for the resulting signal:

- $s(0)$ the result is close enough to ground to be sensed as a digital 0.
- $s(x)$ the result is neither a 0 or 1.
- $s(1)$ the result is close enough to power to be sensed as a digital 1.

Using this formalism produces considerably more detailed component models. We need to expand the $A \equiv B$ in the inverter model to $(A \rightarrow B) \wedge (B \rightarrow A)$. The left half of the inverter model is:

$$\begin{aligned}
&\neg AB_a(T) \wedge AB_a(C) \rightarrow \\
&\quad \Big[INVERTER(x) \rightarrow \\
&\qquad \Big[\neg AB_d(x) \rightarrow \\
&\qquad \Big[[s(in(x,t)) = s(0) \rightarrow d(out(x,t)) = d(1)] \\
&\qquad \wedge [s(in(x,t)) = s(1) \rightarrow d(out(x,t)) = d(0)] \\
&\qquad \wedge d(in(x,t)) = d(R) \\
&\qquad \wedge [d(out(x,t)) = d(0) \vee d(out(x,t)) = d(1)]\Big]\Big].
\end{aligned}$$

We need explicit models to describe how the digital signal at a the node is determined from its drivers. Let $R(v)$ be the resulting signal at node v and $S(v)$ be the collection of drivers of node v. Intuitively, the model for a node is:

- If $d(-\infty) \in S(v)$, then $R(v) = s(0)$.
- If $d(+\infty) \in S(v)$, then $R(v) = s(1)$.
- If $d(0) \in S(v)$, then $R(v) = s(0)$.
- Else, if all drivers are known, and the preceding 3 rules do not apply, then $R(v) = s(1)$.

The resulting model for the node x will depend on $\neg AB_d(x)$ and $AB_a(C)$.

Modeling the inverter to more accurately describe both temporal and causal behavior (labeled TC in Figure 5):

$$\begin{aligned}
&AB_a(T) \wedge AB_a(C) \rightarrow \\
&\quad INVERTER(x) \rightarrow \\
&\qquad \Big[\neg AB_d(x) \rightarrow \\
&\qquad \Big[[s(in(x,t)) = s(0) \rightarrow d(out(x,t+\Delta)) = d(1) \\
&\qquad \wedge [s(in(x,t)) = s(1) \rightarrow d(out(x,t+\Delta)) = d(0)] \\
&\qquad \wedge d(in(x,t)) = d(R) \\
&\qquad \wedge [d(out(x,t+\Delta)) = d(0) \vee d(out(x,t+\Delta)) = d(1)]\Big]\Big].
\end{aligned}$$

The connection models also allow arbitrary bridge faults among circuit nodes. These are described in much more detail in [de Kleer, 2007b].

5.2 Modeling non-intermittency

Figure 8 shows an example where assuming non-intermittency improves diagnostic discrimination. The circuit's inputs and outputs are marked with values observed at times: T_1 and T_2. Note that at T_1, the circuit outputs a correct value and that at T_2, the circuit outputs an incorrect one. By assuming the Or gate behaves non-intermittently, we can establish that the Xor gate is faulty as follows:

If Xor is good, then $\text{In}_1(\text{Xor}, T_1) = 1$. This follows from $\text{In}_2(\text{Xor}, T_1) = 0$, $\text{Out}(\text{Xor}, T_1) = 1$ and the behavior of Xor. Similarly, if Xor is good, then $\text{In}_1(\text{Xor}) = 0$ at T_2. However, if Or behaves non-intermittently, then $\text{In}_1(\text{Xor}, T_2) = 1$. This follows because Or has the same inputs at both T_1 and T_2 and must produce the same output. Thus we have two contradictory predictions for the value of $\text{In}_1(\text{Xor}, T_2)$. Either Xor is faulty or Or is behaving intermittently. Assuming non-intermittency means Xor is faulty.

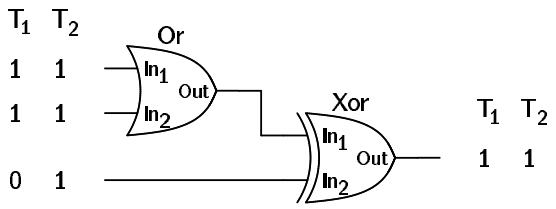


Figure 8: The power of non-intermittency.

All the inferences follow from the defining of non-intermittency:

Definition 13 *[Raiman* et al.*, 1991] A component behaves non-intermittently if its outputs are a function of its inputs.*

This definition sanctions the following inference: if an input vector $\overline{X}$ is applied to an intermittent component at time T, and output Z is observed, then in any other observation T', if $\overline{X}$ is supplied as input, Z will be observed as output.

For the Or-Xor example, the axioms added are:

$$\forall t.\text{Out}(\text{Or}, t) = F(\text{Or}, \text{In}_1(\text{Or}, t), \text{In}_2(\text{Or}, t)) \quad (1)$$

$$\forall t.\text{Out}(\text{Xor}, t) = F(\text{Xor}, \text{In}_1(\text{Xor}, t), \text{In}_2(\text{Xor}, t)) \quad (2)$$

F is a single fixed function for all non-intermittency axioms.

These axioms are implemented in the ATMS/HTMS-based reasoner by deriving prime implicates as follows. At time T_1:

$$AB_d(\text{Xor}) \vee [F(\text{Or}, 1, 1) = 1].$$

At time T_2:

$$AB_d(\text{Xor}) \vee [F(\text{Or}, 1, 1) = 0].$$

Which combine to yield $AB_d(\text{Xor})$.

In the intermittent case, the observation at T_1 equally weights Xor and Or as being correct. If there were other components in the system not affected by the measurement, the observation at T_1 lowers the posterior fault probabilities of Xor and Or.

5.3 Automatic generation of models

The more detailed component models can usually be generated automatically from the most abstract models in a systematic way. In our implementation, the T, C, and I models are automatically derived from the basic $\emptyset$ models by a set of modeling schemas. Consider the most abstract model of an inverter:

$$INVERTER(x) \rightarrow \Big[\neg AB_d(x) \rightarrow [in(x,t) = 0 \equiv out(x,t) = 1]\Big].$$

We use the convention that the function in refers to inputs, and the function out refers to outputs. A non-temporal model can be converted to a simple gate-delay model by replacing every occurrence of $out(x,t)$ (or $out_j(x,t)$) with $out(x, t + \Delta)$.

A non-connection model can be converted to a connection one by first expanding implications, replacing all $in(x,t) = y$ with $s(in(x,t)) = s(y)$ and $d(in,x,t)) = d(R)$ and replacing all $out(x,t) = y$ with $d(out(x,t)) = d(y)$, and adding the usual domain axioms for new variable values.

Non-intermittency requires no change to the component models themselves, but the axioms of Section 5.2 need to be added to the models supplied to the domain MBD.

6 The meta-diagnosis loop

6.1 $\emptyset \rightarrow T$

Consider the three inverter example of Figure 2. The most abstract meta-diagnosis is:

$$\neg AB_a(T) \wedge \neg AB_a(C) \wedge \neg AB_a(I).$$

This meta-diagnosis is supplied to the 'Modeler' module for Figure 4 which chooses the component models at the meta-diagnosis level. The models for all three inverters are described in Figure 7. This produces a failure because all components are known to be good. The 'Meta-Conflicts' module of Figure 4 will construct the meta-conflict:

$$AB_a(T) \vee AB_a(C).$$

$AB_a(I)$ is trivially excluded from the meta-conflict because non-intermittency inferences can only arise when the system has been observed at multiple times.

The abstraction MBD identifies two minimal meta-diagnoses T and C. If both are equally likely, it arbitrarily picks one. Suppose C is chosen. The C models do not resolve the inconsistency either. Figure 9 illustrates the following sequence of inferences with the connection models: (1) Assume the input of A is 1, (2) the causal inverter model drives its output down towards 0, (3) the input of gate B presents a high resistance (low-current) load to its node, (4) the connection model sets the node to 0, (5) the inverter model on B drives its output towards 1, (6) gate C presents a high resistance (low current) load, (7) the connection model sets its node to 1, (8) the inverter drives its output to 0, (9) gate A presents a low resistance (low current) load, and (10) the node model sets the node to 0 which contradicts the input of A being 1. An analogous analysis for the input of A being 0, yields a contradiction as well. The only remaining possible cardinality one diagnosis is T.

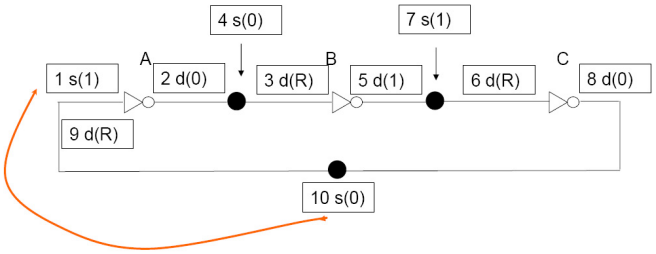


Figure 9: Modeling connections does not remove the failure.

Table 1: Outputs of the inverters of a ring oscillator after t gate delays. The oscillator takes 6 gate delays to return to its initial state, thus the output is a square wave with a period of 6 times the gate delay.

t	0	1	2	3	4	5	6
A	1	1	1	0	0	0	1
B	1	0	0	0	1	1	1
C	0	0	1	1	1	0	0

Using the temporal T models for the inverters produces a consistent analysis demonstrated in Table 1. This circuit is the familiar ring oscillator [Wikipedia, 2007].

6.2 $\emptyset \rightarrow I$

Consider the Or – Xor circuit again (Figure 8). For clarity assume the circuit has one fault. As derived in the Section 5.2, under the $\emptyset$ models, Xor must be faulted. Suppose we measure the output of the Or gate at T_1 and T_2 to be 1 and then 0 respectively. In this case, we have derived the meta-conflict:

$$AB_a(T) \vee AB_a(C) \vee AB_a(I).$$

There are now three minimal candidate meta-diagnoses: T,C,I. The T meta-diagnosis immediately results in a failure yielding the meta-conflict:

$$AB_a(C) \vee AB_a(I).$$

The meta-diagnosis I yields a consistent point of view: Or is failing intermittently. The C meta-diagnosis cannot explain the observation:

$$AB_a(T) \vee AB_a(I).$$

6.3 $\emptyset \rightarrow C$

Consider the Or – Xor example again before the output of the Or gate is observed. Again, for clarity assume the circuit has one fault. Suppose Xor is replaced and the same symptoms reoccur. In this case, both the C and I meta-diagnoses are consistent. Under the I meta-diagnosis, the circuit contains two possible faults:

- Xor is faulted.

- Or is faulted.

The C meta-diagnosis is consistent, with 3 possible faults:

- The node at the output of Xor is shorted to power.
- The connection from the output Xor gate to the node is open and thus it floats to 1.
- The connection to $\text{in}_2(\text{Xor})$ is shorted to ground.

6.4 $\emptyset \rightarrow TCI$

Tasks which require a TCI preferred meta-diagnosis are complicated, but they do occur. Consider the four inverter system of Figure 10. The input to inverter Z is held constant

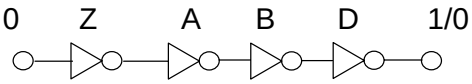


Figure 10: A very simple circuit containing a very hard to pinpoint fault.

at 0. We assume single faults. Observing the output D is usually 0, but outputs 1s with no pattern. The observation $D = 1$ indicates that one of Z, A, B, D is faulted. However, a subsequent observation of $D = 0$ is inconsistent yielding the meta-conflict:

$$AB_a(T) \vee AB_a(C) \vee AB_a(I).$$

No fault in the connections can produce the observations, therefore:

$$AB_a(T) \vee AB_a(I).$$

No temporal fault can lead to this behavior either, so:

$$AB_a(I).$$

Under meta-diagnosis I, the output of A is measured — it is usually 0, but sometimes 1. The output of Z is measured — it is usually 1, but sometimes 0. Therefore Z must be intermittently faulted (under meta-diagnosis I), but replacing it does not change the symptoms. This yields the meta-conflict:

$$AB_a(T) \vee AB_a(C).$$

The CI meta-diagnosis also leads to an inconsistency. There is no fault within the connections that can explain the observations. Likewise there is no fault within the TI meta-diagnosis. The only meta-diagnosis that can explain the symptoms is TCI. The actual fault is an intermittent short between the output of D and output of Z. As the input to Z is 0, its output is 1. The connection models for digital gates are 0-dominant, so that, if a 0 from the output of D were fedback through an intermittent short, it would drive the input to A to 0. Thus for those times in which the intermittent short was manifest, the circuit would be a ring oscillator.

7 Implementation

The analyses described in this paper have been implemented within the ATMS/HTMS framework [de Kleer and Williams, 1987; de Kleer, 1992]. Each domain or abstraction literal is represented by an explicit ATMS assumption in one ATMS instance. A fuller description of the T, C, and I abstractions can be found in [Raiman *et al.*, 1991; de Kleer, 2007b; 2007a; 2007c]. The ATMS/HTMS architecture provides a unified framework to reason over any assumptions, be they about components or abstractions.

8 Related work

Automated model abstraction has a long tradition in Artificial Intelligence. The graph of models ([Addanki *et al.*, 1989]) is similar to the meta-diagnosis lattice (Figure 5) and analyzes conflicts to identify which modeling parameters to change. It is focused on design and analysis and the models that are constructed by hand. It does not use diagnosis to guide the search for models, nor is it applied to diagnosis in some domain. Work on compositional modeling ([Falkenhainer and Forbus, 1991]) also uses ATMS assumptions to represent domain abstractions and conflicts to guide the search for models. Again, the models are constructed by hand and do not use diagnosis at the domain or meta-levels. In context-dependent modeling ([Nayak, 1995]) there is typically a much larger space of model fragments to choose from and explicit context information is used to guide the selection of the domain models. The task is to construct the best causal explanation for a physical phenomena. Yet again, the models are constructed by hand and do not use diagnosis at the domain or meta-levels.

In the model-based diagnosis literature, there has been considerable work on diagnostic assumptions and selecting appropriate models for a diagnostic task [Struss, 1991b; 1991a]. This paper focuses primarily on assumptions associated with choosing domain abstractions.

There has been considerable research on structural abstraction [Chittaro and Ranon, 2004; Hamscher, 1990] where groups of components are combined to form larger systems to reduce computational complexity. [Sachenbacher and Struss, 2005] describes how the task can be used to partition the value of a variable into the qualitative values needed to solve a task. [Torta and Torasso, 2003] presents another approach to partition the value of a variable into qualitative ranges to reduce complexity when there is limited observability of the variables.

9 Conclusions

This paper has presented a general approach to selecting the best domain abstraction level to address a task and has demonstrated it within the context of digital gates. In the case of digital gates the component models can be automatically generated from the basic models using domain schemas.

10 Acknowledgments

Daniel G. Bobrow and Elisabeth de Kleer provided many useful comments.

References

[Addanki *et al.*, 1989] S. Addanki, R. Cremonini, and J. S. Penberthy. Reasoning about assumptions in graphs of models. In *Proc. 11th Int. Joint Conf. on Artificial Intelligence*, pages 1432–1438, Detroit, MI, August 1989.

[Böttcher *et al.*, 1996] C. Böttcher, P. Dague, and P. Taillibert. Hidden interactions in analog circuits. In Suhayya Abu-Hakima, editor, *Working Papers of the Seventh International Workshop on Principles of Diagnosis*, pages 36–43. Val Morin, Quebec, Canada, October 1996.

[Böttcher, 1995] Claudia Böttcher. No faults in structure? how to diagnose hidden interactions. In *IJCAI*, pages 1728–1735, 1995.

[Chittaro and Ranon, 2004] Luca Chittaro and Roberto Ranon. Hierarchical model-based diagnosis based on structural abstraction. *Artif. Intell.*, 155(1-2):147–182, 2004.

[de Kleer and Williams, 1987] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, April 1987. Also in: *Readings in NonMonotonic Reasoning*, edited by Matthew L. Ginsberg, (Morgan Kaufmann, 1987), 280–297.

[de Kleer *et al.*, 1992] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[de Kleer, 1992] J. de Kleer. A hybrid truth maintenance system. PARC Technical Report, January 1992.

[de Kleer, 2007a] J. de Kleer. Diagnosing intermittent faults. In *18th International Workshop on Principles of Diagnosis*, Nashville, USA, 2007.

[de Kleer, 2007b] J. de Kleer. Modeling when connections are the problem. In *Proc 20th IJCAI*, pages 311–317, Hyderabad, India, 2007.

[de Kleer, 2007c] J. de Kleer. Troubleshooting temporal behavior in "combinational" circuits. In *18th International Workshop on Principles of Diagnosis*, Nashville, USA, 2007.

[Falkenhainer and Forbus, 1991] B. Falkenhainer and K. D. Forbus. Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 51(1-3):95–143, 1991. Also in: J. de Kleer and B. Williams (eds.) *Qualitative Reasoning about Physical Systems II* (North-Holland, Amsterdam, 1991 / MIT Press, Cambridge, Mass., 1992).

[Hamscher, 1990] W. C. Hamscher. XDE: Diagnosing devices with hierarchic structure and known component failure modes. In *Proc. 6th IEEE Conf. on A.I. Applications*, pages 48–54, Santa Barbara, CA, March 1990.

[Nayak, 1995] P. Pandurang Nayak. *Automated Modeling of Physical Systems*. Springer Verlag LNCS 1003, Cambridge, MA, 1995.

[Raiman *et al.*, 1991] O. Raiman, J. de Kleer, V. Saraswat, and M. H. Shirley. Characterizing non-intermittent faults. In *Proc. 9th National Conf. on Artificial Intelligence*, pages 849–854, Anaheim, CA, July 1991.

[Sachenbacher and Struss, 2005] Martin Sachenbacher and Peter Struss. Task-dependent qualitative domain abstraction. *Artif. Intell.*, 162(1-2):121–143, 2005.

[Struss, 1991a] P. Struss. A theory of model simplification and abstraction for diagnosis. In *Proc. 5th Int. Workshop on Qualitative Physics*, Austin, TX, May 1991.

[Struss, 1991b] P. Struss. What's in SD? Towards a theory of modeling for diagnosis. In L. Console, editor, *Working Notes of the 2st Int. Workshop on Principles of Diagnosis*, pages 41–51. Technical Report RT/DI/91-10-7, Dipartimento di Informatica, Universitá di Torino, Torino, Italy, October 1991.

[Torta and Torasso, 2003] Gianluca Torta and Pietro Torasso. Automatic abstraction in component-based diagnosis driven by system observability. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 394–402. Morgan Kaufmann, 2003.

[Wikipedia, 2007] Wikipedia. Ring oscillator — Wikipedia, the free encyclopedia, 2007. [Online; accessed 12-February-2007].

A Common Framework for Qualitative and Quantitative Modelling

Klaus Eisenack
eisenack@pik-potsdam.de
Potsdam Institute for Climate Impact Research *

Abstract

The paper introduces model ensembles as a common framework for understanding qualitative differential equations (QDEs) and differential inclusions in a precise mathematical sense. It provides basic insights into the communalities and differences of both approaches to model under uncertainty. On this basis, a set of established methods for QDEs, some hybrid methods and standard quantitative methods can be classified, the notion of a "spurious behaviour" is clarified more thoroughly, and the importance of generality as a concept complementary to uncertainty is underpinned. Further paths for extending qualitative reasoning are outlined.

Introduction

Although much progress has been achieved in integrating qualitative differential equations (QDEs) with quantitative knowledge (Kay & Kuipers 1993; Kuipers 1994; Kay 1998; Berleant & Kuipers 1997), hybrid systems that combine different modelling approaches and types of knowledge from a coherent framework are still urgently needed (Travé-Massuyès, Ironi, & Dague 2004; Price *et al.* 2005). There is no common mathematical theory to my knowledge that describes, e.g. qualitative, semi-qualitative, set-valued, interval-based and order of magnitude reasoning. Relevant approaches as those of Bradley *et al.* (2001) integrate various reasoning techniques hierarchically in a more pragmatic way.

Up to now, it has been an open issue whether QDEs and differential inclusions (DIs) are essentially the same way of representing uncertainty (Kuipers 2000; Saint-Pierre 2004), although methods as Q3 closely resemble the numerical analysis of differential inclusions by considering numerical envelopes on functions and landmarks in an efficient way (Berleant & Kuipers 1997). DIs represent a similar approach to account for uncertainty (Aubin & Cellina 1984), since contingent dynamics can be computed even if no probabilistic knowledge is available. The question is whether QDEs can be mathematically described by differential inclusions, which would provide a valueable bridge between set-valued analysis and computer sience. The works of Dordan (1992; 1995), Aubin (1996), and Hüllermeyer (1997) make considerable steps in that direction. Based on an ordinary differential equation $\dot{x} = f(x)$ they introduce the concept of a monotonic cell, consisting of all states x such that $f(x)$ has a given sign vector. A trajectory can be described qualitatively by the sequence of the monotonic cells it visits. By imposing additional restrictions on f, these authors investigate the issue of the existence of solutions more thoroughly than in the literature from computer science. They also generalise the approach to other partitions of the state space than by signs, called qualitative frames. However, the approach is more restrictive in that only single ordinary differential equations (ODEs) are considered. This is interesting in itself, but not sufficient for many applications of QDEs where uncertainties have to be taken into account.

Another issue which seems unrelated at the first glance is the necessary existence of spurious behaviour (Say & Akin 2002). A spurious behaviour doesn't match the quantitative solution of an ODE covered by the simulation input. Such behaviour is traced back (i) to the impossibility to represent certain types of irrationals, (ii) to the well-known ambiguities of sign algebra, and (iii) generally to the incompleteness of the information about a system that is modelled as a QDE. It is yet unclear what kind of application-oriented models could bring about such paths in a relevant way. Answering such questions requires a precise notion of spurious behaviour that can ideally be linked to established mathematical theory.

This paper formalizes basic ideas about "incomplete knowledge" in a precise sense to clarify the discussion by introducing the general framework of *model ensembles* which includes ODEs, QDEs, differential inclusions, causal loop diagrams and further methods as special cases. Thus the relation between QDEs and differential inclusions can be clarified. Technically, a model ensemble is a (possibly infinite) set $\mathcal{M}$ of functions, where each $f \in \mathcal{M}$ constitutes an ordinary differential equation $\dot{x} = f(x, t)$. By considering not a single model but a whole ensemble of models, a variety of possible system configurations which we can think of under uncertainty is covered. Although not systematized as here, such a style of reasoning is also common, e.g. for param-

*I whish to thank Patrick Saint-Pierre and Matthias Lüdeke for valuable discussions. I am grateful to Gerhard Petschel-Held who strongly influenced this work. He suddenly died in his office in September 2005. Further thanks go to two anonymous referees for valuable comments. This work was supported by the Deutsche Forschungsgemeinschaft ("Viabilität unter Unsicherheit"), and the EU Research Training Network HPRN-CT-2002-00281.

eter variation (e.g. Stainforth *et al.* 2005), model comparison (e.g. Gregory *et al.* 2005), and scenario development (e.g. Nakićenović *et al.* 2000; Millennium Ecosystem Assessement 2005; Swart, Raskin, & Robinson 2004). These basic ingredients are not new, but to my knowledge were never published. Based on this it is shown for a large set of qualitative models that no path of length 2 in an envisionment graph is spurious. At the same time, the definitions contribute to specify new hybrid qualitative-quantitative approaches.

Although a theoretical paper with a broader scope, this work is motivated by applications from sustainability science. The research about sustainable development aims to meet current human needs while maintaining the environment and natural resources for future generations (WCED 1987). In this domain uncertainties about dynamic social-ecological systems pose major challenges, and typologies of such systems and to understand so-called syndromes of global environmental change (Schellnhuber, Lüdeke, & Petschel-Held 2002; Petschel-Held 2005) are an important research field. In this context QDEs are a very valuable tool to analyse causal loop diagrams (Forrester 1968; Sterman 2000) and to deal with uncertainty, generality and non-quantitative knowledge (Petschel-Held *et al.* 1999; Stave 2002).

In the next section, the framework of model ensembles if formally introduced and illustrated with some more conventional examples. Then, basic defintions of QDEs are reformulated using graph theoretical concepts and the framework, including a discussion of spurious behaviour. Subsequently, differential inclusions are recalled and formulated as model ensembles, allowing for a thorough comparison with QDEs. Before concluding, further implications are discussed.

Model Ensembles

A **model ensemble** $\mathcal{M}$ is defined as a set of functions $f : X \times \mathbb{R}_+ \to \mathbb{R}^n$ on a **state space** $X \subseteq \mathbb{R}^n$. These functions are called **models**. In the case of uncertainties, each describes a possible configuration of a real-world system which must be considered. The set $\mathcal{E}$ contains functions $x(\cdot) : \mathbb{R}_+ \to X$, being the space of **admissible trajectories** of the systems, e.g. $\mathcal{E} = C^1(\mathbb{R}_+, X)$. Each model $f \in \mathcal{M}$ defines a family of initial value problems

$$\begin{aligned} \dot{x} &= f(x,t), \\ x(0) &= x_0, \end{aligned}$$

with $x_0 \in X$. It is also possible to consider model ensembles which only contain autonomous models.

Of course, the systems of the model ensemble have (in general) different solutions. Thus, a *set* of trajectories must be assigned to each initial value x_0. The set-valued **solution operator** $\mathcal{S}_\mathcal{M}(\cdot) : X \to \mathcal{P}(\mathcal{E})$ (of a model ensemble $\mathcal{M}$ with respect to a state space X and admissible trajectories $\mathcal{E}$), assigning to an initial state a subset of $\mathcal{E}$, is defined by

$$\begin{aligned} \mathcal{S}_\mathcal{M}(x_0) := \{ & x(\cdot) \in \mathcal{E} \mid \\ & x(0) = x_0, \\ & \exists f \in \mathcal{M} \forall t \in \mathbb{R}_+ : \dot{x}(t) = f(x(t),t) \}. \end{aligned}$$

Depending on $\mathcal{E}$ it may be sufficient that the ODE only holds almost everywhere. The solution operator is written with the initial state x_0 as argument to investigate how properties of the solutions change in different subsets of the state space (see the section on further applications below and Eisenack (2006)). If we are interested in all possible initial states, we take the whole state space X as argument and call the elements of $\mathcal{S}_\mathcal{M}(X)$ the **solutions of the model ensemble** $\mathcal{M}$. The solution operator and the way we denote it is also resembles the concept of an evolutionary system as defined by Aubin (2001). The main challenge in reasoning with model ensembles is to find relevant structure in $\mathcal{S}_\mathcal{M}(X)$. This includes

1. representing a model ensemble in a way which is adequate to the modeller and allows for a formal treatment,
2. efficient algorithms to determine $\mathcal{S}_\mathcal{M}(X)$ from a (possibly infinite) model ensemble $\mathcal{M}$,
3. detecting structural features of the solutions of the model ensemble.

We now provide some examples for model ensembles.

EXAMPLE 1: Let $\mathcal{M}$ contain only one function $f : X \times \mathbb{R}_+ \to \mathbb{R}^n$ which is Lipschitz on X, and let the admissible trajectories be $\mathcal{E} = C^1(\mathbb{R}_+, X)$. Then, $\mathcal{S}_\mathcal{M}(x_0)$ contains the usual solutions of the initial value problem with $x(0) = x_0$ which exist on $\mathbb{R}_+$. □

EXAMPLE 2: Given a function $f' : X \times \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n, (x,t;p) \mapsto f'(x,t;p)$, depending on a parameter vector p, and a finite set P of possible parameterisations, define the finite model ensemble

$$\mathcal{M} := \{ f \in C(X \times \mathbb{R}_+, \mathbb{R}^n) \mid f(x,t) = f'(x,t;p), p \in P \}.$$

Then, the solution operator with respect to a set of admissible trajectories provides all "scenario runs" for the different parameterisations. □

EXAMPLE 3: A causal loop diagram (in its simplest form) is a directed graph with marked edges. Each vertex represents a variable, and each edge an influence of the source variable on the target variable which can be marked as positive or as negative. In traditional systems dynamics modelling (Forrester 1968; Sterman 2000), the causal loop diagram is a starting point to develop a quantitative model, usually in the form of an ordinary differential equation (ODE). Since the diagram only contains qualitative information, there is an infinite number of such ODEs for a given diagram. For example (cf. Richardson 1986), $\mathcal{M}$ can be defined as the set of all ODEs $\dot{x} = f(x)$ with state vector $x \in \mathbb{R}^n$ for which the signs of the partial derivatives $\left[D_j f_i(x) \right]$ correspond to the signs of the edges ($D_j f_i$ denoting the partial derivative of the ith component of f with respect to x_j). If there is no edge between two variables, the partial derivative vanishes. □

In the next sections, QDEs and DIs are introduced as model ensembles.

QDEs as Model Ensembles

For the sake of simplicity I introduce a simplified version of QDEs that only considers monotonic influences on the change of variables. For this model class only the velocity, but not the state space needs to be investigated. The extension to complete QDEs with landmarks is straightforward but very technical insight (see Eisenack 2006 for a treatment). There is one other difference to the original work of Kuipers (1994): the focus is on time-interval states, while time-point states are not represented explicitly. This has the advantage that solutions of QDEs can be displayed in a much more accessible form (Eisenack & Petschel-Held 2002).

At first we specify the kind of model ensemble which constitutes a QDE. By $\mathcal{A} := \{[+], 0, [-]\}$ we denote the domain of signs, and by $\mathcal{A}_* := \{[+], 0, [-], [?]\}$ the domain of extended signs. Qualitative equality is denoted by $\approx$. We will aditionally use tuples and matrices of (extended) signs, and extend the sign operator $[\cdot]$ and qualitative equivalence component wise. Now a model ensemble can be defined:

DEFINITION 1: *For a given $n \times n$ matrix of signs $\Sigma = (\sigma_{i,j})_{i,j=1,\dots,n}$, $\sigma_{i,j} \in \mathcal{A}_*$, and a state space $X \subseteq \mathbb{R}^n$ we define the* ***monotonic ensemble***

$$\mathcal{M}(\Sigma) := \{f \in C^1(X, \mathbb{R}^n) \mid \forall x \in X : \left[\mathcal{J}(f)(x)\right] \approx \Sigma\},$$

where $\mathcal{J}(f)$ denotes the Jacobian of f. We call a function $x(\cdot) \in C^1([0,T], \mathbb{R}^n)$, possibly $T = \infty$, ***reasonable*** *under the usual conditions, and define the space of admissible trajectories $\mathcal{E}$ by all reasonable functions with values in X. We call the systems of the model ensemble $\mathcal{M}(\Sigma)$ a* ***QDE****.*

A monotonic ensemble $\mathcal{M}(\Sigma)$ is a model ensemble which only contains autonomous models. Although a set of ODE systems is not an equation we use this designation in analogy to Kuipers (1994). One reason of the original terminology might be that a QDE can be "solved" by considering a constraint satisfaction problem, i.e. a relational equation over a finite set.

Based on DEF. 1, a set-valued solution operator $\mathcal{S}_{\mathcal{M}(\Sigma)}(\cdot)$ is defined. The set of solutions of the monotonic ensemble $\mathcal{S}_{\mathcal{M}(\Sigma)}(X)$ contains all reasonable solutions of all ODE systems contained in the QDE. It should be noted that the properties of the monotonic ensemble are not sufficient to guarantee a global solution for every $f \in \mathcal{M}(\Sigma)$.

Admissible trajectories are discretized as usual by tracking the sign vectors $[\dot{x}(t)]$ for each solution:

DEFINITION 2: *For a given reasonable function $x(\cdot)$ on $[0,T]$ we have an ordered sequence of sign jump points (t_j) with $t_0 = 0$ which subsequently contains all boundary points of the closures of all sets $\{t \in [0,T] | [\dot{x}(t)] = v\}$ with $v \in \{[-], [+]\}^n$. We construct a sequence of sign vectors $\tilde{x} = (\tilde{x}_j) := \left(\left[\dot{x}(\tau_j)\right]\right)$, where we arbitrarily choose $\tau_j \in (t_j, t_{j+1})$. If the sequence (t_j) is finite with m elements, we choose $\tau_m \in (t_m, T)$. The sequence $\tilde{x}$ is called* ***abstraction*** *of $x(\cdot)$.*

The slight difference compared to the standard definitons is that, e.g. times were trajectories go through a saddle point are ignored. Note that the abstraction $\tilde{x}$ does not depend the concrete values $\tau_j \in (t_j, t_{j+1}), j \in \mathbb{N}$, since the sign vector $\left[\dot{x}(t)\right]$ is constant on any interval (t_j, t_{j+1}). The set of the abstractions of all solutions of a monotonic ensemble are entailed by a finite graph in the following way:

DEFINITION 3: *Let $\mathcal{M}(\Sigma)$ be a monotonic ensemble, $\mathcal{E}$ the set of reasonable trajectories and $\mathcal{S}_{\mathcal{M}(\Sigma)}(\cdot)$ the corresponding solution operator. We denote the set of the abstractions of the solutions by*

$$\begin{aligned}\tilde{\mathcal{S}}_{\mathcal{M}(\Sigma)}(v_0) := \{\tilde{x} \mid & \exists x_0 \in X \text{ with } [x_0] \approx v_0, \\ & \exists x(\cdot) \in \mathcal{S}_{\mathcal{M}(\Sigma)}(x_0) : \\ & \tilde{x} \text{ is the abstraction of } x(\cdot)\}.\end{aligned}$$

Then, the directed ***state-transition graph*** *G of the monotonic ensemble is defined by the vertices*

$$V(G) := \{v \in \mathcal{A}^n \mid \exists\, \tilde{x} \in \tilde{\mathcal{S}}_{\mathcal{M}(\Sigma)}(\mathcal{A}^n), j \in \mathbb{N} : \; \tilde{x}_j = v\},$$

called ***qualitative states****, and the edges*

$$\begin{aligned}E(G) := \{(v,w) \mid & \exists\, \tilde{x} \in \tilde{\mathcal{S}}_{\mathcal{M}(\Sigma)}(\mathcal{A}^n), j \in \mathbb{N} : \\ & \tilde{x}_j = v \text{ and } \tilde{x}_{j+1} = w\},\end{aligned}$$

called ***qualitative transitions****.*

For convenience, the state-transition graph of a monotonic ensemble is also called the state-transition graph of a QDE. Thus, we have defined a directed graph G such that all sequences of $\tilde{\mathcal{S}}_{\mathcal{M}(\Sigma)}(\mathcal{A}^n)$ describe a path in G, i.e. the graph completely covers all reasonable solutions of initial value problems $\dot{x} = f(x), x(0) = x_0$ with $f \in \mathcal{M}(\Sigma)$. Note that G is loop free since subsequent coefficients of the abstraction of a reasonable function are different. Note further that formalizing the state-transition graph in that way does not require a completeness proof, since it is complete by definition. Completeness can only be shown for an algorithm that computes the graph (e.g. the QSIM algorithm). In our framework this requires to prove that at least a supergraph, but definitively not a subgraph is determined. Within this framework spuriousity is defined as follows:

DEFINITION 4: *Let G be the state-transition graph of the monotonic ensemble $\mathcal{M}(\Sigma)$. In G a path $v_1, \dots, v_n$ of lenght n is called* ***spurious*** *if there is no model $f \in \mathcal{M}(\Sigma)$ and no initial velocity $\dot{x}_0$ with $[\dot{x}_0] = v_1$ such that the solution $x(\cdot)$ to the initial value problem $\dot{x} = f(x), \dot{x}(0) = \dot{x}_0$ has $v_1, \dots, v_n$ as the first n coefficients of its abstraction $\tilde{x}$.*

We discuss the occurence of spurious behaviour in the exact state-transition graph below.

Even without running the QSIM algorithm, some features can be shown directly. Which vertices occur in a state-transition graph? Most basically, $\{[-], [+]\}^n \subseteq V(G)$ due to the following reasons: by chain rule $\ddot{x} = \mathcal{J}(f)(x) \cdot \dot{x}$, such that for assumptions about the sign matrix $\left[\mathcal{J}(f)(x)\right]$ not all sign vectors $[\dot{x}]$ are consistent with all sign vectors $[\ddot{x}]$. However, since no claims about $[\ddot{x}]$ are made, no $[\dot{x}] \in \{[-], [+]\}^n$ can be excluded from being a vertex. The situation is more complicated if some $\dot{x}_i \equiv 0$ on (t_j, t_{j+1}), which implies that also $\ddot{x}_i \equiv 0$ on the same interval.

I now present a necessary criterion for such a vertex to exist (see Eisenack 2006 for a proof). For this, we need the set $Z_0(v) := \{i = 1, \dots, n | v_i = 0\}$, which assigns to a sign vector $v \in \mathcal{A}^n$ the indices of vanishing components.

PROPOSITION 1: *If $v \in V(G)$, then for all $i \in Z_0(v)$*

$$\exists j, k \notin Z_0(v), j \neq k : 0 \neq \sigma_{i,j} v_j \approx -\sigma_{i,k} v_k \neq 0$$

or $\forall j \notin Z_0(v) : \sigma_{i,j} = 0$.

Additionally, every state-transition graph contains the vertex 0, representing the equilibria of systems of the monotonic ensemble.

Now I will show a characterisation for the existence of edges in the state-transition graph G. It is simplified by considering only vertices with non-vanishing components. When two qualitative states v, w differ only in one component i, there must be a solution of the monotonic ensemble $x(\cdot)$, defined by a model f, which transgresses the main isocline $f_i(x) = 0$ at some time, because this isocline separates the regions of the phase space where $[f(x)] = v$ and $[f(x)] = w$, respectively. A necessary condition for such a transgression is an appropriate sign of $\ddot{x}_i$ on the main isocline, e.g. if $v_i = [-]$ and $w_i = [+]$, then $[\ddot{x}_i] \approx [+]$ is needed. We define the **intermediate state** $v \wedge w$ for $v, w \in \mathcal{A}^n$ by

$$(v \wedge w)_i := \begin{cases} v_i & \text{if } v_i = w_i, \\ 0 & \text{if } v_i \neq w_i. \end{cases}$$

Thus, $Z_0(v \wedge w)$ are the indices of the components which change from v to w (or which are constant in one or both states).

PROPOSITION 2: *Let $v, w \in V(G)$, $v \neq w$, and $Z_0(v) = Z_0(w) = \varnothing$. Then, $(v, w) \in E(G)$ iff*

$$\forall i \in Z_0(v \wedge w)\, \exists j \notin Z_0(v \wedge w) : w_i \cdot (v \wedge w)_j \approx \sigma_{i,j}.$$

For a detailed proof see Eisenack (2006). Here it is important to notice that the proposition is a full characterisation. It is not only a criterion for determining the edges of the state-transition graph, but also shows that every edge (v, w) in the graph actually corrresponds to at least one model in $\mathcal{M}(\Sigma)$ which visits the qualitative states v and w in that temporal order. The main part of the proof is thus to construct an appropriate model and to show that it is an element of the monotonic ensemble. The consequence is that (at least for QDEs described by a monotonic ensemble) subsequent time-interval states computed by the QSIM algorithm are never spurious.

Problems arise, of course, for paths of length 3 or more. However, if we extend the definition of monotonic ensemble towards

$$\mathcal{M}(\Sigma) := \{f \in C^1(X, \mathbb{R}_+, \mathbb{R}^n) \,|\forall x \in X, t \in \mathbb{R}_+ : \left[\mathcal{J}(f)(x,t)\right] \approx \Sigma\},$$

it is expected that the situation changes dramatically. Since the non-autonomous system can, in principle, switch between the models constructed in the above proof at every qualitative state, *every path* of *arbitrary length* corresponds to at least one model. In that sense, there are no spurious behaviours.

It may be questioned whether these results still hold when full QDEs and not only monotonic ensembles are considered. There are no proofs yet, but the extension seems straightforward – although a lot of cases need to be distinguished. For illustration, an example for a model ensemble containing a landmark (λ) and an algebraic constraint ($+$) is (see Eisenack *et al.* 2007 for further examples)

$$\begin{aligned}\mathcal{M}(\Sigma_1, \Sigma_2) := \{f \in & C^1(\mathbb{R}^2 \times \mathbb{R}_+, \mathbb{R}^2) \mid \\ & \exists \lambda \in \mathbb{R} \forall x \in \mathbb{R}^2, t \in \mathbb{R}_+ \\ & \text{with } x_1 \leq \lambda : \left[\mathcal{J}(f)(x,t)\right] \approx \Sigma_1 \\ & \text{and with } x_1 > \lambda : \left[\mathcal{J}(f)(x,t)\right] \approx \Sigma_2 \\ & \text{and } f_1(x,t) = x_1 + x_2.\}\end{aligned}$$

This example also illustrates the need for precise definitions to be clear about what is meant by a spurious behaviour. Here, addition of real numbers is used in the definition of $\mathcal{M}(\Sigma_1, \Sigma_2)$ for the algebraic constraint. The model ensemble would be much larger if in the last line of the definition, addition is used in the qualitative sense, i.e.

$$\forall x \in \mathbb{R}^2, t \in \mathbb{R}_+ : [Df_1(x,t)] \approx [++], \qquad (1)$$

which would, depending on Σ_1 and Σ_2, be either contradictory or redundant.

The Relation between QDEs and Differential Inclusions

Differential inclusions (DIs) are a generalisation of ordinary differential equations. While an ODE assigns a single velocity to points in the state space, for differential inclusions *multiple* velocities can be assigned. We map a state x to a set of possible velocities $F(x)$, and admit a trajectory $x(\cdot)$ as a solution, if $\dot{x}(t)$ is always an element of $F(x(t))$. As in the case of QDEs we cannot generally expect to obtain unique solutions in such a setting, yielding a set-valued solution operator. The first ideas to this approach arose in the 30s of the last century (Zaremba 1936; Marchaud 1934). A broad overview to the fundamentals and subsequent development of the theory is provided by Aubin (1984). Differential inclusions are applied to problems from, e.g. population dynamics (Křivan & Colombo 1998; Guo, Xue, & Li 2003), physics (Maisse & Pousin 1997), climate change (Chahma 2003), differential games (Chodun 1989; Ivanov & Polovinkin 1995) and natural resource management (Bene, Doyen, & Gabay 2001; Cury *et al.* 2005; Eisenack, Scheffran, & Kropp 2006).

One basic motivation – similar to QDEs – is to consider uncertainties which cannot be expressed in a probabilistic way. We may have an ODE $\dot{x} = f(x, t; u)$, depending on a parameter or a control u. If we do not know u exactly but can restrict the value, say, to an interval J such that $u \in J$, we obtain a set of possible values $F(x,t) := \{f(x,t;u) \mid u \in J\}$. We can formulate this as an infinite monotonic ensemble in the following way. For a given autonomous measurable function $f' : X \times U \to \mathbb{R}^n, (x,u) \mapsto f'(x,u)$, where $U \subseteq \mathbb{R}$ is a given interval of control values, set

$$\mathcal{M} := \{f : X \times \mathbb{R}_+ \to \mathbb{R}^n \text{ measurable } \mid f(x,t) = f'(x, u(t)), u(t) \in U\}.$$

Taking absolutely continuous functions as admissible trajectories, the solution operator $\mathcal{S}_{\mathcal{M}}(x_0)$ describes all trajectories starting from x_0 which result from any measurable open-loop control $u(\cdot) : \mathbb{R}_+ \to U$.

In the set-valued standard definition, for a given set-valued map $F : X \to \mathcal{P}(Y)$ (where Y is the velocity space and $\mathcal{P}()$ denotes the power set), an "equation" of the form

$$\dot{x} \in F(x),$$
$$x(0) = x_0,$$

is called a differential inclusion. In most cases an absolutely continuous function $x(\cdot) : I \to X$ on an interval $I = [0, T]$, possibly $T = \infty$ is called a **solution** if $x(0) = x_0$ and $\dot{x}(t) \in F(x(t))$ almost everywhere on I. There are various theorems on the existence of solutions to a differential inclusion (see e.g. Aubin 1991).

From this general perspective, a set-valued map $F : X \to \mathcal{P}(\mathbb{R}^n)$ defines a model ensemble by

$$\mathcal{M} := \{f : X \times \mathbb{R}_+ \to \mathbb{R}^n \mid f(x,t) \text{ measurable with respect to } t \text{ and } \forall t \in \mathbb{R}_+ : f(x,t) \in F(x)\}.$$

Taking the set of absolutely continuous functions on intervals $I = [0, T]$ as space of admissible trajectories $\mathcal{E}$, we obtain a set-valued solution operator

$$\mathcal{S}_F(x_0) := \{x(\cdot) \in \mathcal{E} \mid x(0) = x_0, \exists f \in \mathcal{M} : \dot{x}(t) = f(x(t), t) \text{ almost everywhere } \}.$$

Can a QDE be "simulated" by a DI? To find all possible trajectories which can be brought about by a simple QDE (we stick to the case without landmarks again), we change the perspective from the state space to the velocity space. We could define a set-valued map by $F(x) := \{f(x) \mid f \in \mathcal{M}\}$ such that the solutions of the differential inclusion describe all trajectories. However, if the QDE is specified by a sign matrix $\Sigma = (\sigma_{i,j}) \in \mathcal{A}_*^{n \times n}$, we run into trouble, as the following shows:

Suppose that $f \in \mathcal{M}(\Sigma)$. Since it follows from $\dot{x} = f(x)$ that $\ddot{x} = \mathcal{J}(f)(x) \cdot \dot{x}$, we obtain a second order differential inclusion in the joint state and velocity space:

$$\ddot{x} \in F(\dot{x}, x),$$
$$F : (\dot{x}, x) \mapsto \{\mathcal{J}(f)(x) \cdot \dot{x} \mid f \in \mathcal{M}(\Sigma)\}.$$

This can be simplified to

$$\ddot{x} \in \hat{F}(\dot{x}) := \{M\dot{x} \mid [M] \approx \Sigma\},$$

where M denotes $n \times n$ matrices over the real numbers. We observe that the components $i = 1, \ldots, n$ of $\hat{F}(\dot{x})$ evaluate to

$$\hat{F}_i(\dot{x}) = \begin{cases} 0 & \text{if } \forall j = 1, \ldots, n : \dot{x}_j \cdot \sigma_{i,j} = 0, \\ \mathbb{R}_+ \setminus \{0\} & \text{else if } \forall j = 1, \ldots, n : [\dot{x}_j] = \sigma_{i,j} \neq 0 \text{ or } \dot{x}_j \cdot \sigma_{i,j} = 0, \\ \mathbb{R}_- \setminus \{0\} & \text{else if } \forall j = 1, \ldots, n : -[\dot{x}_j] = \sigma_{i,j} \neq 0 \text{ or } \dot{x}_j \cdot \sigma_{i,j} = 0, \\ \mathbb{R} & \text{otherwise .} \end{cases}$$

Except the trivial case, this unbounded set-valued map is very irregular and allows for a very broad solution set. This simple approach doesn't provide valuable results.

One way to overcome this are linear-interval differential inclusions, which restrict a monotonic ensemble $\mathcal{M}(\Sigma)$ to models for which prescribed interval constraints, given by set-valued maps, hold for the components of the Jacobian.

DEFINITION 5: *Let U be a matrix of compact intervals $(u_{i,j})_{i,j=1,\ldots,n}$, where each interval either vanishes or does not contain 0. A set-valued map $F : X \to \mathcal{P}(\mathbb{R}^n), F(x) := Ux$, where the latter denotes interval-valued multiplication, is called a* ***linear-interval map****.*

Interval-valued multiplication is defined in the usual way by $Ux := \{Mx \mid M \in U\}$, where a matrix $M = (m_{i,j})_{i,j=1,\ldots,n} \in U$ if and only if $\forall i, j = 1, \ldots, n : m_{i,j} \in u_{i,j}$. We regard singletons as intervals. DEF. 5 guarantees that every coefficient of U has a unique sign (which can be related to the coefficients of Σ). Note that a linear-interval map F defines a model ensemble which includes nonlinear models f such that $\forall x \in X : f(x) \in F(x)$.

We saw above that it is not possible to investigate a QDE by considering a differential inclusion $\ddot{x} \in \hat{F}(\dot{x})$. However, if intervals $u_{i,j}$ are known such that $\forall x \in X : D_j f_i(x) \in u_{i,j}$, the linear-interval differential inclusion

$$\ddot{x} \in F(\dot{x}) = U\dot{x},$$

can be set-up. It is very regular and simulates the monotonic ensemble $\mathcal{M}(\Sigma)$ in the following sense. Define the restricted model ensemble

$$\mathcal{M}'(\Sigma, U) := \{f \in M(\Sigma) \mid \forall x \in X : \mathcal{J}(f)(x) \in U\} \subseteq \mathcal{M}(\Sigma).$$

with the solution operator $\mathcal{S}_{\mathcal{M}'(\Sigma,U)}(\cdot)$. Then $\forall x_0 \in X, x(\cdot) \in \mathcal{S}_{\mathcal{M}'(\Sigma,U)}(x_0) : \dot{x}(\cdot) \in \mathcal{S}_F(\dot{x}(0))$. On the other hand, the differential inclusion also covers solutions of non-autonomous ODEs $\dot{x} = f(x, t)$ with $\mathcal{J}(f)(x, t) \in U$ for all $t \in \mathbb{R}_+$. Linear-interval differential inclusions are more general than QDEs in the sense that they also include non-autonomous models, and are more specific in the sense that they only include bounded models. In contrast, qualitative differential equations are deterministic but subsume a broad set of possible configurations.

Discussion and Further Applications

In many applications of qualitative reasoning the discussion of spurious behaviour is mixed with the existence of qualitative trajectories of a QDE which contradict knowledge about the system available to the modeller that is not expressed by qualitative constraints. If this impression is true, one explanation are the roots of the method in qualitative *physics*, where we construct problem-driven models of physical systems. They are perceived as being ontologically unambigous, completely numerically specified and time-invariant. From this viewpoint, the main reason for qualitative modelling are epistimic limitations, i.e. missing knowledge about the "objective" pysical system ("uncertainties"), or efficiency considerations when it is not needed to

have access to the complete "objective" system for solving a particular task. By interpreting a model ensemble as uncerainty, it covers all cases that could be potentially considered as being valid due to pragmatic or epistemic limitations.

However, the formalization of QDEs as model ensemble illustrates a further interpretation which is highly relevant in the domain of sustainablity science that motivated this work: it may be that there are multiple non-indentical systems, e.g. social-ecological systems like fisheries, agricultural systems or bioreserves which re-appear in many instances on the world. Although every such case may be different, it often appears that some of them share crucial properties and exhibit certain patterns (e.g. qualitative behaviour with typical temporal logic properties). Then, a model ensemble is the collection of all cases which have to be analysed. In that sense, QDEs do not only represent uncertainty, but also *generality*. Such models are not only meant to provide insights for single applications, but should also apply to a broader set of cases with general features in common. In other words, while resolving uncertainty would in principle lead to narrowing down a QDE until it would be refined to an ODE, for representing generality we do not aim at refining to that point, so that the model ensemble still subsumes a broad range of systems. Such generalised ensembles can be so-called "archetypes" of global environmental change (Eisenack, Lüdeke, & Kropp 2006).

Within the domain of sustainability science it is also unappropriate only to consider autonomous ODEs as constituents of the model ensembles, since social-ecological systems are usually influenced by exogeneous environmental and cultural factors which are not constant in time. Therefore, the extended definition of QDEs above may be more appropriate for these kind of applications, at the same time resolving the problem of spurious behaviour. I expect that this interpretation can also be valuable for other domains where the analysis or design of whole classes of systems is needed.

Once we adopt this viewpoint, further questions can be posed within the framework of model ensembles and some established tractability methods can be described. If it is not possible to find relevant features common to all solutions of a model ensemble $\mathcal{M}$ we can try to identify subsets $\mathcal{M}' \subseteq \mathcal{M}$ for which such robust properties exist. The characterisation of $\mathcal{M}'$ is associated with the discovery of structural features which e.g. bring about problematic or desirable system behaviour. In other terms, conditions under which certain (sub)pattern evolve are found. If $\mathcal{M}$ is partially determined by certain control measures imposed on the system, and $\mathcal{M}'$ by alternative control measures, the differences between the solution operators $\mathcal{S}_{\mathcal{M}}(\cdot)$ and $\mathcal{S}_{\mathcal{M}'}(\cdot)$ are of interest.

There are cases where solutions of a model ensemble are artifacts from the assumptions the modeller made. Then it is important to restrict the analysis so that the artifacts are eliminated. Generally, a **restriction** means a restriction of the model ensemble to some $\mathcal{M}' \subseteq \mathcal{M}$, of the admissible trajectories to some $\mathcal{E}' \subseteq \mathcal{E}$, or of the state space to some $X' \subseteq X$. Very "unlikely" or "irrelevant" cases which cannot be refuted on base of the original model ensemble are further reasons to restrict $\mathcal{M}$ or even $\mathcal{E}$. The analytical function constraint and phase plane constraints are examples for the latter, while filtering marginal cases (Eisenack & Petschel-Held 2002; Bouwer & Bredeweg 2002) for the former.

The concept of restricting a model ensemble can also be seen as a formalization of finding the best level of abstraction for practical engineering problems. Qualitative modelling can start with a very general model ensemble which is then successively restricted only up to the level where it becomes concrete enough to achieve its intended task.

Finally, the perspective of model ensembles opens the view for established as well as potential future hybrid or semi-qualitative modelling techniques. One basic motivation for such hybrid methods is to include more than just monotonicity assumptions about a system, if they are available. For example, NSIM restricts $\mathcal{E}$ by introducing envelopes on the solutions (Kay & Kuipers 1993), while Q2 restricts $\mathcal{M}$ to those models where landmarks are (constant) within prescribed quantitative intervals and monotonic functions within monotonic envelopes (Kuipers 1994). Q3 basically remains within this specification but developes more powerful reasoning techniques to determine solutions (Berleant & Kuipers 1997). All these methods, although deterministic in their core, are close to the ideas of differential inclusions. They may be improved by using more results from the respective numerical analysis.

As a new semi-qualitative example I outline a technique which is based on the considerations of the last section (see Eisenack 2006 for details). After solving a QDE with sign matrix Σ, quantitative bounds are considered by setting up a linear-interval differential inclusion $\ddot{x} \in F(\dot{x}) = U\dot{x}$ where the signs of the intervals coefficients of U correspond to the signs of Σ. If we want to identify conditions for a given successor state w to be reached from a state v, we define – in the velocity space – the cones $K(v) := \{\dot{x} \in \mathbb{R}^n \mid [\dot{x}] = v\}$ for $v \in \mathcal{A}^n$. For the linear-interval differential inclusion, the so-called absorption basin $\mathrm{Abs}_F(\bar{K}(v), \bar{K}(v) \cap \bar{K}(w))$ of the closure of such cones contains all initial velocities $\dot{x}_0$ such that for all solutions $\dot{x}(\cdot) \in \mathcal{S}_F(\dot{x}_0)$ with $[\dot{x}_0] = v$ there exists a $T > 0$ with $[\dot{x}(T)] = w$. Such absorption basins can be computed using the viability algorithm (Saint-Pierre 1994; Cardaliaguet, Quincampoix, & Saint-Pierre 1994).

All these methods share the idea to complement qualitative knowledge in the sense of monotonicities and landmarks with quantitative knowledge in the sense of ODEs or set-valued maps. However, we may think of further possibilities. Often more knowledge than about trends and thresholds seems to be available, while it is very difficult to come up with quantitative estimates. This may be due to very poor data conditions (e.g. agricultural yield in developing countries, fish catches in international waters) or due to difficulties in operationalizing variables (e.g. political power or poverty). *It would therefore be of high value to refine model ensembles without resorting to quantities*, raising the question whether there is some relevant type of non-quantitative knowledge that cannot be represented as a QDE. Ordner of magnitudes may be a candidate, but established formalizations still refer to magnitudes on the real line (Travé-Massuyès, Ironi, & Dague 2004).

A further candidate are ordinal assumptions. These relate to the strength of influences in causal loop diagrams. If the influence of one variable on another is stronger than the influence of a third, this can be interpreted as partial order on the partial derivatives of models f of a monotonic ensemble $\mathcal{M}(\Sigma)$ of the form

$$\forall x \in X : D_k f_i(x) > D_l f_j(x),$$

for a set of tuples (i, j, k, l). The restricted model ensemble $\mathcal{M}' \subseteq \mathcal{M}(\Sigma)$ contains only those models that respect a prescribed partial order of this kind. Eisenack et al. (2006; 2007) present some methods to exploit such kind of knowledge. However, it appears that not all implications that can be made from ordinal assumptions are exploited yet. Although some theoretical results exists, well working algorithms are not established yet. Finally, ordinal assumptions obviously boil down to statements about the sign $\left[d_{k,l}^{i,j}\right]$, where $d_{k,l}^{i,j} := D_l f_j \cdot D_k f_i - D_k f_j \cdot D_l f_i$. We can thus re-state the question of whether there is some kind of information "between signs and numbers".

Conclusion

In this paper I presented a formalisation of QDEs within the new framework of model ensembles which appears to embed also differential inclusions and further established quantiative and semi-qualitative methods.

Their particular similarities and differences become visible. While QDEs are deterministic and autonomous, differential inclusions also include non-autonomous dynamics. On the other hand, QDEs are less restrictive in the sense that they do not need to be explicitly quantitatively constrained by set-valued maps. Therefore, neither of these approaches can be reduced to the other.

As a by-product the notion of spurious behaviour can be further clarified. It is shown that a qualitative behaviour consisting of two subsequent time-interval states is never spurious. Furthermore, there are indications that an extension of standard QDEs to non-autonomous models of a certain kind may completely resolve this problem.

The framework of model ensembles can be used to specify the notion of uncertainty typically used in qualitative reasoning and extends it to the notion of generality in a certain sense that is highly relevant for the design of whole classes of systems which only share some common features.

Finally, the framework of model ensembles allows for defining various extensions of QDEs in a consistent way, opening the field for further qualitative and semi-qualitative methods.

References

Aubin, J.-P., and Cellina, A. 1984. *Differential Inclusions. Set-Valued Maps and Viability Theory.* Berlin: Springer Verlag.

Aubin, J.-P. 1991. *Viability Theory*. Basel: Birkhäuser.

Aubin, J.-P. 1996. *Neural Networks and Qualitative Physics*. Cambridge: Cambridge University Press.

Aubin, J.-P. 2001. Viability kernels and capture basins of sets under differential inclusions. *SIAM Journal on Control and Optimization* 40(3):853–881.

Bene, C.; Doyen, L.; and Gabay, D. 2001. A viability analysis for a bio-economic model. *Ecological Economics* 36:385–396.

Berleant, D., and Kuipers, B. J. 1997. Qualitative and quantitative simulation: Bridging the gap. *Artificial Intelligence* 95(2):215–255.

Bouwer, A., and Bredeweg, B. 2002. Aggregation of qualitative simulations for explanation. In Agell, N., and Ortega, J. A., eds., *Working Papers of 16th Workshop on Qualitative Reasoning*, 11–18. Download at http://www.upc.es/web/QR2002/Principal.htm.

Bradley, E.; Easley, M.; and Stolle, R. 2001. Reasoning about nonlinear system identfication. *Artificial Intelligence* 133:139–188.

Cardaliaguet, P.; Quincampoix, M.; and Saint-Pierre, P. 1994. Some algorithms for differential games with two players and one target. *Mathematical Modelling and Numerical Analysis* 28(4):441–461.

Chahma, I. A. 2003. Set-valued discrete approximation of state-constrained differential inclusions. *Bayreuther Mathematische Schriften* 67:3–162.

Chodun, W. 1989. Differential games of evasion with many pursuers. *Journal of Mathematical Analysis and Applications* 142(2):370–389.

Cury, P. M.; Mullon, C.; Garcia, S. M.; and Shannon, L. J. 2005. Viability theory for an ecosystem approach to fisheries. *ICES Journal of Marine Science* 62(3):577–584.

Dordan, O. 1992. Mathematical problems arising in qualitative simulation of a differential equation. *Artifical Intelligence* 55:61–86.

Dordan, O. 1995. *Analyse Qualitative*. Paris: Masson.

Eisenack, K., and Petschel-Held, G. 2002. Graph theoreticel analysis of qualitative models in sustainability science. In Agell, N., and Ortega, J. A., eds., *Working Papers of 16th Workshop on Qualitative Reasoning*, 53–60. Download at http://www.upc.es/web/QR2002/Principal.htm.

Eisenack, K.; Lüdeke, M.; Petschel-Held, G.; Scheffran, J.; and Kropp, J. 2007. Qualitative modeling techniques to assess patterns of global change. In *Advanced Methods for Decision Making and Risk Management in Sustainability Science*. New York: Nova Science Publishers. Forthcoming.

Eisenack, K.; Lüdeke, M.; and Kropp, J. 2006. Construction of archetypes as a formal method to analyze social-ecological systems. In *Proceedings of the IDGEC Synthesis Conference*.

Eisenack, K.; Scheffran, J.; and Kropp, J. 2006. Viability analysis of management frameworks for fisheries. *Environmental Modelling and Assessment* 11:69–79.

Eisenack, K. 2006. *Model Ensembles for Natural Resource Management*. Ph.D. Dissertation, Free University Berlin, DARWIN, http://www.diss.fu-berlin.de/2006/326/index.html.

Forrester, J. W. 1968. *Principles of Systems*. Cambridge: Wright-Allen-Press.

Gregory, J. M.; Dixon, K. W.; Stouffer, R. J.; Weaver, A. J.; Driesschaert, E.; Eby, M.; Fichefet, T.; Hasumi, H.; Hu, A.; Jungclaus, J. H.; Kamenkovich, I. V.; Levermann, A.; Montoya, M.; Murakami, W.; Nawrath, S.; Oka, A.; Sokolov, A. P.; and Thorpe, R. B. 2005. A model intercomparison of changes in the atlantic thermohaline circulation in response to increasing atmospheric CO_2 concentration. *Geophysical Research Letters* 32:L12703.

Guo, M.; Xue, X.; and Li, R. 2003. The osciallation of delay differential inclusions and fuzzy biodynamic models. *Mathematical and Computer Modelling* 37:651–658.

Hüllermeyer, E. 1997. *Reasoning about systems based on incomplete and uncertain models*. Ph.D. Dissertation, Universität-GH Paderborn.

Ivanov, G. E., and Polovinkin, E. S. 1995. On strongly convex differential games. *Differential Equations* 31(10):1603–1612.

Kay, H., and Kuipers, B. J. 1993. Numerical behavior envelopes for qualitative simulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-93)*, 606–613. AAAI/MIT Press.

Kay, H. 1998. SQsim: A simulator for imprecise ODE models. *Computers and Chemical Engineering* 23(1):27–46.

Křivan, V., and Colombo, G. 1998. A non-stochastic approach for modeling uncertainty in population dynamics. *Bulletin of Mathematical Biology* 60(4):721–751.

Kuipers, B. J. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge: MIT Press.

Kuipers, B. J. 2000. Qualitative and semi-qualitative simulation. Unpublished presentation at the Monet Summer School 2000.

Maisse, E., and Pousin, J. 1997. Diffusion and dissolution/precipitation in an open porous reactive medium. *Journal of Computational and Applied Mathematics* 82:279–290.

Marchaud, A. 1934. Sur le champs de demi-cones et les équations differielles du premier ordre. *Bulletin de la Societé Mathématique de France* 62.

Millennium Ecosystem Assessement. 2005. *Ecosystems and Human Well-Being: Synthesis*. Washington: Island Press.

Nakićenović, N.; Alcamo, J.; Davis, G.; de Vries, B.; Fenhann, J.; Gaffin, S.; Gregory, K.; Grübler, A.; Jung, T. Y.; Kram, T.; Rovere, W. K. L.; Michaelis, L.; S.Mori; Morita, T.; Pepper, W.; Pichter, H.; Price, L.; Raihi, K.; Roehrl, A.; Rogner, H.-H.; Sankovski, A.; Schelsinger, M.; Shukla, P.; Smith, S.; Swart, R.; van Rooijen, S.; Victor, N.; and Dadi, Z. 2000. *IPCC Special Report on Emissions Scenarios*. Cambridge, New York: Cambridge University Press.

Petschel-Held, G.; Block, A.; Cassel-Gintz, M.; Kropp, J.; Lüdeke, M. K. B.; Moldenhauer, O.; Reusswig, F.; and Schellnhuber, H.-J. 1999. Syndromes of Global Change: A qualitative modeling approach to support environmental management. *Environmental Modeling and Assessment* 4(4):295–314.

Petschel-Held, G. 2005. Qualitative reasoning and global change research: Experiences form modelling human-environment systems for policy advice. In *Proceedings of the 19th International Workshop on Qualitative Reasoning QR-05*.

Price, C. J.; Travé-Massuyès, L.; Milne, R.; Ironi, L.; Bredeweg, B.; Lee, M. H.; Struss, P.; Snooke, N.; Lucas, P.; and Cavazza, M. 2005. Qualitative futures. In *Proceedings of the 19th International Workshop on Qualitative Reasoning QR-05*.

Richardson, G. P. 1986. Problems with causal-loop diagrams. *System Dynamics Review* 2:158–170.

Saint-Pierre, P. 1994. Approximation of the viability kernel. *Applied Mathematics and Optimization* 24:187–209.

Saint-Pierre, P. 2004. Personal communication, January 2004.

Say, A. C. C., and Akin, H. L. 2002. Sound and complete qualitative simulation is impossible. In Agell, N., and Ortega, J. A., eds., *Working Papers of 16th Workshop on Qualitative Reasoning*, 19–27. Download at http://www.upc.es/web/QR2002/Principal.htm.

Schellnhuber, H.-J.; Lüdeke, M. K. B.; and Petschel-Held, G. 2002. The syndromes approach to scaling: Describing global change on an intermediate functional scale. *Integrated Assessment* 3(2-3):201–219.

Stainforth, D. A.; Aina, T.; Christensen, C.; Collins, M.; Faull, N.; Frame, D. J.; Kettleborough, J. A.; Knight, S.; Martin, A.; Murphy, J. M.; Piani, C.; Sexton, D.; Smith, L. A.; Spicer, R. A.; Thorpe, A. J.; and Allen, M. R. 2005. Uncertainty in predicitons of the climate response to rising levels of greenhouse gases. *Nature* 433:403–406.

Stave, K. A. 2002. Using system dynamics to improve public participation in environmental decisions. *System Dynamics Review* 18(2):139–167.

Sterman, J. D. 2000. *Business Dynamics, Systems Thinking and Modeling for a Complex World*. New York: Irwin/McGraw Hill.

Swart, R. J.; Raskin, P.; and Robinson, J. 2004. The problem of the future: sustainability science and scenario analysis. *Global Environmental Change* 14:137–146.

Travé-Massuyès, L.; Ironi, L.; and Dague, P. 2004. Mathematical foundations of qualitative reasoning. *AI Magazine* 24(4):91–106.

WCED. 1987. Our common future. Oxford: World Commission on Environment and Development.

Zaremba, S. K. 1936. Sur les équations au paratingent. *Bulletin des Sciences Mathémathiques* 60.

Knowledge Representation for Fuzzy Model Composition

Xin Fu and Qiang Shen
Department of Computer Science
University of Wales, Aberystwyth
{xxf06, qqs}@aber.ac.uk

Abstract

Compositional Modelling (CM) has been applied to synthesize automatically plausible scenarios in many problem domains with promising results. However, due to the lack of capability to deal with imprecise or ill-defined information, there is a pressing need to improve the robustness and accuracy of the existing CM work. This paper presents a more flexible knowledge representation formalism that combines fuzzy set theory and recently developed CM methods to support automating the process of generating plausible scenario spaces. The proposed knowledge representation incorporates both fuzzy parameters and fuzzy constraints into the representation of conventional model fragments. The fuzzy model composition process is illustrated by means of a simple worked example for aiding in crime investigation.

Introduction

One of the hallmark contributions of qualitative reasoning is the method for creating models automatically for a specific task given a problem domain theory. Compositional Modelling (CM) (Falkenhainer & Forbus 1991) (Keppens & Shen 2001) (which has already become standard in qualitative reasoning) has been employed to synthesize and store plausible scenario spaces effectively and efficiently in many problem domains (e.g. physical (Hamscher, Console, & de Kleer 1992; Nayak & Joskowicz 1996; Rickel & Porter 1997), ecological (Keppens & Shen 2004; Salles *et al.* 2003) and criminological (Shen *et al.* 2007)). The use of CM enables the construction of scenario descriptions automatically under widely varying circumstances without having to rely on an overly large knowledge base. This is rooted in the observation that in a scenario space the constituent parts of different scenarios are not normally unique to any one specific scenario, and that there are potentially many scenarios that possess common or similar properties locally or globally. The scenario elements and their relationships can therefore be modelled as generic and reusable fragments and they only need to be recorded once in the knowledge base.

Given a specific task, the plausible models which can solve or explain this task can be modelled in a variety of ways. Such model fragments are generally applicable to various scenario models, hence this results in a significantly increased efficiency and flexibility. For example, for applications like serious crime detection and prevention, rather than describing each scenario individually, a wide range of composing states and events, say factually and potentially available evidence, investigating actions and hypotheses can be captured in abstract form and be organized and stored in a knowledge base. Given obtained evidence (e.g. crime location and involved victims), scenario descriptions that may explain such evidence can then be synthesized dynamically by combining those potentially relevant composing states and events which are instantiated with the evidence and facts provided.

Having recognized this, CM has been applied to the building of an intelligent crime investigation decision support system (Shen *et al.* 2007) to assist human investigators by automatically constructing plausible scenarios and analyzing the likely further investigating actions with promising results. Despite the promising performance and results of the existing system, it is assumed that the model fragments and expert knowledge within the knowledge base can all be expressed by precise and crisp information. However, in reality, the degree of precision of the available evidence and intelligent data can vary greatly. In many cases, precise information is relatively more difficult to obtain than low resolution information. For instance, in cognitive modelling, different people may hold different conceptual models of the world. Indeed, under many circumstances, it is difficult to express a view with a crisp value. For example, consider the police discovered the dead body of Smith in his bedroom. Bob who is the next-door neighbour witnessed somebody going into Smith's house; however, it is difficult for Bob to state an accurate height for that person (e.g. 180 cm). Intuitively, he might just describe the height of the person as tall, short or average.

Furthermore, in the existing work, each scenario fragment employs a set of probability distributions to represent the likelihood of its associated outcomes, and these are described in numerical forms. However, such assessment of likelihood typically reflects the expertise and knowledge of experienced investigators and is normally available in lin-

guistic terms instead (Halliwell, Keppens, & Shen 2003). The use of seemingly accurate numeric probabilities suffers from an inadequate degree of precision. It would be more appropriate and desirable to incorporate a measurement of imprecision in depicting the probability distributions.

Fuzzy set theory offers a useful means of capturing and reasoning with uncertain information at varying degree of precision. Although fuzzy set theory has been applied to addressing various problems, it has not been integrated to compose a fuzzy model. This paper presents an initial attempt to extend the existing CM work to allow for representing and use of vague knowledge and linguistic probability (Cooman 2005; Halliwell & Shen 2007). It follows the existing literature in applying CM to support crime investigation by generating automatically plausible crime scenarios. This problem domain is well suited to illustrating the underlying ideas of integrating fuzzy set theory in CM, since the scenario fragments as well as the causal relations between them are highly subjective and often related to inexact and vague information.

The development of fuzzy CM mechanisms involves two conceptually distinct aspects: 1)fuzzification of parameters in the model fragments, including the identification and definition of fuzzy variables in a generic sense; and 2)fuzzy probabilistic assessment of the constraints between the states and events of the world in question.

After presenting a brief overview of the basic concepts of CM, the knowledge presentation of both fuzzy parameters and fuzzy constraints in defining fuzzfied scenario fragments is given. This is followed by an illustration of applying fuzzy model fragments to a small crime investigation problem, showing the composition process of a plausible scenario space from given evidence and facts. The final section concludes this paper and points out future work.

Basic Concepts of Compositional Modelling

In CM, the knowledge base of the model-building system consists of a number of generic scenario fragments, interchangeably termed model fragments as above, which represent generic relationships between domain objects and their states for certain types of partial scenario. In particular, a scenario fragment has two parts that encode domain knowledge: 1) the relations between domain elements which are often represented in a form that is similar to conventional production rules but in a much more general format where predicates are used to describe the properties of these domain elements; and 2) a set of probability distributions that represent how likely it is that the corresponding relationships are related.

More formally, a scenario fragment μ is a tuple $\langle \upsilon^s, \upsilon^t, \phi^s, \phi^t, A \rangle$ and is represented in the following form:

If $\{\phi^s\}$
Assuming $\{A\}$
Then $\{\phi^t\}$
Distribution ϕ^t
$\{\upsilon_1^s \ldots \upsilon_n^s \rightarrow \upsilon_1^t : q_1 \cdots \upsilon_m^t : q_m\}$

where

- υ^s is a set of variables named source-participants, referring to already identified objects of interest in the partial scenario, which can be real, artificial or conceptual objects.
- υ^t is a set of variables named target-participants, representing new objects that will be added to the partial scenario description if the model fragment is instantiated (i.e. when both the conditions and assumptions are presumed to be true).
- ϕ^s is a set of relations called structural conditions, whose free variables are elements of υ^s. Normally, the structural conditions appear in the antecedent part and describe how the source-participants are related to one another, often encoded in the form of predicates.
- ϕ^t is a set of relations called post-conditions, whose free variables are elements of $\upsilon^s \bigcup \upsilon^t$. Normally, the post-conditions appear in the consequent part and define new relations between source-participants and/or target-participants, also often encoded in the form of predicates.
- A is a set of assumptions, referring to those pieces of information which are unknown or cannot be inferred from other scenario fragments, but they may be presumed to be true for the sake of performing hypothetical reasoning.

The **If** statement describes the required conditions for a partial scenario to become applicable. These conditions must be factually true or logical consequences of other instantiated fragments.

The **Assuming** statement indicates the reasoning environment. With the purpose of performing hypothetical reasoning, this environment specifies the uncertain events and states which are presumed in a partial scenario description.

The **Then** statement describes the consequent when the conditions and presumed assumptions hold. They may represent a piece of new knowledge or relations which are derived from the hypothetical reasoning.

The **Distribution** statement indicates the probability distributions of the consequent variables or those of their relations. The left hand side of the "implication" sign in each instance of such a statement is a combination of variable-value pairs, involving antecedent and assumption variables, and the right hand side indicates the likelihood of each alternative outcome if the fragment is instantiated.

For example, the following scenario fragment shows a piece of generic forensic knowledge that, assuming that suspect S overpowers victim V, there is a 75% chance that fibres will be transferred from S to V:

If {suspect(S), victim(V)}
Assuming {overpowers(S,V)}
Then {transfer(fibers,S,V)}
Distribution transfer(fibers,S,V){
true, true, true →
true: 75%, false: 25% }

Given a collection of such local model fragments and some observations (evidence), CM applies an inference procedure to create a space of scenario descriptions at a global level. As the details of this procedure are very similar to

what is to be employed in fuzzy CM to be reported later, they are omitted here. Interested readers can refer to (Shen *et al.* 2007) for further details.

Foundations of Fuzzy CM

This section focuses on the creation of a structured knowledge representation scheme which is capable of storing and managing vague or ill-defined data including facts, evidence and assumed information. Effort has been made to encode fuzzy scenario fragments in a pre-specified format. The research developed here is loosely based on knowledge representation given in (Shen *et al.* 2007) and its related work; however, it is adapted to represent imprecise and uncertain information, including both parameters and constraints.

Fuzzy parameters

For many problems, there may be many variables that share similar properties while most of these properties only involve minor variations from one another if encoded computationally, in terms of knowledge representation. This is independent of whether the variables are fuzzy or not. For example, variables such as quantity, volume and proportion all reflect the concept of capacity. This group of variables may all be expressed by linguistic terms such as large, average or small (which can be conveniently represented by fuzzy sets). Therefore, when defining a fuzzy variable, rather than redefining a new quantity space for it completely from scratch each time, it has a natural appeal to group fuzzy variables which share something in common into the same class. In each class, the common features shared by the variables are extracted and represented by an abstract variable with its quantity space specified over a normalized universe of discourse. The quantity space of a variable belonging to a given class is created by inheriting the common features from the abstract variable and by embellishing it with new or modified properties.

To enable this development, fuzzy taxonomies that describe vague states and events for use in the scenario fragments are introduced here. A taxonomy is considered to be a hierarchy, where those variables at a lower level are more specific than their ancestors and represent a more specialized group of fuzzy variables. In so doing, fuzzy variables in a CM knowledge base are organized in a structured manner. This does not only improve the efficiency of storing knowledge via reusing abstract fuzzy variables, but also helps reveal both the commonality and speciality of different variables. More importantly, the use of fuzzy taxonomies supports the construction of scenario spaces in a systematic and concise manner due to the inheritance property of the hierarchies.

Consider, for instance, the taxonomies shown in Fig. 1. The first organises a set of fuzzy variables relating to an abstract fuzzy variable named *Measurement*. Hence, fuzzy variables height, distance, width, depth and length share certain properties in defining their quantity spaces as they inherit such common features from the abstract *Measurement* variable; all of them can be measured with respect to a certain measurable unit and can be described as long, average or short. Similarly, the variables in the second taxonomy are all used to describe levels of different concepts. Although they may denote rather distinct or even seemingly irrelevant properties (e.g. temperature and difficulty), they all take on values from the same underlying abstract quantity space in terms of various levels such as high, average or low.

Note that, in these taxonomies, even the fuzzy variables which are classified into different classes may still have some more generic and deep underlying commonalities. For instance, temperature in the second taxonomy is also a measurable variable. Hence, from a more generic aspect, they may still be allocated to a superclass which is more abstract. However, in order to maintain the clarity of representation and the comprehensibility of inference drawn from such representations, fuzzy taxonomies are not built in the most generic way possible, but are classified with easy interpretability in mind.

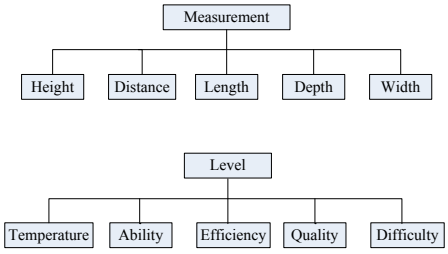


Figure 1: Example taxonomies of fuzzy variables

From above, it is clear that in defining scenario fragments fuzzy variables can be divided into two types: abstract or non-abstract. Abstract fuzzy variables are actually variable classes that cannot be instantiated themselves in an effort to describe any actual scenario and non-abstract fuzzy variables are those that can be instantiated. Clearly, in Fig. 1 *Measurement* and *Level* are abstract fuzzy variables, and depth, distance, height, efficiency, etc. are non-abstract variables.

In implementation, abstract fuzzy variables are indicated by means of the keyword **abstract**. Defining such a variable involves specifying the following fields:

- **Name:** A constant that uniquely identifies the abstract fuzzy variable.
- **Universe of discourse:** The domain of the abstract variable. The default definition is $[0, 1]$. Any descendant of an abstract fuzzy variable can modify the universe of discourse according to their physical dimension.
- **Cardinality of partition:** The number of fuzzy sets which jointly partition the universe of discourse. This is represented by a symbol n which will be substituted by a positive integer in a lower level non-abstract variable.
- **Quantity Space:** A set of ordinal relationships that describe the value of a continuous parameter. Here, these relationships are represented by the membership functions of each fuzzy set that jointly cover the partitioned domain.

For example, the aforementioned abstract fuzzy variable *Level* can be defined as follows (adhering to the conventional representation style of model fragments):

Define ***abstract*** fuzzyvariable {
Name: Level
Universe of discourse: $[0, 1]$
Cardinality of partition: n
Quantity space:

$$
\begin{aligned}
fs_1 &= \left[0, 0, \tfrac{1}{n-1}\right] \\
&\cdots \\
fs_i &= \left[\tfrac{i-2}{n-1}, \tfrac{i-1}{n-1}, \tfrac{i}{n-1}\right] \\
&\cdots \\
fs_n &= \left[\tfrac{n-2}{n-1}, 1, 1\right]
\end{aligned}
$$

}

It would be inefficient and practically unnecessary to store and manipulate fuzzy sets with arbitrarily complex membership functions. Only the triangular membership functions are considered in this initial work. Thus, a quantity space specification consists of an ordered list of triples comprising the start, top and end points of each membership function. For both computational and presentational simplicity, triangular membership functions in which the edge of a fuzzy set's membership function is exactly intersected to the centroid of the neighboring one are used in this paper. For example, assume $n = 5$, then the defined quantity space of the abstract fuzzy variable *Level* is shown in Fig. 2.

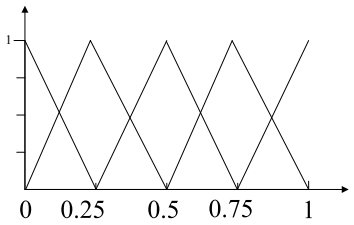


Figure 2: A quantity space

Non-abstract fuzzy variables are identified by means of the absence of the keyword **abstract**. Such definition involves "is-a" relationships in which a non-abstract fuzzy variable is said to inherit from an abstract fuzzy variable. It requires addition of fields that are specific to the variable under definition, with shared commonalities already defined in the corresponding superior abstract fuzzy variable. In fuzzy CM, such new fields are defined as follows:

- **Is-a:** The name of an abstract fuzzy variable which refers to the immediate parent of the current fuzzy variable in a given taxonomy.
- **Scalar:** A constant which is used to scale up or down the normalized universe of discourse of the corresponding abstract variable.
- **Unit:** The variable's physical dimension. If a fuzzy variable has no unit, a default value of none is set for this field.
- **Name of fuzzy sets:** The name of each fuzzy set in the defined quantity space.
- **Unifiability:** The declaration of a unifiable property of the variable, specified by a predicate.

The following example defines a non-abstract fuzzy variable named *Chance* that inherits from *Level*.

Define fuzzyvariable {
Name: Chance
Is-a: Level
Cardinality of partition: 5
Scalar: 1
Unit: none
Name of fuzzy sets: {extremely_unlikely, slim_chance, likely, very_likely, good_chance}
Unifiability: Chance(X)
}

Obviously, this non-abstract fuzzy variable *Chance* is a kind of *Level*. Due to property inheritance, its universe of discourse equals to the normalized universe of discourse multiplied by the scalar over the corresponding physical dimension. Its quantity space is evenly partitioned by 5 fuzzy sets which are described respectively by the five linguistic terms given. Also due to inheritance, the membership functions of those fuzzy sets are obtained once again by multiplying the corresponding key points in each fuzzy set by the scalar.

Fuzzy constraints

In CM, knowledge is normally expressed as constraints or relations which must be obeyed by certain variables involved in a given problem domain. For example, velocity and duration relations often appear in physical reasoning systems; population growth and competition relations often appear in ecological reasoning system; length and angle relations often appear in spatial reasoning systems. Such constraints as used in the existing work require numerical values to quantify the probability of a consequence's occurrence, as previously illustrated.

Since such subjective probability assessments are often the product of barely articulate intuitions, the seemingly numerically precise expressions may cause loss of efficiency, accuracy and transparency (Cooman 2005; Halliwell, Keppens, & Shen 2003; Halliwell & Shen 2007). Under many circumstances, an expert may be unwilling or simply unable to suggest a numerical probability. For example, consider the following scenario: a dead body of Smith was discovered at home and the cause of the death was suspected to be suicide. A psychologist was then invited to examine the mental condition of Smith by analysing his diary. Consultation with the psychologist is unlikely to yield much beyond vague statements like "According to his diary, he is extremely unlikely to kill himself" or "According to his diary, he stood a good chance of killing himself". Therefore, the initial work developed here models the vagueness of the probability distribution in terms of subjective linguistic probabilities. Rather than using numerical representation as in the literature, a fuzzy variable called *Chance* which inherits the properties of the abstract fuzzy variable *Level* is introduced to capture subjective probabilistic assessments. Both the *Chance* variable and its superior abstract variable *Level* have been presented in previous section.

Similar to the existing approach, a scenario fragment includes a set of probability distributions over the possible assignments of the consequent ϕ^t, for those interested combinations of assignments to the variables within the structural conditions and assumptions. Note that, it is not required to define each combination, the probability distribution only focuses on those of interest. This can be generally represented by:

$$P(a_1 : v_1, \ldots, a_m : v_m \rightarrow c : v_{cp}) = fs_p \qquad (1)$$

where $a_i : v_i$, $i\epsilon\{1, 2, \cdots, m\}$ denotes the assignment obtained by assigning v_i to variable a_i, $c : v_c$ has a similar interpretation, and fs_p is a member of the quantity space that specifies the fuzzy variable *Chance*.

As an example, the following fragment illustrates the concepts and applicability of fuzzy constraints:

If {height(S), height(V)}
Assuming {attempted_to_kill(S,V)}
Then {difficult_level(overpower(S,V))}
Distribution difficult_level(overpower(S,V)) {
tall, short, true →
easy: good_chance, difficult: slim_chance }

It describes a causal relation holding among structural condition a_1 and a_2, assumption a_3 and post-condition c, here

$a_1 = height(S)$ indicates the height of a suspect S, which is a fuzzy variable that takes on values from a predefined quantity space of {very_short, short, average, tall, very_tall}.
$a_2 = height(V)$ indicates the height of a victim V, whose possible value assignment is the same as S.
$a_3 = attempted\_to\_kill(S, V)$ describes that suspect S attempted to kill victim V, representing a conventional boolean predicate.
$c = difficulty\_level(overpower(S, V))$ describes the difficulty level for suspect S to overpower victim V, with possible assignments being easy, average and difficult.

Note that, when defining probability distributions in scenario fragments, the names of those variables within the structural conditions, assumptions and post-conditions (e.g. a_1, a_2, a_3 and c) are omitted when such omissions do not affect the interpretation of the meaning of the associated values, for the sake of presentational simplicity. Thus, the probability distributions can be rewritten as follows:

$$v_1, v_2, \cdots, v_m \rightarrow v_{c1} : fs_1, \cdots, v_{cp} : fs_p$$

The above fragment reveals a general relation between the heights of two people involved in a fight and the difficulty level for one to overpower the other, and it can be applied to modelling various scenarios. For example, this fragment covers a fuzzy production rule which indicates that if suspect S is tall, while victim V is short, and the suspect indeed attempted to kill the victim, then the suspect stands a good chance of overpowering the victim easily. Conversely, if the suspect is shorter than the victim and he indeed attempted to kill the victim, then there is only a slim chance for the suspect to overpower the victim easily.

Application to Crime Investigation: Outline of Scenario Composition

The proposed knowledge representation formalism and how it is used to support CM is illustrated here with a sample application to the generation of plausible scenarios reflecting a crime situation in which a number of fibers matching Bob's clothes (Bob is the suspect) have been found on the dead body of Dave. Relevant evidence and the key scenario fragments of the sample knowledge base are presented in Appendix A. From the given facts, collected evidence and this knowledge base, a structural scenario space can be generated by joint use of two conventional inference techniques named *abduction* and *deduction*. Note that since the degree of precision of the information (including both predefined knowledge and available evidence/facts) can vary greatly, the collected evidence and the knowledge base cannot in general be matched precisely. Thus, a fuzzy matching method is applied for scenario fragment instantiation.

Initialization

To generate a space of plausible scenarios, collected evidence and any available facts are firstly entered. The present example shows one piece of evidence in which a number of fibers collected from Dave's body have been identified matching the fibers of Bob's clothes, and two available facts in which Dave is known to be the victim and Bob is under suspicion. The result of this initialization phase is shown in Fig. 3.

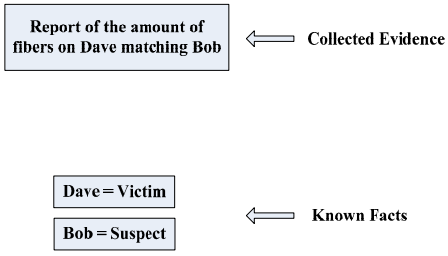


Figure 3: Result of initialization

Backward chaining phase

This phase involves the abduction of all domain objects and their states that might cause the available evidence. These plausible causes are created by instantiating the conditions and assumptions of the scenario fragments in the knowledge base, whose consequences match the collected evidence in the emerging scenario space. After that, the newly created instances of all plausible causes are recursively used in the same manner as the original piece of evidence, instantiating all relevant fragments and adding new nodes that correspond to the instantiated conditions and assumptions to the emerging scenario space. For the present example, this phase leads to what is shown in Fig. 4.

A brief explanation of how such *abduction* phase works with respect to the following sample fragment and collected evidence/facts is given below:

If {degree_of_fight(S,V)}
Assuming {transfer(X,S,V),find_match(X,V,S)}

Then { `evidence(amount(transferred(`X`,`V`,`S`)))`}
Distribution `evidence(amount(transferred(`X`,`V`,`S`)))`
`{intensive,true,true→many:good_chance,few:slim_chance`
`weak,true,true→many:slim_chance,few:good_chance}`

Given the collected evidence that a number of fibers matching Bob's clothes have been found on the dead body of Dave, which matches the consequent variable of the above scenario fragment, the variables within the structural conditions and assumptions X, S and V are firstly instantiated with fibers, Bob and Dave, respectively. The resulting instantiated nodes (e.g. Transfer fibers from Bob to Dave, Degree of fight between Bob and Dave and Find fibers on Dave matching Bob) are then added to the emerging scenario space.

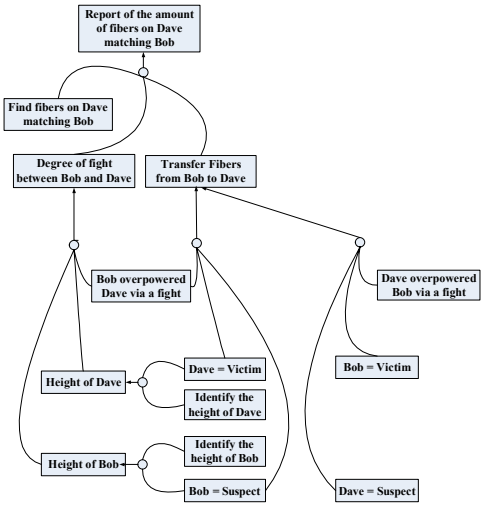


Figure 4: Result of backward chaining

Fuzzy matching To allow instantiation of a fuzzy scenario fragment when given a piece of evidence, the extended compositional modeller requires matching specific data items with broader and relatively subjective information in the knowledge base. As aforementioned, the evidence and the knowledge base cannot always be matched precisely. Under many circumstances, however, the values of the involved fuzzy variables do not have to be identical, partial matching suffices. Such matching is done by the following process.

First, find those scenario fragments that involve the same variables as the underlying fuzzy variables that describe the collected evidence. For example, in the backward chaining phase, the consequence and collected evidence in the above example both contain the amount of the transferred substance X (with the amount being a fuzzy variable). Second, identify the degree of the match between the evidence and the found scenario fragments. Third, return a matched scenario fragment for instantiation if the match degree is larger than a predefined threshold, otherwise, no match between them is found.

Fig. 5 illustrates how such a fuzzy match mechanism actually works. Given the collected evidence that a number of transferred fibers exist, a match degree of 0.8 is obtained by calculating the maximum membership value over the overlapping area between "a number of" and "many" fuzzy sets. Note that more complex calculi for matching degree may be developed; however, for computational simplicity and thanks to the employment of triangular fuzzy sets only, this straightforward matching method is adopted here. Clearly, much remains to be done in order to have a more general approach regarding the set-up of the important threshold used in the third step. Yet, this does not affect the understanding of the underlying inference techniques introduced herein.

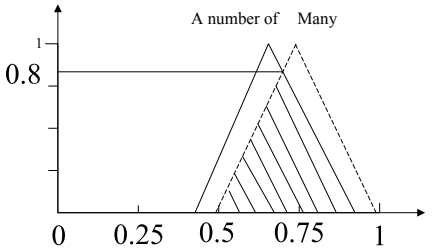


Figure 5: The fuzzy matching mechanism

Forward chaining phase

While all plausible causes of the collected evidence and some pieces of additional evidence may be introduced to the emerging scenario space during the backward chaining phase, the forward chaining phase is responsible for extending the scenario space by adding all plausible consequences of the fragments whose conditions and assumptions match the instances created in the last phase. This produces potential pieces of evidence that have not yet been identified but may be used to improve the plausible scenario description.

This procedure applies logical *deduction* to all the scenario fragments in the knowledge base, whose conditions and assumptions match the existing nodes in the emerging scenario space. The actual matching method used is basically the same as that used previously (except step 1 obviously). For the running example, based on those newly introduced nodes such as "Bob = victim", "Dave = suspect" and "Dave overpowered Bob via a fight", their deduced corresponding consequences are then created and added to the emerging scenario space. Fig. 6 depicts the resulting scenario space that may be the outcome of this phase (depending on the actual knowledge base used).

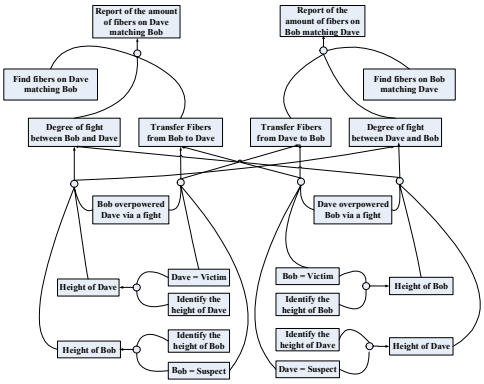


Figure 6: Result of forward chaining

Removal of spurious nodes

In the backward chaining phase, some spurious nodes may have been added to the emerging scenario space. Such nodes

are root nodes in the space graph which are neither facts or instantiated assumptions nor the justifying nodes that support the instantiated assumptions. This step aims to remove the spurious nodes and their immediate consequences. In this example, the emerging scenario space containing the following information that Dave is both the suspect and victim at the same time, and the same for Bob. Since Dave is known to be the victim whereas Bob is known as the suspect, the nodes "Dave = suspect" and "Bob = Victim" as well as their directly supported nodes can be removed from this emerging scenario space. The remaining scenario space is shown in Fig. 7.

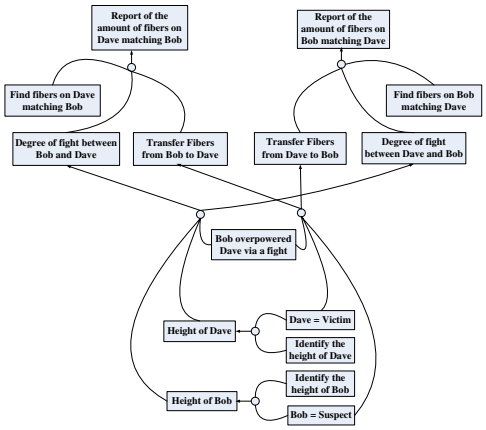


Figure 7: Result of spurious node removal

Use of generated scenario space

Once the plausible scenario space is generated, it provides effective assistance for crime investigators by allowing them to seek potential answers to a range of possible queries. For instance, an investigator may query the system for scenarios by inputing his/her interested evidence or hypotheses. Also, the investigator might discover that a tall person was observed entering the crime scene on a CCTV camera, and wonders whether this would rule out homicidal death. The system can answer this type of question by adding this new evidence to the set of collected pieces of evidence and modifying the generated scenario description to establish whether the new evidence indeed supports the hypothesis. Note that compared with previous work, the present approach provides more flexible query support, as it has the capability to deal with fuzzy queries.

Conclusions

This paper has enriched and adapted the knowledge representation formalism in existing CM work, to enable it to represent, store and support reasoning about vague and imprecise data, by the use of fuzzy sets. The new knowledge representation formalism concerns both fuzzy parameters and fuzzy constraints by incorporating them into the representation of conventional model fragments. The applicability of the proposed method is illustrated by means of a simple worked example for aiding inexperienced crime investigators in speculating about all plausible causes of the collected evidence.

Note that, attempts to model probabilistic terms using fuzzy sets have proven more successful. For example, a relatively sophisticated experimental method for eliciting fuzzy models of probabilistic terms has been developed in (Wallsten *et al.* 1986) and the inter-subjective stability of generated terms has been examined with promising results. In addition, it has been reported in (Zimmer 1990) that verbal expressions of probabilistic uncertainty can be "more accurate" than numerical values in estimating the frequency of multiple attributes by experimental studies. Whilst there are outstanding problems such as context sensitivity with the fuzzy approach to modelling probabilistic terms, these psychometric studies are unanimous in preferring fuzzy descriptions of probability to numerical estimates.

While the proposed method presented here shows powerful potential functionalities and significant benefits in supporting qualitative reasoning, there are still many open problems and areas that require further research. In particular, the proposed method is not yet able to analyze the generated scenarios space and therefore to provide evidence collection strategies for decision support. In order to improve the effectiveness of evidence collection, the generated plausible scenarios need to be evaluated by means of calculating the most likely scenario.

Also, the fuzzy constraints within a single scenario fragment are defined by employing a fuzzy variable named *Chance*. However, when dynamically composing these potential relevant scenario fragments into plausible scenario descriptions, the fuzzy constraints will be propagated from individual fragments to their related ones. How to combine and propagate fuzzy probabilities, in conjunction with the backward and forward propagation of the fuzzy matching degrees, in an emerging model space is a tough problem that needs to be taken into account in further research. Original work as represented in (Halliwell, Keppens, & Shen 2003; Halliwell & Shen 2007) may serve as a starting point for this.

While solving complex problems, the size of the knowledge base and the number of attributes involved might become very large, the abduction and deduction inference mechanism is quite expensive to generate the scenario spaces and is only practical for simple knowledge bases. In order to enhance the effectiveness and efficiency of the generation of scenario spaces by selecting the most relevant attributes, another important piece of future work concerns the use of dynamic constraint satisfaction problem (DCSP) (Mittal & Falkenhainer 1990) techniques where activity constraints are employed to dynamically determine which attributes should be activated in the problem, thus the problem of dimensionality may be greatly reduced.

Acknowledgments

This work was supported in part by UK EPRSC grant EP/D057086. The first author was also supported by a UK ORS award. The authors are grateful to Mark Lee, Jeroen Gunning and Ruiqing Zhao for their helpful discussions, but will take full responsibility for the views expressed in this paper.

Appendices

Key Sample Data and Scenario Fragments

```
Define action{
name = find_match
description = find the substance X on V matching S
unifiability = find_match(X,V,S)}

Define action{
name = identify_height
description = identify the height of P
unifiability = identify(height(P))}

Define evidence{
name = report_of_amount
description = report of the amount of X
unifiability = evidence(amount(X))}

Define fuzzyvariable {
name = height
is-a = measurement
cardinality of partition = 5
scalar = 250
unit = centimeter
names of fuzzy sets = {very_short, short, average, tall,
very_tall}
unifiability = height(P)}

Define fuzzyvariable {
name = amount
is-a = capacity
cardinality of partition = 5
scalar = 1
unit = none
names of fuzzy sets = {none, few, several, a_number_of, many }
unifiability = amount(X)}

If {suspect(S),victim(V)}
Assuming {overpower(S,V)}
Then { transfer(X,S,V)}
Distribution transfer(X,S,V){
true,true,true→true:good_chance, false:slim_chance}

If {suspect(S),victim(V)}
Assuming {overpower(S,V)}
Then { transfer(X,V,S)}
Distribution transfer(X,V,S){
true,true,true→true:good_chance, false:slim_chance}

If {person(P)}
Assuming {Identify(height(P))}
Then { height(P)}
Distribution height(P){
true,true→true:1, false:0}

If {degree_of_fight(S,V)}
Assuming {transfer(X,S,V),find_match(X,V,S)}
Then { evidence(amount(transferred(X,V,S)))}
Distribution evidence(amount(transferred(X,V,S)))
{intensive,true,true→many:good_chance,few:slim_chance
weak,true,true→many:slim_chance,few:good_chance}

If {height(V), height(S)}
Assuming {overpower(S,V)}
Then {degree_of_fight(S,V)}
Distribution degree_of_fight(S,V)
{tall,short,true→intensive:slim_chance,weak:good_chance
short,tall,true→intensive:slim_chance,weak:good_chance
tall,tall,true→intensive:good_chance,weak:slim_chance
short,short,true→intensive:good_chance,weak:slim_chance}

If {height(V), height(S)}
Assuming {overpower(S,V)}
Then {degree_of_fight(V,S)}
Distribution degree_of_fight(V,S)
{tall,short,true→intensive:slim_chance,weak:good_chance
short,tall,true→intensive:slim_chance,weak:good_chance
tall,tall,true→intensive:good_chance,weak:slim_chance
short,short,true→intensive:good_chance,weak:slim_chance}

Translation {unifiability = overpower(S,V)
description: S overpowers V}

Translation {unifiability = degree_of_fight(S,V)
description: the degree of fight between S and V}

Translation {unifiability = transfer(X,S,V)
description: X were transferred from S to V}

Translation {unifiability = find_match(X,V,S)
description: find the substance X on V matching S}

Translation {unifiability = amount(X)
description: the amount of X}

Translation {unifiability = identify(height(P))
description: identify the height of person P}

Translation {unifiability = evidence(amount(transferred(X,V,S)))
description: report of the amount of transferred X found on V
matching S}
```

References

Cooman, G. d. 2005. A behavioural model for vague probability assessments. *Fuzzy sets and Systems* 154(3):305–358.

Falkenhainer, B., and Forbus, K. 1991. Compositional modelling: Finding the right model for the job. *Artificial Intelligence* 51:95–143.

Halliwell, J., and Shen, Q. 2007. Linguistic probabilities: Theory and application. *To appear in Soft Computing*.

Halliwell, J.; Keppens, J.; and Shen, Q. 2003. Lingustic bayesian networks for reasoning with subjective probabilities in forensic statistics. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, 42–50.

Hamscher, W.; Console, L.; and de Kleer, J. 1992. *Readings in Model-Based Diagnosis*. San Francisco, CA, USA: Morgan-Kaufmann.

Keppens, J., and Shen, Q. 2001. On compositional modelling. *Knowledge Engineering Reivew* 16(2):157–200.

Keppens, J., and Shen, Q. 2004. Compositional model repositories via dynamic constraint satisfaction with order-of-magnitude preferences. *Journal of Aritificial Intelligence Research* 21:499–550.

Mittal, S., and Falkenhainer, B. 1990. Dynamic constraint satisfaction problems. In *Processdings of the 8th National Conference on Aritifical Intelligence*, 25–32.

Nayak, P., and Joskowicz, L. 1996. Efficient compositional modeling for generating causal explanations. *Artificial Intelligence* 83:193–227.

Rickel, J., and Porter, B. 1997. Automated modeling of complex systems to answer prediction questions. *Artificial Intelligence* 93:201–260.

Salles, P.; Bredeweg, B.; Araujo, S.; and Neto, W. 2003. Qualitative models of interactions between two populations. *AI Communications* 16(4):291–308.

Shen, Q.; Keppens, J.; Aitek, C.; Schafer, B.; and Lee, M. 2007. A scenario driven decision support system for serious crime investigation. *To appear in Law, Probability and Risk*.

Wallsten, T. S.; Budescu, D. V.; Rapoport, A.; Zwick, R.; and Forsyth, B. 1986. Measuring the vague meanings of probability terms. *Journal of Experimental Psychology: General* 115(4):348–365.

Zimmer, A. C. 1990. What uncertainty judgements can tell about the underlying subjective probabilities. In Kanal, L., and Lemmer, J., eds., *Uncertainty in Artificial Intelligence*. New York, NY, USA: Elsevier Science Inc.

On Qualitative Probabilities for Legal Reasoning about Evidence

Jeroen Keppens
Department of Computer Science
King's College London
Strand, London WC2R 2LS, UK
jeroen.keppens@kcl.ac.uk

Abstract

A crucial aspect of evidential reasoning in crime investigation involves comparing the support that evidence provides for alternative hypotheses. Recent work in forensic statistics has shown how Bayesian Networks (BNs) can be employed for this purpose. However, the specification of BNs requires conditional probability tables describing the uncertain processes under evaluation. When these processes are poorly understood, it is necessary to rely on subjective probabilities provided by experts, which are difficult to describe in a manner that is both accurate and precise. Recent work in qualitative reasoning has developed methods to perform this type of reasoning using coarser representations. However, the latter types of approaches are too imprecise to compare the likelihood of alternative hypotheses. This paper examines the shortcomings of the qualitative approaches when applied to the aforementioned problem, and identifies and integrates techniques to refine them.

Introduction

While legal reasoning about evidence is mostly similar to abductive diagnostic problem solving (Keppens & Schafer 2006), there are some crucial differences. The most important of these is that there is no need for decision making systems on legal evidential reasoning problems. Decisions of this nature must be made by humans, be they judges or jury panels. As such, decision support support systems for legal evidential reasoning must be primarily concerned with explaining the extent to which pieces of evidence support alternative plausible scenarios (Keppens, Shen, & Lee 2005).

So-called Bayesian techniques for testing hypotheses have proven to be a particularly influential approach to this problem. On the one hand, and unlike conventional symbolic inference mechanisms, the Bayesian approach to evidence evaluation enables the use of the well-understood classical probabilities in order to obtain a precise assessment of the relative strength of support of a piece of evidence for certain hypotheses. On the other hand, and unlike conventional techniques for statistical hypothesis testing, the Bayesian approach supports causal reasoning on how a piece of evidence can materialise as a consequence of a hypothetical scenario. This makes it particularly suitable for modelling situations that occur relatively infrequent and are difficult to synthesise. Moreover, it provides a foundation from which explanations for the results of a probabilistic analysis can be derived.

However, the Bayesian approach is not without its drawbacks. In legal evidential reasoning, it normally necessitates the use of subjective probabilities (as does any probabilistic reasoning approach). These are numbers that express a person's believe in the proposition of interest. Such probabilities are more prone to inaccuracy than ones that express the proportions of outcomes of a frequently repeated experiment in which the proposition of interest is true. Moreover, the acquisition of a set of precise expert beliefs in propositions that is consistent with the axioms of classical probability theory is a substantial problem in its own right. A rigourous evaluation of the impact of these potential inaccuracies and inconsistencies, by such techniques as sensitivity analysis, may help overcome the problem associated with these drawbacks. But this may make the technique inaccessible for many people responsible for evidence evaluation, such as crime investigators, juries and judges.

A substantial part of the difficulty of applying the Bayesian approach is due to the amount of knowledge required to acquire precise and accurate numerical probabilities. But in this domain, precise numerical probabilities are not required. Indeed, in the evaluation of forensic evidence, the objective is normally merely to produce a justifiable indication of the difference in magnitude of support for one hypothesis over another, given the available evidence. Therefore, various approaches for qualitative Bayesian inference have been developed, such as qualitative probabilistic networks (QPNs) (Wellman 1990), qualitative certainty networks (QCNs) (Parsons & Mamdani 1993) and linguistic Bayesian networks (Halliwell & Shen 2002).

Generally speaking, abductive diagnostic systems require an approach to compose complete models from partial ones. While earlier work has shown that models in the form of numerical Bayesian networks can be composed from descriptions of influences between variables, strong and somewhat unrealistic independence assumptions have to be made in order to enable the composition of influences (Keppens, Shen, & Lee 2005). However, the qualitative approaches to Bayesian inference, such as QPNs and QCNs, provide a means to reason explicitly about independent influences and, with certain extensions such as those developed by Renooij et. al. (Renooij, van der Gaag, & Parsons 2002), influ-

ences upon influences. As such, these approaches be easily adapted to perform abductive reasoning.

The main limitation of the qualitative approaches is that they are too vague to provide any useful information regarding the relative support of evidence for hypothetical scenarios, in all but the most obvious cases (Biedermann & Taroni 2006). This is due to qualitative overabstraction. Parsons (Parsons 1995) has suggested the incorporation of order of magnitude reasoning (Raiman 1991) in QPNs and QCNs, which refines the precision of these qualitative probabilistic reasoning approaches while maintaining their composability.

This paper aims to identify how qualitative probabilistic reasoning techniques can be employed to perform legal reasoning about evidence. It will examine the nature of typical relationships between variables in this domain and the limitations of the basic QPN/QCN methods in representing them. Various extensions that may address some of these issues, and which have been developed independently in the literature, are identified. By means of simple but realistic examples, the paper shall demonstrate how these individual extensions can be integrated with one another in order to produce a rich qualitative approach to Bayesian inference, which is sufficiently precise to help human decision makers assess the relative support of evidence for alternative hypotheses while retaining composability.

Bayesian Evidential Reasoning

Underlying Bayesian methods for evidence evaluation lies the notion that statistical hypothesis testing constitutes a suitable paradigm for this purpose. Cook et. al. propose that a piece of evidence e be evaluated by determining its likelihood under alternative hypotheses h_1 and h_2 (Cook *et al.* 1998). If the probability of e's occurrence assuming hypothesis h_1, denoted $Pr(e|h_1)$, is substantially higher than the probability of e's occurrence assuming hypothesis h_2, denoted $Pr(e|h_2)$, then the investigator can conclude that e provides stronger support for h_1 than for h_2.

Bayesian Networks

A Bayesian network (BN) is a representation that facilitates the computation of joint probability distributions over a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$. Reasoning with joint probability distributions over a large set of variables $\{X_1, \ldots, X_n\}$ is problematic because the number of variable assignment combinations that need to be considered increases exponentially with n. A BN simplifies these calculations by considering the independencies between variables.

A BN consists of a directed acyclic graph (DAG) that describes the independencies between variables, and a set of probability distribution tables that quantify the relations between variables. Figure 1 is an example of such a BN, which is partially based on work by Aitken et. al. (Aitken, Taroni, & Garbolino 2003). The DAG contains a node for each variable. And intuitively, each arc in the DAG from a variable A to a variable B represents the notion that A influences B.

Independencies are defined formally in a BN by means of the concept of *d-separation*. More specifically, a chain

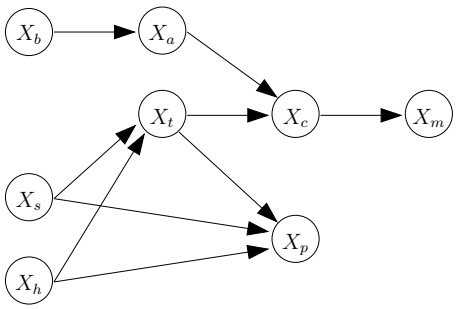


Figure 1: Sample Bayesian network

Table 1: The variables of the sample BN

Symbol	Meaning
X_b	background of suspect involves handling blood
X_a	suspect may have background blood splatters
X_s	the suspect stabbed the victim
X_h	the suspect examined the victim's body
X_t	blood was transferred from victim to suspect
X_c	investigator chooses blood splatter from victim on suspect for investigation
X_m	a blood splatter matching the victim's dna was found on the suspect
X_p	the blood splatter is a projected stain

of variables from A to B, formed by following arcs in the DAG in either direction, is said to be *blocked* by a set $\mathbf{C}$ of observed variables, if it contains

- a variable $D \notin \mathbf{C}$ with two incoming arcs in the chain (e.g. given no observations the chain $X_t \rightarrow X_c \leftarrow X_a$ is said to be blocked), or
- a variable D that is either observed (i.e. $D \in \mathbf{C}$) or that has an observed descendant (e.g. if X_p is observed, the chain $X_s \rightarrow X_t \rightarrow X_c$ is blocked).

Two variables A and B are said to be d-separated by a set $\mathbf{C}$ of observed variables if all chains between A and B are blocked by $\mathbf{C}$. Variables that are d-separated in a BN by an observation, are deemed conditionally independent given that observation.

The latter feature of BN models helps simplify the calculations of joint probability distributions. This is perhaps best explained by means of the example in Figure 1. Let each variable X_i in this BN have a boolean domain $\{x_i, \overline{x_i}\}$, where x_i denotes that X_i =true and $\overline{x_i}$ that X_i =false. Then, the joint probability that all variables are true can be computed by:

$$\begin{aligned} &P(x_b,x_a,x_s,x_h,x_t,x_c,x_m,x_p) \\ &\quad = P(x_m|x_c) \times P(x_p|x_t x_s x_h) \times P(x_c|x_t x_a) \times \\ &\qquad P(x_a|x_b) \times P(x_b) \times P(x_t|x_s x_h) \times P(x_s) \times P(x_h) \end{aligned}$$

Generally speaking, $2^8 - 1$ or 255 distinct probabilities are required to specify the joint probability distribution of the 8 variables in the BN of figure 1. With the BN, that number

Table 2: Conclusions drawn from likehood ratios by the Forensic Science Service

Likelihood Ratio	Support of evidence to prosecution claim over defense claim
1 to 10	limited
10 to 100	moderate
100 to 1,000	moderately strong
1,000 to 10,000	strong
over 10,000	very strong

can be reduced to $2^1 + 2^3 + 2^2 + 2^1 + 1 + 2^2 + 1 + 1$ or 23 distinct probabilities. These probabilities are specified in so-called conditional probability tables and prior probability distributions.

For each non-root node X in its DAG, a BN contains a conditional probability table that, in turn, contains a conditional probability distribution for each set of assignments of parent nodes of X. For each root node in its DAG, a BN contains a prior probability distribution. For example, the nodes X_t and X_c could have the following probability distributions:

$P(x_t|x_s x_h) = 0.99 \quad P(x_c|x_t x_a) = 0.3$
$P(x_t|x_s \overline{x_h}) = 0.99 \quad P(x_c|x_t \overline{x_a}) = 1$
$P(x_t|\overline{x_s} x_h) = 0.7 \quad P(x_c|\overline{x_t} x_a) = 0$
$P(x_t|\overline{x_s}\,\overline{x_h}) = 0.01 \quad P(x_c|\overline{x_t}\,\overline{x_a}) = 0$

The Likelihood Ratio Approach to Evidence Evaluation

The likelihood ratio approach evaluates a piece of evidence e with respect to two or more hypotheses. Let h_p be the hypothesis corresponding to the claim made by the prosecution and h_d be the one made by the defence in a trial. Then, the likelihood ratio (LR):

$$LR = \frac{Pr(e|h_p)}{Pr(e|h_d)}$$

expresses the degree to which the evidence is more plausible under the prosecution's claim than under the defence's claim. For instance, the Forensic Science Service, a major forensic laboratory in England and Wales, employs the likelihood ratio to report their findings in court. Table 2 summarises the way they report their conclusions.

A BN is a natural representation to help compute the conditional probabilities in the numerator and the denominator of a LR. BNs are particularly suitable to represent uncertain causal relations between relevant variables in a domain. In the context of evidence evaluation in crime investigation, they are used to describe how hypothetical situations and events may lead to observable evidence. In such a model, the hypotheses of interest typically correspond to one or more root nodes and a piece of evidence to a leaf node.

Consider, for example, an investigation of suspicious death where the victim died from a stab wound. The crime investigators have arrested a suspect, whom they believe has stabbed the victim to death. And, an examination of the suspect's clothes revealed blood splatter matching the victim's dna. The suspect's defence attorney claims that the suspect did not stab the victim, but instead discovered the victim's body and tried to revive him. It is assumed that the suspect does not come into contact with blood on a regular basis, under either hypothesis. In this case, the prosecution hypothesis is specified by $\{x_s, \overline{x_h}, \overline{x_b}\}$, the defense hypothesis is $\{\overline{x_s}, x_h, \overline{x_b}\}$, and the evidence can be represented as x_m. The likelihood ratio $\frac{P(x_m|x_s\overline{x_h}\overline{x_b})}{P(x_m|\overline{x_s}x_h\overline{x_b})}$ can be calculated easily by means of BN software, such as Hugin.

Discussion

While there is no universally accepted approach to evidence evaluation within the forensic science and crime investigation community, there are some important benefits to Bayesian evidence evaluation, which stem from combining the advantages of statistical and logic-based approaches.

As a statistical method, a BN can be employed to compare the relative support for alternative hypotheses by given pieces of evidence. This differentiates the Bayesian approach to evidence evaluation from logic-based ones as the latter are typically restricted to roles such as abductive reasoning about plausible hypotheses, reasoning about the implication and validity of arguments and explanation generation. And while the latter roles are important, their remains a crucial need for decision support in the area of comparing the plausibility of hypotheses under consideration (Cook *et al.* 1998).

Similar to logic-based approaches and contrary to most conventional statistical inference methods, a BN can model causal explanations for available evidence (Lacave & Díez 2002; Pearl 1988). Such causal structures are important because they enable the forensic expert to justify the results of evidence evaluation in court and identify how it relates to the plausible crime scenarios under consideration. Eventually, these structures can be transformed into arguments that constitute a basis for legal proceedings in criminal cases (Bex *et al.* 2003; Schum 1994).

There are some important objections to the Bayesian approach, however. Although, as explained in Section , BNs reduce the requirement for probabilistic knowledge, collecting sufficient and suitable conditional probability distributions remains an important stumbling block in many practical applications of BNs (Druzdzel & van der Gaag 2000). There are processes that produce certain types of evidence, which are well understood: e.g. the effect of mixtures of DNA material on the corresponding profile (Mortera, Dawid, & Lauritzen 2003). However, for many types of hypothesis and evidence, it is difficult to identify the conditional probability distributions from the underlying physical processes. For example, it is very difficult to categorise and relate types of contact between two people and the amount of trace material that is transferred between those people as a consequence (Aitken, Taroni, & Garbolino 2003).

In the latter case, experts may be able to provide estimates of the conditional probabilities based on their knowledge and experience. Such probabilities are called *subjective probabilities* because they reflect the personal opinion of one expert. The difficulty of obtaining point estimates of probabilities from experts has been widely reported (Kahneman,

Table 3: Sign operations

$\otimes$	+	0	-	?
+	+	0	-	?
0	0	0	0	0
-	-	0	+	?
?	?	0	?	?

$\oplus$	+	0	-	?
+	+	+	?	?
0	+	0	-	?
-	?	-	-	?
?	?	?	?	?

Slovic, & Tversky 1985; Zimmer 1983). Moreover, it has been reported that verbal expressions of probabilistic uncertainty were more accurate than numerical values in estimating the frequency of multiple attributes (Druzdzel & Henrion 1993; Zimmer 1986). This has led to the development of a range of qualitative approaches to perform Bayesian inference.

Qualitative Bayesian Inference

Qualitative Certainty Networks (QCNs) (Parsons & Mamdani 1993) constitute qualitative abstractions of various probabilistic and possibilistic reasoning techniques. Similar to a BN, a QCN consists of a DAG that describes the independencies between variables. But instead of a conditional probability table, each arc $A \rightarrow C$ between two variables A and C in the DAG is associated with a set of so-called qualitative derivatives, one for each pair of values a_i and c_j from the domains of A and C.

Definition 1 *Given a QCN containing two variables A and C connected by an arc $A \rightarrow C$, a value a_i of A and a value c_j of C, the qualitative derivative $[\frac{dP(c_j)}{dP(a_i)}]$ has the value $+$ iff for all values $a_k \neq a_i$ of A and all assignments $\mathbf{x}$ of the parent variables of C in the DAG other A.:*

$$P(c_j|a_i, \mathbf{x}) \geq P(c_j|a_k, \mathbf{x}) \quad (1)$$

Informally, the qualitative derivative $[\frac{dP(c_j)}{dP(a_i)}]$ is said to be positive iff C is more likely to take c_j when A is more likely to take a_i. The definitions for $[\frac{dP(c_j)}{dP(a_i)}] = 0$ and $[\frac{dP(c_j)}{dP(a_i)}] = -$ can be specified in the same way as Definition 1 by replacing $\geq$ with $=$ and $\leq$ respectively. If $[\frac{dP(c_j)}{dP(a_i)}]$ does not equal $+$, 0 or $-$, it is said to be ambiguous and takes value ?.

Let $[\Delta P(a_i)]$ denote a change in the sign of the probability of variable A taking value a_i. Then, such a change can be propagated along an arc by:

$$[\Delta P(c_j)] = [\frac{dP(c_j)}{dP(a_i)}] \otimes [\Delta P(a_i)]$$

where $\otimes$ denotes sign multiplication. The effect of multiple sign changes are combined with sign addition $\oplus$. Both operations are defined in Table 3.

In the remainder of this paper, all variables are assumed to have boolean domains. In this case, the domain of a variable Y can be denoted as $\{y, \overline{y}\}$. This allows the notation to be simplified substantially because a single qualitative derivative implies all the others in these circumstances. That is, when C takes values c and $\overline{c}$ and A takes values a and $\overline{a}$, then $[\frac{dP(c)}{dP(a)}] = +$ implies that $[\frac{dP(c)}{dP(\overline{a})}] = -$, $[\frac{dP(\overline{c})}{dP(a)}] = -$ and $[\frac{dP(\overline{c})}{dP(\overline{a})}] = +$ because $P(c|a, \mathbf{x}) \geq P(c|\overline{a}, \mathbf{x})$ implies that $P(c|\overline{a}, \mathbf{x}) \leq P(c|a, \mathbf{x})$, $P(\overline{c}|a, \mathbf{x}) \leq P(\overline{c}|a, \mathbf{x})$ and $P(\overline{c}|\overline{a}, \mathbf{x}) \geq P(\overline{c}|a, \mathbf{x})$. In what follows, the sign of qualitative derivative $[\frac{dP(c)}{dP(a)}]$ will be denoted by $[S(A, C)]$. Note that, as shown by Parsons (Parsons 1995), the qualitative derivative $[\frac{dP(c)}{dP(a)}]$ of variables A and C with a boolean domains $\{a, \overline{a}\}$ and $\{c, \overline{c}\}$ equals the sign of the qualitative influences in Qualitative Probabilistic Networks (Wellman 1990).

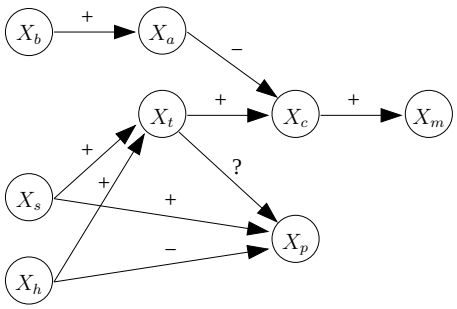


Figure 2: Sample QPN/QCN

It follows from the above that $[S(A, C)] = [S(C, A)]$. This property has enabled Druzdzel and Henrion (Druzdzel & Henrion 1993) to devise an algorithm to propagate a change in sign of any assignment h in these QPN/QCNs. In a nutshell, this algorithm first initialises the change in likelihood for every assignment to 0. Then, the algorithm propagates the sign change of h to every other node in the QPN/QCN, via every path, from H to other nodes, that is not blocked. The sign change in a node A that is directly connected to a node C via an arc $A \rightarrow C$ or $A \leftarrow C$ on a path that is not blocked is propagated by:

$$[\Delta P(c)] = [S(A, C)] \otimes [\Delta P(a)]$$

Given that the sign change has been provisionally set to $[\Delta P(c)]_{\text{current}}$ by the algorithm and that a newly considered sign change equals $[\Delta P(c)]_{\text{influence}}$, then the combined sign change equals:

$$[\Delta P(c)] = [\Delta P(c)]_{\text{current}} \oplus [\Delta P(c)]_{\text{influence}}$$

Figure 2 shows a QPN/QCN describing a qualitative version of the BN of Figure 1. This model indicates that both the hypothesis that the suspect stabbed the victim (x_s) and the hypothesis that the suspect tried to determine whether he could help the victim by examining the body (x_h) justify the observation of a blood splatter matching the victim's dna on the suspect (x_m). It also suggests that the blood splatter is more likely to be a projected bloodstain (x_p) if the suspect stabbed the victim and less likely to be a project bloodstain ($\neg x_p$), i.e. a contact stain, if the suspect examined the victim's body.

Consider, for instance, that blood splatter on the suspect matching the victim's dna is observed. This corresponds to $[\Delta P(x_m)] = +$. Druzdzel and Henrion's algorithm propagates this change as follows: $[\Delta P(x_c)] = +$, $[\Delta P(x_a)] =$

$-$, $[\Delta P(x_b)] = -$, $[\Delta P(x_p)] =?$, $[\Delta P(x_s)] = +$ and $[\Delta P(x_h)] = +$. Note that the change $[\Delta P(x_p)] =?$ is not propagated to X_s and X_h because the paths $X_t \rightarrow X_p \leftarrow X_s$ and $X_t \rightarrow X_p \leftarrow X_h$ are blocked.

Evidentiary reasoning with a QCN

The likelihood ratio approach is not directly applicable to QCNs as they do not provide any numerical information with which to calculate the fraction. However, the spirit of the approach can be applied by comparing the effect of different sets of hypotheses on a given piece of evidence. Let $\Delta_d P(e)$ denote the change of obtaining a piece of evidence e if a given defence scenario is true and let $\Delta_p P(e)$ denote the change of obtaining a piece of evidence e if a given prosecution scenario is true. The relative effect of the defence scenario on $P(e)$ compared to that of the prosecution scenario indicates how much more or less the evidence supports the defence scenario compared to the prosecution one.

Because QCNs can only provide information on the signs of $\Delta_d P(e)$ and $\Delta_p P(e)$, which indicates a negative, zero, positive or ambiguous change in $P(e)$, their comparisons rarely yield useful information in practice. Effective defendants and prosecutors tend to hypothesise scenarios that provide seemingly reasonable explanations for the available evidence. In such situations, the hypotheses being compared both render the available evidence more likely: i.e. $[\Delta_d P(e)] = [\Delta_p P(e)] = +$.

One approach to address this issue is the use of so-called product synergy in QPNs. Let A, B and C be variables connected by arcs $A \rightarrow C$ and $B \rightarrow C$ in a QPN. Then, there is said to be negative product synergy between A and B for a value c of C iff:

$$P(c|ab\mathbf{x})P(c|\overline{a}\overline{b}\mathbf{x}) - P(c|a\overline{b}\mathbf{x})P(c|\overline{a}b\mathbf{x}) \leq 0 \qquad (2)$$

It has been shown that negative product synergy enables hypotheses to be explained away (Wellman & Henrion 1993). Clearly, given (2), the observation of c implies that an increase in the likelihood of a makes b less likely and vice versa. Thus, if there is negative product synergy between A and B for a value c of C, the observation of c entails that $[S(A,B)] = -$, thereby enabling evidence that confirms a to be used to undermine b. But while negative product synergies provide the mechanism to infer counter-arguments within the framework of a QPN, it leaves much room for ambiguity. Indeed, the negative qualitative derivative implied by a negative product synergy works both ways and evidence that confirms *either* hypothesis undermines the other.

In the QPN/QCN of Figure 2, the prosecution hypothesis corresponds to $\{[\Delta P(x_s)] = +, [\Delta P(x_b)] = -\}$ and the defense hypothesis to $\{[\Delta P(x_h)] = +, [\Delta P(x_b)] = -\}$. Both hypotheses yield $[\Delta P(x_m)] = +$. As such, a QPN/QCN is not able to differentiate between both hypotheses.

QCN with orders of magnitude

Another approach to refine the reasoning that can be accomplished with a QCN involves the use qualitative or semi-quantitative representations of the magnitudes of changes in probabilities and qualitative derivatives. In such an approach, each direction of change of a probability $[\Delta_i P(a)]$ and each qualitative derivative $[S(A,C)]$ is also associated with a magnitude of change. These are denoted as $|\Delta_i P(a)|$ and $|S(A,C)|$ respectively. Note that while $[\frac{dP(c)}{dP(a)}] = [\frac{dP(a)}{dP(c)}]$ in a QCP with an arc $A \rightarrow C$, it is not necessarily the case that $|\frac{dP(c)}{dP(a)}|$ equals $|\frac{dP(a)}{dP(c)}|$. Therefore, the propagation mechanisms discussed in the remainder of the paper only apply in the direction of the arcs.

Table 4: Multiplication of relative orders of magnitude

rel_1 \ rel_2	$\otimes$	$\approx$	$\sim$	$\simeq$	$\ll$	$\gg$
	$\approx$	$\approx$	$\sim$	$\simeq$	$\ll$	$\gg$
	$\sim$	$\sim$	$\sim$	U	$\ll$	$\gg$
	$\simeq$	$\simeq$	U	U	U	U
	$\ll$	$\ll$	$\ll$	U	$\ll$	U
	$\gg$	$\gg$	$\gg$	U	U	$\gg$

A range of order of magnitude reasoning (OMR) techniques has been devised to express magnitudes in a qualitative manner (Raiman 1991). There are two types of OMR: relative OMR and absolute OMR. *Relative OMR* defines orders of magnitudes of variables by relating them to one another. For example, Dague's Relative OMR, named ROM, relates variables to one another using four types of ordering relations between pairs of quantities: x is close to y (denoted $x \approx y$), x is comparable to y ($x \sim y$), x is distant from y ($x \simeq y$) and x is negligible compared to y ($x \ll y$) (Dague 1993). Parsons (Parsons 2003) has devised a method that can be employed to propagate such order of magnitude information in a QCN. Let A, B, C and D be variables in a QCN in which A and B are connected by an arc $A \rightarrow B$ and C and D by an arc $C \rightarrow D$. Then, it can be shown that if $|S(A,B)|rel_1|S(C,D)|$ and $|\Delta_i P(A)|rel_2|\Delta_j P(C)|$, then rel_3 in $|S(A,B)|.|\Delta_i P(A)|rel_3\ |S(C,D)|.|\Delta_j P(C)|$ is given by Table 4.

The sample QCN of Figure 2, for instance, may include the ordering relation $|S(X_s,X_t)| \gg |S(X_h,X_t)|$. When comparing the effects of the prosecution and defence hypotheses, it can be assumed that the strength of the hypotheses are of a similar order of magnitude: i.e. $\Delta_p P(x_s) \approx \Delta_d P(x_h)$ and $\Delta_p P(x_b) \approx \Delta_d P(x_b)$. Based on these inputs, ROM based QCN infers that

$$\begin{aligned} |\Delta_p P(x_s)| \otimes |S(X_s,X_t)| = \\ |\Delta_p P(x_t)| \gg & |\Delta_d P(x_t)| \\ = & |\Delta_d P(x_h)| \otimes |S(X_h,X_t)| \end{aligned}$$

because according to Table 4, $\approx \otimes \gg$ yields $\gg$. Similarly, ROM based QCN infers $|\Delta_p P(x_c)| \gg |\Delta_d P(x_c)|$ and $|\Delta_p P(x_m)| \gg |\Delta_d P(x_m)|$. Thus, QCN extended with ROM computes that the discovery of transfer evidence supports the prosecution hypothesis more strongly than the defence hypothesis.

A difficulty arises, however, when the model were to be extended with an additional node, say, X_o, which describes

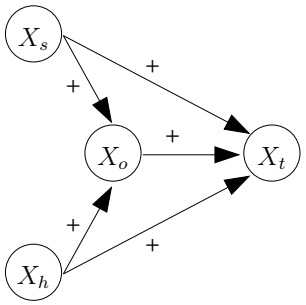


Figure 3: A more difficult application of ROM

Table 5: Intervals for qualitative derivatives

Symbol	Name	Definition
SP	Strong positive	$]1, \alpha]$
WP	Weak positive	$[\alpha, 0[$
Z	Zero	0
WN	Weak negative	$]0, -\alpha]$
SN	Strong negative	$[-\alpha, -1[$

whether or not the suspect touched the victim. The affected part of the QPN/QCN is shown in Figure 3. This change complicates the application of ROM considerably. This does not only introduce a requirement of additional pairwise ordering information regarding qualitative derivatives affecting the same node, but also between qualitative derivatives affecting different nodes in order to determine whether there are causal chains between the hypothesis nodes and X_t that have a dominant effect. In this way, the need for ordering information can grow exponentially with the network, thereby making this method impractical.

Absolute OMR defines magnitudes as numerical intervals rather than by means of ordering relations between individual magnitudes. Parsons (Parsons 1995) has briefly examined the use of absolute OMR for integration in QCNs and suggested the interval distribution shown in Table 5 for each qualitative derivative magnitude $|S(A, C)|$ and the interval distribution shown in Table 6 for each variable change $\Delta_i P(a)$. He has also shown, using conventional interval calculus, that if $\alpha = 0.5$ and $\beta = \frac{1}{3}$, the results for multiplication and addition of intervals are given in Table 7 and Table 8 respectively. Note that in these tables, $[I_1, I_2]$ refers to the combined interval containing both I_1 and I_2, and that empty cells refer to impossible combinations of values.

Figure 4 shows a version of the QPN/QCN of Figure 1 with the aforementioned absolute order of magnitude scale. In this approach, the prosecution scenario corresponds to $|\Delta_p P(x_s)| = CP$ and $|\Delta_p P(x_b)| = CN$ and the defence scenario to $|\Delta_d P(x_h)| = CP$ and $|\Delta_d P(x_b)| = CN$. These values can be propagated as follows:

Table 6: Intervals for variable changes

Symbol	Name	Definition
CP	Complete positive	1
BP	Big positive	$]1, 1-\beta]$
MP	Medium positive	$[1-\beta, \beta]$
LP	Little positive	$[\beta, 0[$
Z	Zero	0
LN	Little negative	$]0, -\beta]$
MN	Medium negative	$[-\beta, \beta-1]$
BN	Big negative	$[\beta-1, -1[$
CN	Complete negative	-1

Table 7: Interval multiplication

$\otimes$	SP	WP	Z	WN	SN
CP	$[BP, MP]$	$[MP, LP]$	Z	$[MN, LN]$	$[BN, MN]$
BP	$[BP, MP]$	$[MP, LP]$	Z	$[MN, LN]$	$[BN, MN]$
MP	$[MP, LP]$	$[MP, LP]$	Z	$[MN, LN]$	$[MN, LN]$
LP	LP	LP	Z	LN	LN
Z	Z	Z	Z	Z	Z
LN	LN	LN	Z	LP	LP
MN	$[MN, LN]$	$[MN, LN]$	Z	$[MP, LP]$	$[MP, LP]$
BN	$[BN, MN]$	$[MN, LN]$	Z	$[MP, LP]$	$[BP, MP]$
CN	$[BN, MN]$	$[MN, LN]$	Z	$[MP, LP]$	$[BP, MP]$

Table 8: Interval addition

$\oplus$	CP	BP	MP	LP	Z
CP					CP
BP				$[CP, BP]$	BP
MP			$[CP, MP]$	$[CP, MP]$	MP
LP		$[CP, BP]$	$[CP, MP]$	$[BP, LP]$	LP
Z	CP	BP	MP	LP	Z
LN		$[BP, MP]$	$[MP, Z]$	$[LP, Z]$	
MN		$[BP, Z]$	$[MP, Z]$		
BN		$[LP, Z]$			
CN					

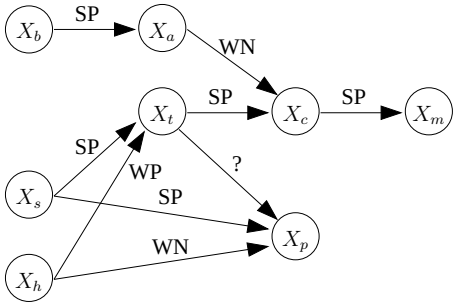


Figure 4: Sample QPN/QCN with absolute orders of magnitude

$$
\begin{aligned}
|\Delta_p P(x_t)| &= SP \otimes CP = [BP, MP] \\
|\Delta_p P(x_a)| &= SP \otimes CN = [BN, MN] \\
|\Delta_p P(x_c)| &= (WN \otimes [BN, MN]) \oplus (SP \otimes [BP, MP]) \\
&= [MP, LP] \oplus [BP, LP] = [CP, LP] \\
|\Delta_p P(x_m)| &= SP \otimes [CP, LP] = [BP, LP]
\end{aligned}
$$

The outcome of this analysis (i.e. $|\Delta_p P(x_m)| = [BP, LP]$) is that there is a small to substantial increase in likelihood to obtain a blood splatter from the suspect matching the victim's dna under the prosecution scenario. The hypotheses of the defence scenario can be propagated in the same way, resulting in $|\Delta_d P(x_m)| = [BP, LP]$. As such, this approach is not able to differentiate between both scenarios in this case.

Note that a basic QPN/QCN as defined in Section is a special case of a QPN/QCN with orders of magnitude. It employs the following intervals for both changes in likelihood of variables and qualitative derivatives:

$$
0 = [0] \qquad + = [0, 1] \qquad - = [-1, 0] \qquad ? = [-1, 1] \tag{3}
$$

Reducing over-abstraction

The survey of qualitative Bayesian inference methods has shown how these techniques can be applied to evidence evaluation in crime investigation. An important limitation of the approaches discussed in the survey is that, even with the introduction of order of magnitude calculi, they tend to produce very abstract results. Even in simple examples, such as the one used throughout this paper, the propagation of likelihood changes quickly yields intervals that are too wide for comparison. This problem of over-abstraction is inherent to all types of interval calculi, including those involving orders of magnitude and fuzzy sets. However, the nature of influences between variables in models for evidential reasoning in crime investigation exhibits certain features that enable the knowledge engineer to reduce the effects of qualitative abstraction.

Cause v context

The order of magnitude approaches discussed herein presume that each arc $A \rightarrow C$ implies that the likelihood of c is either proportionate or inversely proportionate to the likelihood of a. As such, these approaches are particularly well suited to model processes whereby A adds to or removes from the likelihood of$C$ (or vice versa). These are situations in which A is a direct *cause* of C (or vice versa). However, in evidential reasoning, this is not always the case. Certain variables, which will be called *context* variables, affect the process rather than the consequence. Figure 5 illustrates this distinction.

The crucial difference between a causal and a context variable of an influence is that the causal variable always affects the consequent, whereas the effect of the context on the consequent determines the magnitude with which the process takes place. Four types of context variables can be

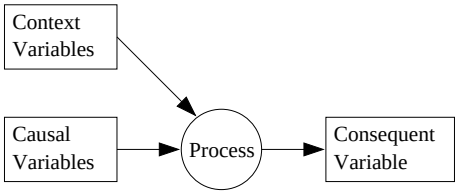


Figure 5: Cause v context

identified: *enablers*, which are conditions for the process to take place; *disablers*, which prevent the process from taking place; *amplifiers*, which increase the effect of the process; and *inhibitors*, which decrease the effect of the process.

The example used throughout this paper contains two context variables. Firstly, X_a is an inhibitor to $X_t \rightarrow X_c$. Here, the transfer of blood from the the victim to the suspect (X_t) generates blood traces that are relevant to the crime and may be retrieved by the investigators (X_c). Blood splatter on the suspect from a source unrelated to the crime (X_a) makes it less likely that the investigators will retrieve blood splatter related to the crime. Thus, the likelihood of x_a has a negative effect on the likelihood of x_c, but only if x_t is true to begin with. Secondly, X_t is an enabler to $X_s \rightarrow X_p$ and to $X_h \rightarrow X_p$. Both hypotheses, i.e. the suspect stabbed the victim (x_s) and the suspect examined the body of the victim (x_h), affect the pattern of the blood splatter that is transferred X_p. However, blood must be transferred from suspect to victim for there to be a pattern to examine.

Context variables can be identified in a conventional numerical BN as follows. Let A be a cause of C and B be a context variable, such that

$$
P(c|ab) = \alpha \quad P(c|a\bar{b}) = \beta \quad P(c|\overline{a}b) = \epsilon_1 \quad P(c|\overline{a}\bar{b}) = \epsilon_2.
$$

where $\epsilon_1 \approx 0$ and $\epsilon_2 \approx 0$. Then, B is an enabler if $\alpha > \beta = 0$, a disabler if $\beta > \alpha = 0$, an amplifier if $\alpha > \beta$ and an inhibitor if $\alpha < \beta$. As such, the likelihood of context variables assignments is not proportional to the likelihood of consequent variable assignments, which, in turn, makes the order of magnitude approaches unsuitable.

Renooij et. al. (Renooij, van der Gaag, & Parsons 2002) have extended the basic (i.e. sign-only) QPN approach with so-called *non-monotonic* influences. The sign of a non-monotonic influence changes with the assignment of another variable. For example, in the QPN of Figure 2, the signs of the qualitative derivatives $S(X_s, X_p)$ and $S(X_h, X_p)$ could be specified as follows:

$$
[S(X_s, X_p)] = \begin{cases} + & \text{if } x_t \\ 0 & \text{if } \overline{x_t} \end{cases} \quad \text{and} \quad [S(X_h, X_p)] = \begin{cases} - & \text{if } x_t \\ 0 & \text{if } \overline{x_t} \end{cases}
$$

Let $S_{\mathbf{x}}(A, C)$ denote the qualitative derivative in effect given the set of assignments $\mathbf{x}$ of the context variables of $A \rightarrow C$. Then, the above derivative signs can be denoted as: $[S_{x_t}(X_s, X_p)] = +$, $[S_{\overline{x_t}}(X_s, X_p)] = 0$, $[S_{x_t}(X_h, X_p)] = -$ and $[S_{\overline{x_t}}(X_h, X_p)] = 0$.

When the assignments of the context variables are not all know, then the smallest possible range of effects that includes all plausible contexts must be assumed (using the interval definitions of (3)). Thus, given an influence $A \rightarrow C$

and two sets of assignments $\mathbf{x}$ and $\mathbf{y}$ of the context variables of $A \rightarrow C$, then:

$$[S_{\mathbf{x}\cap\mathbf{y}}(A,C)] \subseteq [S_{\mathbf{x}}(A,C)] \cup [S_{\mathbf{y}}(A,C)]$$

It follows that if $\mathbf{x}'$ denotes a partial specification of the context variables of $A \rightarrow C$, then

$$[S_{\mathbf{x}'}(A,C)] = \bigcup_{\mathbf{x}'\subseteq\mathbf{x}} [S_{\mathbf{x}}(A,C)]$$

where the $\mathbf{x}$ are all assignments of the context variables of $A \rightarrow C$ such that $\mathbf{x}' \subseteq \mathbf{x}$. Note that it follows from (3) that:

$$0 \subset + \qquad 0 \subset - \qquad + \subset ? \qquad - \subset ?$$

Therefore, if in the example, the assignment of the context variable X_t of $X_s \rightarrow X_p$ and $X_h \rightarrow X_p$ is unknown, then:

$$[S(X_s,X_p)] = [S_{x_t}(X_s,X_p)] \cup [S_{\overline{x_t}}(X_s,X_p)] = + \cup 0 = +$$
$$[S(X_h,X_p)] = [S_{x_t}(X_h,X_p)] \cup [S_{\overline{x_t}}(X_h,X_p)] = - \cup 0 = -$$

which is identical to the original specification of the network in Figure 2.

Clearly, this idea can be generalised to absolute orders of magnitudes by using more precise interval definitions than those of (3). Generally speaking, given an influence $A \rightarrow C$ and two sets of assignments $\mathbf{x}$ and $\mathbf{y}$ of the context variables of $A \rightarrow C$, the context specific magnitudes of qualitative derivatives must adhere to:

$$|S_{\mathbf{x}\cap\mathbf{y}}(A,C)| \subseteq |S_{\mathbf{x}}(A,C)| \cup |S_{\mathbf{y}}(A,C)|$$

Categorical influences

Because Bayesian inference models in general, and qualitative abstractions of such models in particular, have an explanatory role in addition to a computational one, it is important that the structure of the network matches the way the human user would organise his/her arguments. For example, in practice, the arcs in BNs often describe causal relations between variables, even though that is not necessary. However, causal relations are often the most natural way of justifying analyses.

To improve the explanatory benefits of a QPN/QCN, additional variables that do not introduce any source of uncertainty may need to be introduced. In the original version of the BN shown in Figure 1, X_a is such a variable (Aitken, Taroni, & Garbolino 2003). Its probability distribution is defined as $P(x_a|x_b) = 1$ and $P(x_a|\overline{x_b}) = 0$. The variable describes that a certain background of a suspect (x_b) may constitute an alternative source of blood splatter on the suspect (x_a), which in turn inhibits the discovery of blood splatter matching the victim's dna on the suspect (in the hypothetical case that blood has been transferred from victim to suspect).

Categorical influences represent precisely this type of information in the restricted setting of a sign calculus (Parsons 1995; 2004). Table 9 displays sign multiplication in a setting where a qualitative derivative $[S(A,C)]$ can take values $++$ and $--$, indicating a categorical influence. Variable change signs $[\Delta P(a)]$ can take values $++$ and $--$, which describe that the variable increases to 1 or decreases to 0 respectively.

Table 9: Sign multiplication with categorical influences

$\otimes$	++	+	0	-	--	?
++	++	+	0	-	--	?
+	+	+	0	-	-	?
0	0	0	0	0	0	0
-	-	-	0	+	+	?
--	--	-	0	+	++	?
?	?	?	0	?	?	?

The approach can be generalised and integrated into the context of the absolute order of magnitude reasoning. In essence, a categorical influence $A \rightarrow C$ propagates any changes in $P(a)$ directly to $P(c)$. Formally, given a magnitude change $\|DeltaP(a)|$ and a categorical influence $S(A,C) = ++$, the magnitude change $|\Delta P(c)| = |\Delta P(a)|$. Similarly, given a magnitude change $|\Delta P(a)|$ and a categorical influence $S(A,C) = --$, $|\Delta P(c)| = -|\Delta P(a)|$.

Magnitudes

As illustrated by the example, the absolute order of magnitude scale discussed in Section is too abstract to derive a firm conclusion. The main problem with the use of absolute orders of magnitude is that every propagation of probability changes along an arc on a path from a hypothesis node to an evidence node involves an interval multiplication. And, every interval multiplication produces a result that tends to be wider than the constituent factors. Context variables and categorical influences can, to some extent, alleviate these issues. However, in more complex models, a more refined order of magnitude scale has to be employed.

One approach of defining absolute order of magnitude scales, which limits the amount of interval size expansion and facilitates flexible definition of absolute order of magnitude scales, is NAPIER (Nayak 1992). In NAPIER, the order of magnitude of a quantity is defined as the nearest lowest integer of the logarithm of that quantity. That is:

$$om(p) = \lfloor \log_b |p| \rfloor \tag{4}$$

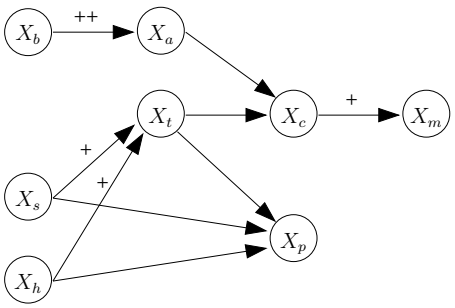
where b is the base of the logarithm. Thus, in this approach, magnitudes are defined by integers, where low integers indicate values closer to 1. In addition to the integer id of the order of magnitude, each qualitative derivative and magnitude change remains associated with a sign indicating the direction of change. An example of an order of magnitude scale and corresponding verbal qualifications of the corresponding values is shown in Table 10. Note that this table also includes the additional magnitudes $++$ and $--$ to denote categorical changes and influences discussed in Section
.

Using this approach, orders of magnitude for multiplication of two quantities p_1 and p_2 is given by (see (Nayak 1992)):

$$om(p_1 \times p_2) = [om(p_1) + om(p_2), om(p_1) + om(p_2) + 1]$$

Table 10: A sample absolute order of magnitude scale

Id	Description	Range
++/--	complete positive/negative	1 or -1
0	very strong positive/negative	$[0.8, 1[$
1	strong-very strong positive/negative	$[0.64, 0.8[$
2	strong positive/negative	$[0.51, 0.64[$
3	moderate-strong postive/negative	$[0.41, 0.51[$
4	moderate positive/negative	$[0.33, 0.41[$
5	weakly moderate positive/negative	$[0.26, 0.33[$
6	weak-weakly moderate postive/negative	$[0.21, 0.26[$
7	weak positive/negative	$[0.17, 0.21[$
8	very weak postive/negative	$[0.13, 0.17[$



Magnitudes and non−monotic influence signs:

$|S(X_s, X_t)| = 0$
$|S(X_h, X_t)| = 2$

$|S_{x_a}(X_t, X_c)| = 5$
$|S_{\overline{x_a}}(X_t, X_c)| = ++$

$|S(X_c, X_m)| = 0$

$[S_{x_t}(X_s, X_p)] = +$ $\quad |S_{x_t}(X_s, X_p)| = 0$
$[S_{\overline{x_t}}(X_s, X_p)] = 0$
$[S_{x_t}(X_h, X_p)] = -$ $\quad |S_{x_t}(X_h, X_p)| = 5$
$[S_{\overline{x_t}}(X_h, X_p)] = 0$

Figure 6: QPN/QCN with logarithmic order of magnitude scale, categorical influences and non-monotonic influences

Integrating the refinements

The refinements discussed above can now be combined in the ongoing example. Figure 6 shows the resulting QPN/QCN with integrated non-monotonic influences, categorical influences and the logarithmic order of magnitude scale of Table 10. With this approach, the prosecution scenario corresponds to $|\Delta_p P(x_s)| = (++)$ and $|\Delta_p P(x_b)| = (--)$ and the defence scenario to $|\Delta_d P(x_h)| = (++)$ and $|\Delta_d P(x_b)| = (--)$. These values can be propagated as follows:

$$|\Delta_p P(x_t)| = (++) \otimes (0) = (0)$$
$$|\Delta_p P(x_a)| = (++) \otimes (++) = (++)$$

The latter result implies that the magnitude $|S_{x_a}(X_t, X_c)| = ++$ is in effect. Therefore,

$$|\Delta_p P(x_c)| = (++) \otimes (0) = (0)$$
$$|\Delta_p P(x_m)| = (0) \otimes (0) = (0, 1)$$

Similarly,

$$|\Delta_d P(x_t)| = (++) \otimes (2) = (2)$$
$$|\Delta_d P(x_a)| = (++) \otimes (++) = (++)$$
$$|\Delta_d P(x_c)| = (++) \otimes (2) = (2)$$
$$|\Delta_d P(x_m)| = (2) \otimes (0) = (2, 3)$$

This result indicates that obtaining blood matching the victim's dna from the suspect's clothes is somewhat more likely under the prosecution scenario than under the defence scenario, which is consistent with our intuition. Note that this outcome does not entail a claim regarding the extent to which the case of the prosecution is shown, as that is ultimately to be decided in court. However, in combination with the supporting network providing causal explanations, it captures all the information that is relevant regarding this piece of evidence and its support of the alternative hypotheses. As such, qualitative representations of this type constitute a suitable basis upon which decision support systems for legal evidential reasoning may be built.

Conclusions and Future Work

This paper has discussed qualitative approaches to probabilistic reasoning. It has examined the need for qualitative probabilistic reasoning and shown how qualitative probabilistic inference methods can be employed to perform analyses similar to that of Bayesian evidence evaluation. However, it has been clarified that while the level of abstraction employed in the qualitative probabilistic reasoning enables the generation of intuitive explanations to justify decisions, it also prevents Bayesian-like evidence evaluation. A critical survey has presented a range of extensions designed to improve the precision of qualitative probabilistic reasoning techniques while maintaining their explanation generation ability. This has identified number of features that Bayesian models designed for evidentiary reasoning exhibit. By means of small examples, it has been shown that the each of the extensions can effectively describe some of these features. Last but not least, the paper has shown how these extensions can be integrated with one another.

The ideas discussed herein can form the foundation for future applications that model the lines of inquiries of crime investigators, both for educational and case management software. As such, the development and analysis of such software constitutes an interesting area of future research. Apart from this, some important theoretical concerns remain. One of these relates to the potential availability of knowledge of varying degrees of precision about the probability distributions involved in the models. Especially at the early stages of an investigation, the investigators can employ a broad range of evidence types, including some that can benefit the investigation but may not be admissible in court. These include: hearsay, witness testimony, records and recordings, psychological profiles and, of course, the entire spectrum of physical evidence. The lack of knowledge regarding reliability and accuracy varies considerably between these different types of evidence. Thus, future works should examine if and how these can be represented and integrated in a single model.

Acknowledgments

This work has been partially supported by the UK Nuffield Foundation grant NAL/32730.

References

Aitken, C.; Taroni, F.; and Garbolino, P. 2003. A graphical model for the evaluation of cross-transfer evidence in dna profiles. *Theoretical Population Biology* 63:179–190.

Bex, F.; Prakken, H.; Reed, C.; and Walton, D. 2003. Towards a formal account of reasoning about evidence, argument schemes and generalizations. *Artificial Intelligence and Law* 11:125–165.

Biedermann, A., and Taroni, F. 2006. Bayesian networks and probabilistic reasoning about scientific evidence when there is a lack of data. *Forensic Science International* 157(2–3):163–167.

Cook, R.; Evett, I.; Jackson, G.; Jones, P.; and Lambert, J. 1998. A model for case assessment and interpretation. *Science and Justice* 38(3):151–156.

Dague, P. 1993. Symbolic reasoning with relative orders of magnitude. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1509–1514.

Druzdzel, M., and Henrion, M. 1993. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the 11th Conference on Artificial Intelligence*, 548–553.

Druzdzel, M., and van der Gaag, L. 2000. Building probabilistic networks: where do the numbers come from? *IEEE Transactions on Knowledge and Data Engineering* 12(4):481–486.

Halliwell, J., and Shen, Q. 2002. Towards a linguistic probability theory. In *Proceedings of the 11th International Conference on Fuzzy Sets and Systems*, 596–601.

Kahneman, D.; Slovic, P.; and Tversky, A. 1985. *Judgement Under Uncertainty*. Cambridge University Press.

Keppens, J., and Schafer, B. 2006. Knowledge based crime scenario modelling. *Expert Systems with Applications* 30(2):203–222.

Keppens, J.; Shen, Q.; and Lee, M. 2005. Compositional bayesian modelling and its application to decision support in crime investigation. In *Proceedings of the 19th International Workshop on Qualitative Reasoning*, 138–148.

Lacave, C., and Díez, F. 2002. A review of explanation methods for bayesian networks. *The Knowledge Engineering Review* 17:107–127.

Mortera, J.; Dawid, A.; and Lauritzen, S. 2003. Probabilistic expert systems for dna mixture profiling. *Theoretical Population Biology* 63:191–205.

Nayak, P. 1992. Order of magnitude reasoning using logarithms. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 201–210.

Parsons, S., and Mamdani, E. 1993. On reasoning in networks with qualitative uncertainty. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*.

Parsons, S. 1995. Refining reasoning in qualitative probabilistic networks. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*.

Parsons, S. 2003. Qualitative probability and order of magnitude reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 11(3):373–390.

Parsons, S. 2004. On precise and correct qualitative probabilistic inference. *International Journal of Approximate Reasoning* 35(2):111–135.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan-Kaufmann.

Raiman, O. 1991. Order of magnitude reasoning. *Artificial Intelligence* 51:11–38.

Renooij, S.; van der Gaag, L.; and Parsons, S. 2002. Context-specific sign-propagation in qualitative probabilistic networks. *Artificial Intelligence* 140:207–230.

Schum, D. 1994. *Evidential foundations of probabilistic reasoning*. John Wiley and Sons.

Wellman, M., and Henrion, M. 1993. Explaining explaining away. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3):287–292.

Wellman, M. 1990. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence* 44:257–303.

Zimmer, A. 1983. Verbal vs. numerical processing of subjective probabilities. In Scholz, R., ed., *Decision making under uncertainty*. North-Holland, Amsterdam.

Zimmer, A. 1986. What uncertainty judgements can tell about the underlying subjective probabilities. In Kanal, L., and Lemmer, J., eds., *Uncertainty in Artificial Intelligence*. North-Holland, Amsterdam.

Learning Domain Theories via Analogical Transfer

Matthew Klenk and Kenneth D. Forbus

Qualitative Reasoning Group, Northwestern University
2133 Sheridan Road, Evanston, IL 60208 USA
{m-klenk, forbus}@northwesterrn.edu

Abstract

Learning domain theories is an important challenge for qualitative reasoning. We describe a method for learning new domain theories by analogy. We use analogies between pairs of problems and worked solutions to create a mapping between the familiar and the new domains, and use this mapping to conjecture general knowledge about the new domain. After some knowledge has been learned about the new domain, another analogy is made between the domain theories themselves providing conjectures about the new domain. An experiment is described where the system learns to solve rotational kinematics problems by analogy with translational kinematics problems, outperforming a version of the system that is incrementally given the correct domain theory.

Introduction

Progress in qualitative reasoning has led to a variety of techniques for model formulation, making predictions, performing diagnosis, and handling other tasks. However, little effort has focused on the process of learning domain theories. To be sure, in some cases hand-engineering domain theories is sufficient. However, this can be a very time-consuming process, requiring considerable effort. Being able to re-use this investment by automatically constructing theories for similar domains could be of great practical value. Furthermore, there is ample evidence that people heavily use analogy to learn new domains (Gentner & Gentner 1983; Gentner 2003). Systems that learn domain theories by analogy could be used to model human learning.

Falkenhainer's (1988) PHINEAS system was the first QR system to address this problem. Based on the hypothesis that diagnosis, explanation and theory formation are all intertwined, PHINEAS used *similarity-driven* explanation to show how analogy can be used to develop new theories about specific situations. As a learning agent works in a new domain, it should be able to transfer knowledge from previous well understood domains. Falkenhainer called the inability to offer a best guess or apply knowledge across domains the *adaptability problem*.

Textbook authors routinely exploit human adaptability (Shive & Weber 1982). In the linear kinematics section of the textbook used for this study (Giancoli 1991), there are eight worked out examples, *worked solutions*, which show all of the different ways in which the four linear kinematics equations can be used. But in the later rotational kinematics section, there are only two worked solutions. Furthermore, two of the rotational kinematics equations are not part of any worked solutions in the book. The summary section of rotational motion chapter invites the learner to use analogy to fill in the details: "The dynamics of rotation is analogous to the dynamics of linear motion" (p. 197, Giancoli 1991). This is common practice in textbooks, and analogies between domains form the basis of system dynamics (Olson 1966; Shearer *et al.* 1967).

This paper describes how analogies between worked solutions can be used to learn domain theories. Our strategy is itself analogous to that used in PHINEAS, which used comparisons of (simulated) behavior to create an initial cross-domain mapping that was subsequently used to create a partial theory for the new domain. It differs, however, in several significant ways: (1) We use analogies between worked solution pairs to drive the process, (2) We are learning quantitative, rather than qualitative, domain theories, which requires very different verification work, and (3) We are using a more psychologically plausible retrieval mechanism. While our current work focuses on quantitative domain theories, our method should also be usable for qualitative domain theories as well.

We start by describing our representations and problem-solver. Next we review the ideas of structure-mapping theory and our computational models which are used in this work. Then we describe our learning method, and present an experiment showing that it can learn rotational kinematics by analogy with translational kinematics, and do so faster than a system that is told the laws of the domain incrementally. We close with a discussion of related work and future plans.

Representation and Problem Solving

Representing physics problems requires a broad background of everyday knowledge, including the object and event types found in such problems. We use the ResearchCyc[1] knowledge base contents, augmented with our own extensions, as our starting point. Our extensions

[1] http://research.cyc.com/

```
(isa Car-2-6 Automobile)
(isa Acc-2-6
  TransportWithMotorizedLandVehicle)
(objectStationary (StartFn Acc-2-6) Car-2-6)
(primaryObjectMoving Acc-2-6 Car-2-6)
(valueOf
  ((QPQuantityFn Distance) Car-2-6 Acc-2-6)
  (Meter 30))
...
(query (valueOf ((QPQuantityFn Time-Quantity)
  Acc-2-6) Duration-2-6))
```

Figure 1: Problem 2-6 Representation (sample)

concern QP theory (Forbus 1984) and problem-solving strategies, and are small compared to the 30,000+ concepts and 8,000+ predicates already defined in the KB. Thus, objects, relations, and events that appear in physics problems such as "rotor", "car", and "driving" are already defined in the ontology for us, rather than being created specifically for this project.

Example Problem and Worked Solution

All problems and worked solutions used in this work were taken from the same physics textbook (Giancoli 1991). Problems are defined as cases. Consider the problem of "How long does it take a car to travel 30m if it accelerates from rest at a rate of 2 m/s^2?" (Example 2-6, p. 26). This problem is represented in our system as a case of 10 facts, a subset of which appears in Figure 1.

Worked solutions are represented at the level of examples found in textbooks, which is more abstract than a proof or problem-solving trace. For example, the worked solution for problem 2-6 consisted of four steps:

1. Categorize the problem as a constant acceleration linear mechanics problem
2. Instantiate the distance by velocity time equation ($d = v_i t + .5at^2$)
3. Because the car is stationary at the start of the event infer that its velocity is zero ($v_i = 0$ m/s)
4. Solve the equation for t (t = 5.8s)

Figure 2 shows how step 3 is represented.

```
(isa Gia-2-7-Step-3 WorkedSolutionStep)
(hasSteps Gia-2-7-WS Gia-2-7-Step-3)
(priorStep Gia-2-7-Step-3 Gia-2-7-Step-2)
(stepType Gia-2-7-Step-3 AssumingValue)
(stepUses Gia-2-6-WS-Step-3
 (objectStationary (StartFn Acc-2-6) Car-2-6))
(stepResult Gia-2-6-WS-Step-3
 (valueOf
    (AtFn ((QPQuantityFn Speed) Car-2-6)
    (StartFn Acc-2-6))
  (MetersPerSecond 0)))
```

Figure 2: Problem 2-6 worked solution step 3

Domain Theories

Our domain theories consist of *encapsulated histories* (Forbus 1984) representing equations. Encapsulated histories are templates describing pieces of histories (Hayes 1978). They were motivated by two concerns. First, some phenomena are best described by discontinuous

```
(def-encapsulated-history
 VelocityByTime-1DConstantAcceleration
  :participants
 ((theObject :type PointMass)
  (theEvent  :type Constant1DAccelerationEvent))
 :conditions
 ((primaryObjectMoving theEvent theObject))
 :consequences
 ((equationFor VelocityByTime
  (mathEquals
     (AtFn (Speed theObject)
           (EndFn theEvent))
      (PlusFn (AtFn (Speed theObject)
                    (StartFn theEvent))
       (TimesFn
          (AtFn (Acceleration theObject) theEvent)
           (Time-Quantity theEvent)))))))
```

Figure 3: Example Encapsulated History

patterns of events (e.g., collisions). Second, they permit constraints to be placed on time itself, which is not possible for model fragments, given their semantics (i.e., time is implicit, and their consequences hold throughout whatever period they are active). Equations like the velocity/time law above hold over events (e.g., translational motion under constant acceleration), and hence encapsulated histories are the appropriate mechanism for describing the conditions under which they hold.

Figure 3 illustrates the encapsulated history representing the equation of velocity as a function of time ($v_f=v_i+at$). There are two participants, `theObject` and `theEvent`, which must satisfy their type constraints, the abstractions `PointMass` and `Constant1DAccelerationEvent` respectively. Furthermore, the conditions of the encapsulated history must be satisfied in order to instantiate it and conclude its consequences. In this case, it is necessary that `theObject` be the object moving in `theEvent`. The compound form shown in Figure 3 is automatically translated into a set of predicate calculus facts. While the consequence of this encapsulated history is a quantitative equation, the same representation could be used to represent qualitative relationships. Similarly, this technique should be adaptable to learning model fragments as well.

Solving a Problem

Our system solves for quantities in three ways. First, the quantity may already be known as part of the problem. Second, rules can be used to apply modeling assumptions, i.e., "Objects at rest have no velocity". Third, an encapsulated history may be instantiated that results in an equation containing the sought after quantity. This is done by satisfying the participant constraints and the encapsulated history conditions statements in the problem. Once the encapsulated history has been instantiated, the system solves for the other quantities in the equation, and then attempts to solve the equation for the original parameter. The algebra routines are based upon the system in Forbus and de Kleer (1993). Both the problem-solving strategies and the mathematics knowledge are fixed in the current system, and cannot be extended via learning.

Structure-mapping and Analogy

We use Gentner's (1983) structure-mapping theory, which postulates that analogy and similarity are based on structural alignment between two structured representations (the *base* and *target*) to find the maximal structurally consistent match between them. A structurally consistent match must satisfy the constraints of *tiered-identicality*, *parallel connectivity*, and *one-to-one mapping*. Tiered-identicality constraint provides a strong preference for only allowing identical predicates to match, but allows for exceptions, when doing so would enable a much larger structure to match. The parallel connectivity constraint says that if two statements are matched then their arguments must also match. One-to-one mapping constraint requires that each element in the base corresponds to at most one element in the target, and vice versa. To explain why some analogies are better than others, structure-mapping uses the principle of *systematicity*: a preference for mappings that are highly interconnected and contain deep chains of higher order relations.

The Structure Matching Engine (SME) simulates the process of analogical matching between a base and target (Falkenhainer *et al.* 1989). The output of this process is one or more *mappings*. A mapping is a set of *correspondences* representing a construal of what items (*entities* and *expressions*) in the base go with what items in the target. Mappings include a *structural evaluation score* indicating the strength of the match, and *candidate inferences* which are conjectures about the target using expressions from the base which, while unmapped in their entirety, have subcomponents that participate in the mapping's correspondences. SME operates in polynomial time, using a greedy algorithm (Forbus & Oblinger, 1990).

MAC/FAC (Forbus *et. al.* 1994) models similarity-based retrieval. The inputs are a case, the *probe*, and a library of cases. The first stage (MAC) uses a computationally cheap, non-structural matcher to filter candidates from a pool of memory items, returning up to three if they are very close. The second stage (FAC) uses SME to compare the cases retruned by MAC to the probe and returns the best candidate (or candidates, if they are very similar). Both SME and MAC/FAC have been used as performance systems in a variety of domains and as cognitive models to account for a variety of psychological results (Forbus 2001).

Different domains are often represented using different predicates, especially when they are first being learned and underlying commonalities with previous knowledge have not yet been found. *Minimal ascension* (Falkenhainer 1988) is one method for matching non-identical predicates. If two predicates are part of a larger aligned structure and share a close common ancestor in the taxonomic hierarchy, then SME can include them in the mapping. For example, given the statements in Figure 4, if the `stepUses` statements are aligned as well as the `Step-Base` and `Step-Target`, `Obj-Base` and `Obj-Target`, and `Event-Base` and `Event-Target`, then SME will attempt to match `primaryObjectMoving` with `objectRotating`. They are siblings in the ResearchCyc ontology, and hence minimal ascension allows them to be placed into correspondence.

Base Expression:

```
(stepUses Step-Base
  (primaryObjectMoving Event-Base Obj-Base))
```

Target Expression:

```
(stepUses Step-Target
  (objectRotating Event-Target Obj-Target))
```

Figure 4: Minimal Ascension maps `primaryObjectMoving` to `objectRotating`

Analogical Learning of Domain Theories

Our system learns a domain theory by using multiple analogies. Learning is invoked when it fails to solve a problem. After failing to solve a problem, the system is given a worked solution for that problem, as a student might get out of a textbook. It uses this worked solution to create conjectures about knowledge in the new domain, using the algorithm outlined in Figure 5. The case library contains a set of worked solutions from the known domain. First, the worked solution for the failed problem is used as a probe to MAC/FAC, to retrieve an analogous worked solution from memory. A comparison is made using SME, with the retrieved worked solution constituting the base and the worked solution for the failed problem as the target. The mappings SME produces are then combined to create a *domain mapping*. The reason for combining multiple mappings is that each mapping often covers only some aspects of the solution. The best mapping is used as a starting point, with correspondences drawn from the others included only if they do not violate the one-to-one

1. Retrieve analog using the target worked solution as a probe in MAC/FAC
2. Use SME to create a match between the analog and the worked solution
3. Retrieve correspondences from resulting mappings
4. Create domain mapping by selecting correspondences in which the base element appears in the base domain theory
5. Initialize target domain theory using these correspondences
6. Use SME to create a match between the base and the target theories constrained by the domain mapping
7. Transfer domain theory using the candidate inferences
8. Verify learned domain theory by attempting the failed problem again
9. If failure, go once more to step 1. Otherwise, accept new target domain knowledge as correct

Figure 5: Analogical Domain Learning

constraint.

When the system gets the first problem in a new domain, its theory for that domain is empty. The candidate inferences for the domain mapping thus become the basis for a new domain theory. We currently require that every concept in the encapsulated history is mentioned in the domain mapping, i.e., there are no analogy skolems where we must postulate a new predicate or category of entity. If there is enough similar structure between the worked solutions, at least one encapsulated history will be created. If no encapsulated histories can be created due to an inability to find a satisfactory domain mapping, the system does not try to learn anything from this particular failure.

The system also extends a partially learned, or just initialized, domain theory with another analogy. The domain mapping becomes required correspondence constraints of a new analogy between the base and target domain theories themselves, ensuring that the overall domain theory is consistent. As before, any encapsulated history imported into the target becomes a conjecture about the new domain theory.

While powerful, analogies are not guaranteed to be sound. Consequently, we verify the newly proposed domain knowledge by trying again to solve the problem whose failure motivated the learning. If this problem is solved correctly, our system assumes that the new domain theory constructs are correct. Otherwise, it deletes both the new domain theory constructs and the domain mapping. Then, it tries one more time, considering the next best worked solution retrieved from memory.

Experiment

To examine how well this analogical learning method works, we need a baseline. Our baseline *spoon-fed* system consists of the same problem-solver, but with analogical learning turned off. When it receives a problem it cannot solve, it is given not just a worked solution, but whatever general encapsulated histories are needed to solve that specific target domain problem. In other words, it is given the correct knowledge, in its internal representations, ready for future use. This makes for a tough comparison, since our system in the analogy condition must figure out the encapsulated histories for itself.

a) Through how many turns does a centrifuge rotor make when accelerating from rest to 20,000 rpm in 5 min? Assume constant angular acceleration
b) A phonograph turntable reaches its rated speed of 33 rpm after making 2.5 revolutions, what is its angular acceleration?
c) Through how many turns does a centrifuge rotor make when accelerating from rest to 10,000 rpm in 270 Seconds? Assume constant angular acceleration
d) An automobile engine slows down from 3600 rpm to 1000 rpm in 5 seconds, how many radians does the engine turn in this time?
e) A centrifuge rotor is accelerated from rest to 20,000 rpm in 5 min, what is the averaged angular acceleration?

Figure 6: Test Problem Set

Method

Both systems begin with a linear kinematics domain theory, two worked solutions of linear kinematics problems, and hard-coded rules for problem-solving strategies and making modeling decisions. The systems are then tested on how quickly they can learn rotational kinematics problems. The testing materials are 5 problems, listed in Figure 6, and worked solutions. Learning curves were created by running 120 trials representing every possible ordering of the test materials. In each trial, after each problem, the system was given either the worked solution or encapsulated histories for that problem, depending on the condition. After each trial, the system's knowledge was reset.

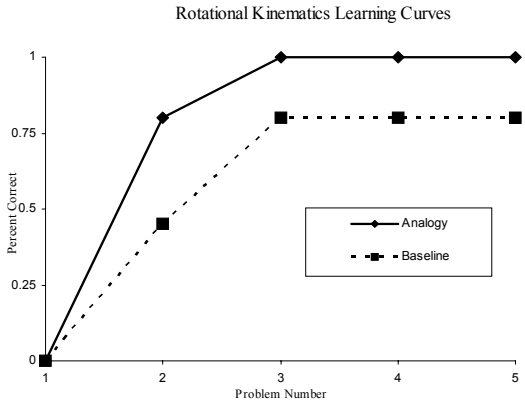


Figure 7: Experiment Results

Results

Figure 7 compares the learning curves for the analogy and baseline conditions. After studying just one worked solution, the analogy system was able to solve next problem correctly 80 percent of time. Furthermore, the analogy system has perfect performance after working on just two problems. The baseline system's ceiling was at 80 percent, and after one problem it was only able to get the next problem correct 45 percent of the time.

Further analysis of these results details the strength of the analogy approach. The baseline system failed to score above 80 percent of any of the conditions. The baseline system was unable to solve problem 'b' from Figure 6 regardless of what problems it has already seen, because none of the other problems use the same equation. The analogy system performed quite well, only in one situation

did the analogical domain transfer fail to learn the whole rotational kinematics domain after just one worked solution. This occurred when problem 'b' was the first problem. Problem 'b' makes no mention of a time quantity preventing a correspondence to be created for it. While a time quantity exists in both of these domains, it does not necessarily mean they should be aligned. The strength of the analogical approach is that transfer is guided by structural similarity. This is critical for broader application of this theory. For example, in linear and rotational dynamics, both domain theories have a mass quantity, but transfer is only possible when a domain mapping is made between mass, in linear dynamics, and moment of inertia, in rotational dynamics. (e.g. F=ma and $T=I\alpha$)

Related Work

As noted above, the closest work is Falkenhainer's PHINEAS (1988), which learned qualitative descriptions of processes based on analogies involving behaviors. PHINEAS used envisioning to verify its conjectures, whereas we use mathematical problem solving. Klenk & Forbus (2007) describe a system that learns by accumulating examples to solve AP physics problems within the same domain. Klenk *et al.* (2005) describe a system that learns causal models via analogies involving sketches annotated with causal knowledge. Both of these systems only learn within the same domain, and neither constructs general domain theories, unlike the system described here. Silver (1986) used explanation-based learning to acquire new mathematical skills, by contrast our system's mathematical knowledge is hard-wired.

In the QR community, de Kleer's work (1977) in reasoning on sliding motion problems demonstrated that qualitative reasoning was required for solving many quantitative physics problems. More recent AI work on transfer learning has recognized the importance of generating mappings between domains to allow for knowledge transfer, Liu and Stone (2006) use a version of SME to accelerate learning of state action policies in keep away soccer. Instead of using structure-mapping to accelerate learning, we use structure-mapping to learn new general domain concepts.

Discussion

We have shown that a domain theory for solving physics problems can be learned via cross-domain analogies. Our experiment shows furthermore that such analogical learning can be very efficient, when the two domains are sufficiently similar. The process of constructing domain mappings by exploiting similarities in worked solutions, and using that to import theories from one domain to another, is, we believe, a general and powerful process.

There are several directions we intend to pursue next. First, we have only tested this method with encapsulated histories, so we want to extend it to handle other types of domain knowledge. Based on experience in other analogical learning tasks, we believe that this will mainly involve figuring out the appropriate verification techniques. Second, we plan to integrate this algorithm into the Companion-based learning system of Klenk & Forbus (2007), so that we can combine both ways of analogical learning. We plan to explore a broader range of domain pairs, including domains which are quite distant, to explore better strategies for making use of weaker matches. We suspect that model-based diagnosis techniques could be used to debug analogically-derived domain theories, based on their success with diagnosing misconceptions in student models (de Koning *et al.* 2000).

We also expect that these techniques could be used more broadly in the QR community for accelerating the process of constructing domain theories. That is, given modeling environments designed to help domain experts create theories (cf. Bredeweg *et al.* 2006), there should be a growing library of domain theories to draw upon. An analogy-based assistant could help spot cross-domain connections, accelerating the process of constructing new domain theories.

Acknowledgements

This research was supported by the Cognitive Science Program of the Office of Naval Research.

References

Bredeweg, B., Salles, P., Bouwer, A., and Liem, J. 2006. Garp3 – A new Workbench for Qualitative Reasoning and Modelling. *Proceedings of the 20th International Workshop on Qualitative Reasoning*. Hanover, NH.

de Kleer, J. 1977. Multiple representations of knowledge in a mechanics problem solver. In *Proceedings of IJCAI-77*.

de Koning, K., Bredeweg, B., Breuker, J., and Wielinga, B. 2000. Model-Based Reasoning about Learner Behavior. *Artificial Intelligence*, 117(2).

Falkenhainer, B. 1988. Learning from Physical Analogies. Technical Report No. UIUCDCS-R-88-1479, University of Illinios at Urbana-Champaign. (Ph.D. Thesis)

Falkenhainer, B., Forbus, K. and Gentner, D. 1989. The Structure-Mapping Engine. *Artificial Intelligence.* 41.

Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence* 24.

Forbus, K. 2001. Exploring analogy in the large. In Gentner, D., Holyoak, K., & Kokinov, B. (Eds.) *Analogy: Perspectives from Cognitive Science.* MIT Press.

Forbus, K. & de Kleer, J. 1993. *Building Problem Solvers*, MIT Press.

Forbus, K., Gentner, D., & Law, K. MAC/FAC: A model of similarity-based retrieval. 1994. *Cognitive Science*, 19.

Forbus, K. & Oblinger, D. (1990). Making SME greedy and pragmatic. In *Proceedings of CogSci-1990*.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy, *Cognitive Science* 7(2).

Gentner, D. 2003. Why we're so smart. In Gentner, D. and Goldin-Meadow, S. (Eds.), *Language in mind: Advances in the study of language and thought.* pp. 195-235. MIT Press.

Gentner, D. and Gentner, D. R. 1983. Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner & A. Stevens (Eds.), *Mental Models*. Lawrence Erlbaum Associates.

Giancoli, D. 1991. Physics: Principles with Applications. 3rd Edition. Prentice Hall.

Hayes, P. 1978. The Naive Physics Manifesto. In D. Michie (ed), *Expert Systems in the Microelectronic Age*. Edinburgh University Press, Edinburgh, Scotland.

Klenk, M., Forbus, K., Tomai, E., Kim,H., and Kyckelhahn, B. 2005. Solving Everyday Physical Reasoning Problems by Analogy using Sketches *Proceedings of 20th National Conference on Artificial Intelligence (AAAI-05)*.

Klenk, M., and Forbus, K. 2007. Measuring the level of transfer learning by an AP Physics problem-solver. In *Proceedings of AAAI-07*. Vancouver, CA.

Olson, H. 1966. *Solutions of Engineering Problems by Dynamical Analogies*, D. Van Nostrand.

Shearer, J., Murphy, A., and Richardson, H. 1967. *Introduction to Systems Dynamics*. Addison-Wesley Publishing Company.

Shive, J. and Weber, R. 1982. *Similarities in Physics*. Adam Hilger Ltd.

Silver, B. 1986. *Meta-Level Inference: Representing and Learning Control Information in Artificial Intelligence*. Elsevier.

Collaborative Conceptual Modeling: Share, Search and Reuse

Jochem Liem, Anders Bouwer and Bert Bredeweg

Human Computer Studies Laboratory – University of Amsterdam
Kruislaan 419 (Matrix I), 1098VA Amsterdam, The Netherlands {jliem,bouwer,bredeweg}@science.uva.nl

Abstract

Within the Qualitative Reasoning community there is a desire to collaborate by integrating work and reusing parts of existing models. Although there has been much attention for the knowledge representation formalisms required for these tasks, activities performed by knowledge engineers such as copying model parts, searching for relevant models, and sharing intermediate modeling results are often not supported by existing modeling tools. This paper presents a set of new features in the Garp3 qualitative reasoning and modeling workbench to support these activities.

Introduction

An interesting idea of Qualitative Reasoning (QR) is to build a generic library of model fragments that can be applied by different users to simulate specific scenarios [4,7]. However, in practice the development of unified libraries has been limited. Modelers seem to prefer creating their own idiosyncratic libraries that are tailored to their specific needs, reusing only certain parts of previous modeling efforts, and adapting or leaving out other parts. In any case, whether a modeler wants to integrate modeling work of different modelers to create a unifying library, or reuse existing model fragments to create his own specific library, functionality is required to share and reuse parts of (partially) developed libraries. In this paper we present this functionality as implemented in the Garp3 workbench.

Taking a part of one body of knowledge and integrating it into another has several issues. For instance, knowledge parts usually relate to other knowledge parts. Without this other knowledge semantics may get lost. On the other hand, the existing body of knowledge may clash with the part of knowledge that is being reused. In order to prevent problems arising from such issues, we have defined a set of principles to support knowledge reuse. These are enumerated as: (1) *Syntactical correctness should be maintained*. Knowledge is usually represented in some formalism. After knowledge from one body of knowledge has been added to another, the augmented knowledge body should still adhere to the formalism. (2) *Knowledge should remain complete*. Knowledge parts often depend on other knowledge parts. When knowledge is reused in another body, the knowledge parts on which it depends should also be copied to that new context. (3) *No redundant knowledge should be added*. Two knowledge bodies may have overlapping parts. When reusing knowledge from one body in another, and some of the knowledge already exists, this knowledge should be reused as much as possible. (4) *Existing knowledge should not be altered*. The knowledge a modeler is working on can be assumed to be tailored to the needs of this modeler. Therefore, the knowledge should not be changed when knowledge from another knowledge body is added, as it could break the purpose for which the knowledge was developed. (5) *Semantics should be preserved as much as possible*. Copying knowledge should not cause the meaning of the knowledge to be changed or lost. (6) *Reuse solutions should be user-friendly*. This means the modeler should not have to provide too much additional input, be asked difficult questions, and that the functionality is easy to use.

Next to reuse functionality, two other conditions have to be fulfilled to efficiently reuse previously created bodies of knowledge. Firstly, modelers should be able to share their work within a community and the shared knowledge should be made searchable and accessible to the entire community. Otherwise, there is no knowledge to reuse. Secondly, modelers should be able to search through the shared knowledge in order to find knowledge that is potentially reusable for their needs. In the Garp3 workbench search is facilitated by meta-data, the model itself, and by high-level descriptions of the model and its expected behavior, referred to as Sketches.

Garp3 – QR Workbench

In this paper, the reuse functionality is addressed in the context of the Garp3 workbench (http://www.garp3.org), which implements a diagrammatic approach to modeling and simulating qualitative models [2]. Modeling in Garp3 starts by creating *model ingredient definitions*. These definitions include entities, agents, assumptions, configurations, quantities and quantity spaces. *Entities*, which represent the structural objects in a system, are organized in a sub-type hierarchy. They are defined by their name and their position in the hierarchy. Agents and assumptions are defined in the same way. *Agents* cause influences from outside of the system, while *assumptions* are labels that indicate that certain conditions are presumed to be true. *Configurations* are structural relations between entities that are defined by their name. *Quantities* represent the features of entities and agents that change during

simulation, and are defined by their name and a set of possible quantity spaces. *Quantity spaces* represent the possible values a magnitude (or derivative) of a quantity can have, and are defined by their name and an ordered set of possible values. Quantity spaces are associated to the quantities of entities or agents.

Next to the model ingredients defined by the modeler, there is also a set of predefined model ingredients. These include causal dependencies (proportionalities and influences), correspondences, the operator relations plus and minus, value assignments, and inequalities.

The model ingredient definitions described above can be used (instantiated) to create model fragments (MFs) and scenarios. MFs can be seen as composite ingredients that incorporate other ingredients as either conditions or consequences. They are organized in a subtype hierarchy, meaning that a child MF inherits the model ingredients of its parents. Furthermore, a MF can incorporate other MFs as conditional ingredients. MFs instantiated in another MF are called Imported Model Fragments (IMFs). An example MF incorporating another MF twice is shown in Figure 1.

Scenarios are also composite model ingredients. They describe specific system situations. During simulation, MFs are sought which match on the scenario (i.e. the model ingredients fulfill the conditions of the MF). The consequences of matching MFs are merged with the scenario to create an augmented state from which the next states of behavior can be determined.

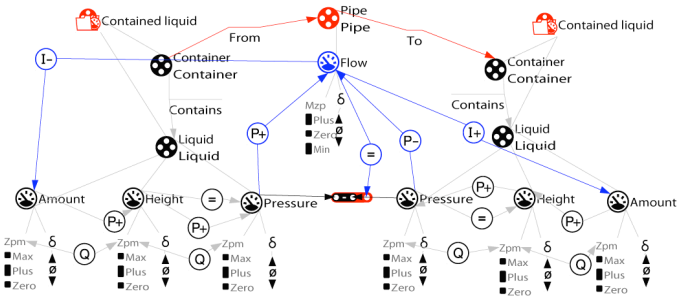


Figure 1: Liquid flow includes two Contained Liquid IMFs, the Pipe, and the configurations as conditions, and flow, its calculation and causal relations as consequences.

Reusing Parts of Models

A user-friendly way to support modelers with functionality to reuse model parts (model ingredient definitions, MFs and scenarios) is by allowing them to copy a model part in one model and paste it into another model (hereafter called *copy functionality*). From the modeler's perspective, copying model parts should be as easy as copying text between documents (principle 6).

The result of a copy should be assured to result in a syntactically correct model (principle 2). For example, ingredients must have unique names and arguments of relations must have correct type. This is achieved by rebuilding the copied model part in the target model as if the modeler had created it from scratch. Since Garp3 checks each of the modeler's actions, this assures that the model remains syntactically correct.

When a model part is copied to another model, no redundant information should be added to the model (principle 3). Therefore, when a model part and an already existing model part have the same name, the existing model part is reused if possible. The assumption is that if the copied model part and the existing model part have the same user-given name, they describe concepts in the same domain. If the existing model parts cannot be reused because semantics differ despite having the same name, the copied model part receives a suffix to indicate this.

Storing Copied Model Ingredients

Model parts often depend on other model parts without which they are incomplete. The completeness of the model part has to be maintained (principle 2) when a model part is copied. Our solution is to create a complete sub-model that contains the copied model part and all the model parts it requires. This sub-model is self-contained, meaning it can exist on its own, and is stored in a copy buffer (a model data structure). Details on how this works for each type of model part are explained in the next subsections.

Copying Model Ingredient Definitions

Entities are defined by their name and their position in the entity hierarchy. When a set of entities is copied to another model, they have to be integrated with the already existing entity hierarchy in some way. If an entity with the name already exists, redundancy should be avoided (principle 3). Therefore, the entity is not created, since it assumed to represent the same concept.

The entities should be integrated into the hierarchy in such a way that as much of the semantics is preserved as possible (principle 5). For entities this means that their position in the hierarchy should match as closely as possible. This is not straightforward since the modeler can select a subset of the entities in different branches. Only these selected entities should be copied, as it would not be user-friendly if the copy functionality would add more entities than the modeler has selected (principle 6). As a result, the final entity hierarchy will not always contain the parents of each entity.

Our solution to preserve as much of the semantics as possible is to also store all the ancestors of the selected entities in the copy buffer. When the selected entities are copied to the target hierarchy, the algorithm checks whether an ancestor (parent, grandparent, etc.) of each entity already exists. The entity is placed below the closest ancestor to recreate its semantics as closely as possible. If, no ancestor exists, the definition is placed below the root node. The modeler is allowed to choose a different position in the hierarchy where the copied entities should be recreated (principle 6). The hierarchy of entities is then created below this definition. Agents and assumptions are defined and copied in the same way.

Configurations are the simplest model ingredients, since they are only defined by their name. To avoid redundancy (principle 3), a copied configuration is only created if it does not exist yet in the target model.

Quantity spaces are defined by their name and their total order of values. The order of the values is important, since it defines to which values the magnitude (or derivative) of a quantity can change. Again, redundancy has to be avoided (principle 3). Therefore, a quantity space is not created if a quantity space with the same name and the same set of ordered values already exists. When a quantity space with the same name does not exist, it can be created normally. However, if a quantity space with the same name but with different values (or differently ordered values) exists, the values of the existing quantity space cannot be altered to match the values of the copied quantity space. The reason is that changing an existing quantity space would potentially alter the possible values of already existing quantities and cause simulations to generate different behavior (violating principle 4). Instead, a new definition is created with the suffix '(other values)'.

Quantities are defined by their name and a set of associated quantity spaces (of which only one can be chosen when it is added to a MF). To assure completeness (principle 2), the associated quantity spaces have to exist before the quantity can be created. Therefore, the associated quantity spaces of a quantity are created (as described above) before the quantity is copied. A quantity is copied normally if a quantity with the same name does not exist. A quantity is considered redundant and is not created if a quantity with the same name and the same quantity spaces already exists (principle 3).

When a quantity with the same name already exists, but has different associated quantity spaces, there are two options. The first option is creating a new quantity by adding the suffix '(different quantity spaces)' to its name. This potentially adds redundant knowledge (violating principle 3). The second option is to merge the sets of associated quantity spaces, which means existing knowledge is altered (violating principle 4). We choose this second option in our approach, since the associated quantity spaces only indicate the *possible* values for quantities. Therefore, adding quantity spaces to the set of possible quantity spaces does not really change the semantics of the quantity in model fragments (i.e. the simulation results remain the same), but only provides the possibility to use the quantity in a different way.

Copying Model Fragments

To copy a MF to another model, the algorithm has to deal with the subtype hierarchy, the MFs imported as conditional elements (see Figure 1), the model ingredient definitions of which instances are used in the MF, and the actual creation of the MF and its contents.

Dealing with imported and parent model fragments. In order to create a MF, all MFs it imports and its parent MFs have to exist. Each parent MF and reused MF has the same requirements. Therefore copying a set of MFs requires that their required MFs are collected and created first. A list of the to-be-created MFs and their required MFs is determined in several steps. Firstly, MFs inherited from parents are considered to be IMFs. Secondly, the IMFs within the MFs are gathered. Thirdly, the MFs corresponding to these IMFs are added to the list of required MFs. For each of the MFs added to the list the same three steps are performed until no more IMFs can be found. Finally, duplicates in the list of required MFs are removed, and the to-be-copied MFs are added to the list.

Determining and copying used model ingredient definitions. In addition to the model ingredients that the copied MFs use, also the model ingredients that the required MFs use have to be created. Given the MFs list created in the previous step, finding the required model ingredient definitions is easy. A list of model ingredient definitions is created for each model ingredient type. Then, by looping through the model ingredients of each MF, the definition of each model ingredient is added to the list of its type (if it is not already there). The end result is a set of lists that contain all model ingredient definitions needed to create the set of MFs.

The required definitions are copied as if the definitions were individually copied (as described in the 'Copying Model Ingredient Definitions' section). There is one difference when dealing with entities, agents and assumptions. During development it became apparent that the semantics about entities was lost when certain scenarios (see Section 'Garp3 – QR Workbench') are copied before MFs. Scenarios tend to use more specific concepts (lower in the hierarchy) since they represent specific situations, while MFs use general concepts (higher in the hierarchy) since they model general situations. Consider a model that defines the entities container, barrel (which is a type of container), liquid, and water (a type of liquid). Copying a scenario modeling a barrel with water would add both the barrel and the water concept to the hierarchy in the target model. Copying a MF that models a liquid in a container afterwards, would add the container and the liquid to two other branches in the hierarchy, since no information about the children of entities is stored in the copy buffer. The fact that barrel is a container, and that water is a liquid would be lost. This issue is solved by not only copying the required definitions, but also their ancestors. In the example, the liquid and container concepts are immediately created when the scenario is copied, preserving the semantics (principle 5).

Creating the model fragments and their contents. After these steps, all required model ingredient definitions are present, and each of the MFs in the list (of required and to-be-copied MFs) has to be created. The parents and the MFs each MF imports have to exist before that MF can be created. Therefore, the order in which the MFs are created is important. Instead of ordering the MFs, the algorithm loops through the list of MFs and checks whether the required MFs exist for each MF. If not, it skips to the next one. If they exist, the MF is created and the MF is removed from the list. This continues until the list is empty.

The creation of a MF also requires the creation of the contents of the MF. Again, the order in which the model ingredients are created is important, since Garp3 prevents model ingredients to be created that would result in a syntactically incorrect model (principle 1). Therefore, relations cannot be created if their arguments do not exist (e.g. a 'preys on' relation between two populations cannot exist without the two populations), quantities need their entities (e.g. a 'size' quantity of a 'population' cannot exist without the population), and value assignments need a quantity space before they can be created. To create the MF contents while maintaining a syntactically correct model at all times, the ingredients in the source MF are ordered. IMFs have to be created first, as model ingredients can be related to one of the model ingredients in the IMFs. Then, entities, assumptions, quantities, attributes, configurations, causal dependencies, value assignments, correspondences, plus or minus relations, and inequalities should be created. The algorithm loops through the sorted list of model ingredients creating each of the model ingredients and assigning it the same position on screen as in the source model.

Imported Model Fragments

When recreating the model ingredients of a MF in another model, it is complex and inefficient to have to determine for each model ingredient to which other model ingredient it is connected (e.g. to which entity a quantity should be added, or what the arguments of a relation are). The model ingredient(s) to which a model ingredient is related are called its *arguments*. To be able to determine the arguments of a to-be-created model ingredient in a target MF, a mapping has to be maintained between the ingredients in the source MF and the ingredients in the target MF. The model ingredients in the target MF to which the arguments in the source MF are mapped are the arguments of the to-be-created model ingredient.

Maintaining a mapping between the model ingredients in a source MF and a target MF is easy when model ingredients are created one at a time. When looping through the ordered list of model ingredients in the source MF to create the model ingredients in the target MF, the model ingredient in the list is mapped to the newly created ingredient. However, when a MF is imported, a set of model ingredients is added to the MF. Therefore, a mapping between the model ingredients of the IMF in the source model and the ingredients of the IMF in the target model is harder to establish.

To create the mapping, the model ingredients are sorted in the same way as when creating the contents of a MF. The algorithm loops through the sorted model ingredients of the IMF in the source MF, and selects one of the imported model ingredients of that type in the target model. It checks whether the ingredient has the same name, associated arguments and relations. If the checks succeed, the correct model ingredient is chosen and a mapping between the ingredient in the IMF in the source MF and the ingredient in the IMF in the target MF is saved. Creating this mapping for each of the model ingredients in the IMF always succeeds, since the IMFs are guaranteed to be identical in both models. The mapping is also used to update the positioning information for the model ingredients in the IMF in the target MF.

Reusing Existing Model Fragments

When a set of MFs is copied to another model, they might clash with MFs that already exist in the target model. These MFs cannot simply be reused, since the semantics of these MFs might be different. The existing MF might contain different model ingredients, or model ingredients might be differently connected than the model ingredients in the source MF. To avoid redundancy (principle 3), the existing MF should be reused if possible. A MF can only be reused if its ingredients are equal or a superset of the ingredients in the source MF, and if the corresponding model ingredients are connected in exactly the same way.

To determine if a MF can be reused, the mapping algorithms used to deal with IMFs is used. The source MF and the target MF are treated as IMFs, and the algorithm tries to create a mapping between the contents of the MFs. This mapping only succeeds if the target MF contains at least all the model ingredients that are in the source MF, and the model ingredients are connected in the same way. In contrast to the mapping between IMFs, the mapping can also fail, meaning that there is no mapping possible and that the MF cannot be reused. Then, the semantics of the to-be-created MF is different from the existing MF, and the MF is created with the suffix '(copy)'. When the MF is reused in other copied MFs, this new copy is used instead of the existing MF to preserve the semantics of the copied model fragment (principle 5). The existing MFs keep using the existing MF (preserving principle 4).

To preserve the semantics of a MF (principle 5), the reused MFs should be identical to those in the source model, as reusing different MFs results in a different complete MF. On the other hand, not reusing a MF which is a superset of the source MF (i.e. contains more ingredients), but which is otherwise equal requires a new, possibly redundant, MF to be created (violating principle 3). We feel that the best way to solve this issue is to ask the modeler for feedback. Although this is a difficult question, it makes the modeler aware that there are options, and each choice has a significantly different end result. This solution is more user-friendly than letting the algorithm make the choice for the modeler (principle 6).

Sharing and Searching for Models

To reuse models of others, modelers have to be able to share their work and access work of others. This is solved by allowing models to be uploaded to and downloaded from a central online model repository. However, the number of models in the repository can potentially become large, which means that modelers need to be supported by search functionality to find reusable models.

Typically, a modeler will want to search for models which contain a certain entity or quantity (e.g. a model which contains both an entity *population* and a quantity *size*). Normal search engines search for keywords in text and are unable to interpret the explicit knowledge representation in qualitative models. So the search engine is unable to distinguish between different types of model ingredients, or between domain specific and domain independent knowledge (i.e. the QR vocabulary and the knowledge formalized by the modeler). This hampers the search engine's ability to find relevant models. A search solution should make use of the explicit knowledge representation in qualitative models to allow modelers to focus their search using the QR vocabulary.

QR Models in the Web Ontology Language

The Semantic Web initiative proposes that "semantic search" becomes possible by making content machine-accessible [1]. The Web Ontology Language (OWL) is a description-logic based knowledge representation language, which is represented in RDF/XML, and is being developed as part of the Semantic Web initiative. It has become the de-facto standard for the sharing of knowledge on the web in the form of *ontologies*. By formalizing qualitative models as OWL ontologies, the models become interpretable by OWL search engines, and searching for models in which certain model ingredients or certain structures are used becomes possible. Additionally, the formalization of models in OWL opens up the possibility for other QR-tool developers to implement functionality to import these files. This could potentially make models accessible to communities using different QR tools.

There is no clear methodology for the creation of ontologies, therefore we have created our own. Firstly, the qualitative reasoning vocabulary was formalized as an ontology that consists of a hierarchy of all the model ingredients and their usage restrictions. Based on this domain-independent ontology, an OWL file-format for qualitative models was developed that refers to concepts defined in the vocabulary ontology. Using OWL reasoners, both the vocabulary and a set of model ontologies were checked for consistency, and the model ontologies were checked for correctness using the formalized usage restrictions. Functionality to export models to OWL and import them again was added to Garp3. The machine-accessible OWL-model files allow search engines to use the explicit knowledge representation of QR models.

Originally, we had the aim to use OWL reasoners to perform QR reasoning, but this proved to be impossible. Since the OWL reasoners are classification engines, the formalization should allow scenarios to be classified as being instances of MFs. However, due to limits in the expressiveness of OWL it is not possible to formalize MFs in a way that this reasoning can be performed [13]. In general, it is impossible to formalize general situations in OWL in a way that specific situations can be classified [11]. Due of this lack of expressiveness, the OWL representation of MFs needed to be adapted. However, this change has little effect on model search.

An earlier effort to support the interchange and reuse of MFs is the Compositional Modeling Language (CML) [5], which aimed to enable this functionality by defining CML in the Knowledge Interchange Format [10]. We have chosen to use OWL instead of CML, since it has a large user base and tools that are being actively developed.

Sharing and Searching in the Model Repository

A qualitative model repository[1] was implemented as a webpage that allows modelers to share their own models as OWL files, and search and download models of others. The main issue of implementing the repository is making it usable for modelers. The repository should be instantly usable for the user. Therefore, modelers should not be required to learn an OWL query language.

There are two different ways of implementing search functionality. The first is building an interface on top of an OWL query language, and the second is programming our own solution. Since building an interface on top of an OWL query language is complex, and implementing dedicated solutions has become easier due to the availability of semantic web libraries, we have chosen the second solution. The model repository is developed using the SWI-Prolog Semantic Web Library[2] and PHP[3].

The search functionality shows the model ingredient definitions of all the models. Selecting a definition reduces the list of matching models, allowing the modeler to iteratively refine the list of potentially useful models.

Sketch: Supporting Structured Modeling

The Garp3 workbench has been extended with the Sketch environment to allow modelers to create high-level representations of systems before starting the model implementation. The goal of the Sketch environment is threefold:

- to offer guidance during the modeling process, by providing editors that support different steps in the structured modeling methodology [3];
- to document initial ideas and intermediate modeling decisions by allowing the creation of external representations for them. Although not all captured ideas may end up in the final model, these Sketch representations can aid in communicating about the domain and establishing consensus between collaborating modellers;

[1] http://hcs.science.uva.nl/QRM/models/repository/
[2] http://www.swi-prolog.org/packages/semweb.html
[3] http://www.php.net

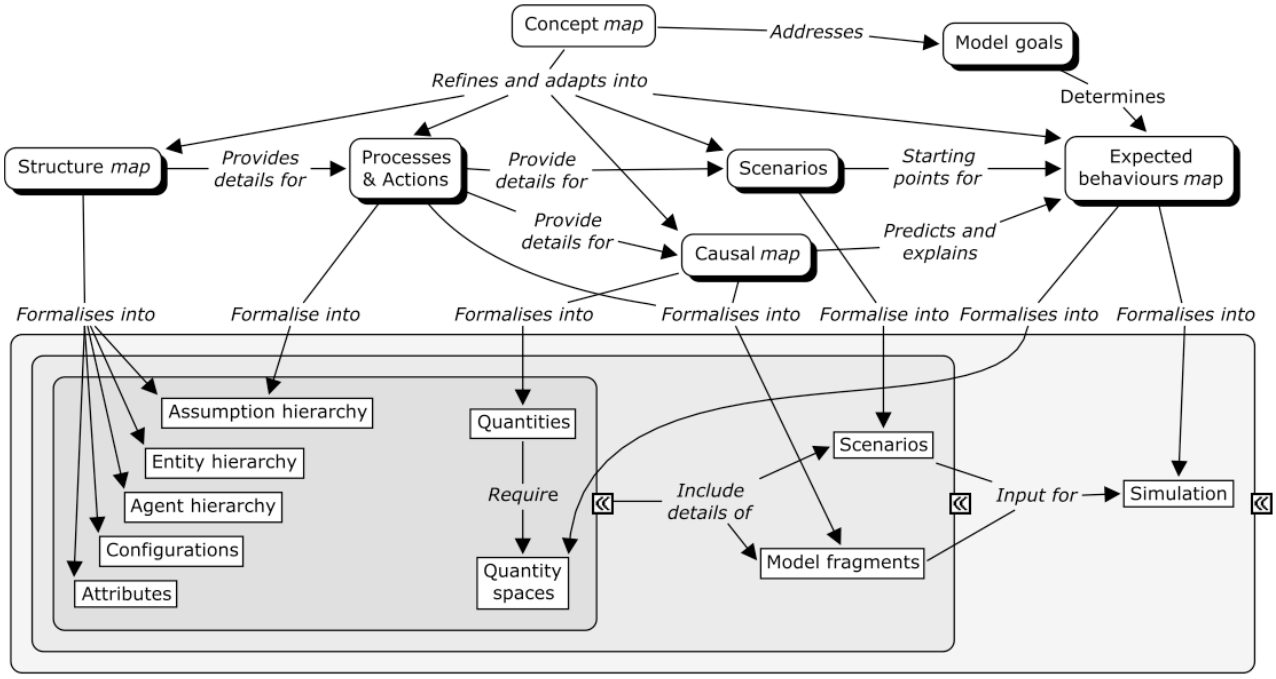


Figure 2: Overview of the intermediate representations used in the structured approach.

- to facilitate determining whether an existing model is relevant for a modeler, by providing a set of Sketch representations as a high-level abstraction and introduction to the model. Together with the metadata that was already introduced in Garp3 [2] (including abstract, keywords, and descriptions of the model goals, domain, and intended audience), this makes it possible to find out what the model is about, without having to analyze the details of the model implementation, which might be hard to understand at first glance.

Compared to the Build environment interface of Garp3 as described in [2], the editors in the Sketch environment have been designed to have a sparser user-interface. Each editor focuses on a specific kind of knowledge, so that the modeler has to focus on only a few types of ingredients per editor. Furthermore, the Sketch editors do not impose some of the grammatical constraints associated with the model implementation (e.g., quantities do not have to be associated to an entity, and quantity space values do not have to be characterized as points or intervals), to facilitate the flow of ideas in the initial stages of modeling. Not enforcing these constraints does not create a problem in the Sketch environment because the Sketches are not used directly as input for the simulation engine.

The Sketch Editors

The Sketch[4] environment consists of seven different editors. Their recommended use is in the order matching Figure 2, which shows an overview of the intermediate modeling results and how they follow up on and refine each other.

In the Concept Map editor, inspired by the IHMC Cmap Tools [14], a modeler specifies the concepts and relationships that are considered important in the domain as a graph consisting of labeled nodes and links, respectively. No additional building blocks or constraints are given at this stage (such as having to create modeling ingredients in a particular order), allowing the modeler to freely specify his or her initial ideas. The concept map addresses the model goals and serves as a basis for refinement into the other Sketches.

In the Structural Model editor, the modeler needs to focus on the physical structure of the system and how it relates to the environment. The graphical format is similar to the Concept Map editor, but here each node is assigned a type (entity, agent, assumption, or undefined concept). This guides the modeler to be more specific about the nature of what is represented. Common structural relationships have been predefined (connected-to, contains, is-a), but the modeler can add new relation definitions as well.

The generic knowledge about system behavior can be represented in three editors: the Process Definitions editor, the Actions and External Influences Definitions editor, and the Causal Model editor. The Process Definitions editor allows the modeler to define processes that affect the system by specifying the related entities, quantities, start conditions, effects, stop conditions, and behavioural assumptions. The Actions and External Influences Definitions editor is used to specify influences exerted from outside the system, and is similar to the Process Definitions editor except for an additional field for the agents causing the influence. The Causal Model editor is used to describe the causal dependencies between quantities, to indicate how they affect each other. This type of editor relates to tools such as VModel [9] and Betty's Brain [12]. In the Sketch Causal Model editor there are

[4] The term 'Sketch' is used here to refer to the preliminary and relatively unconstrained nature of the representations, rather than free-form drawing

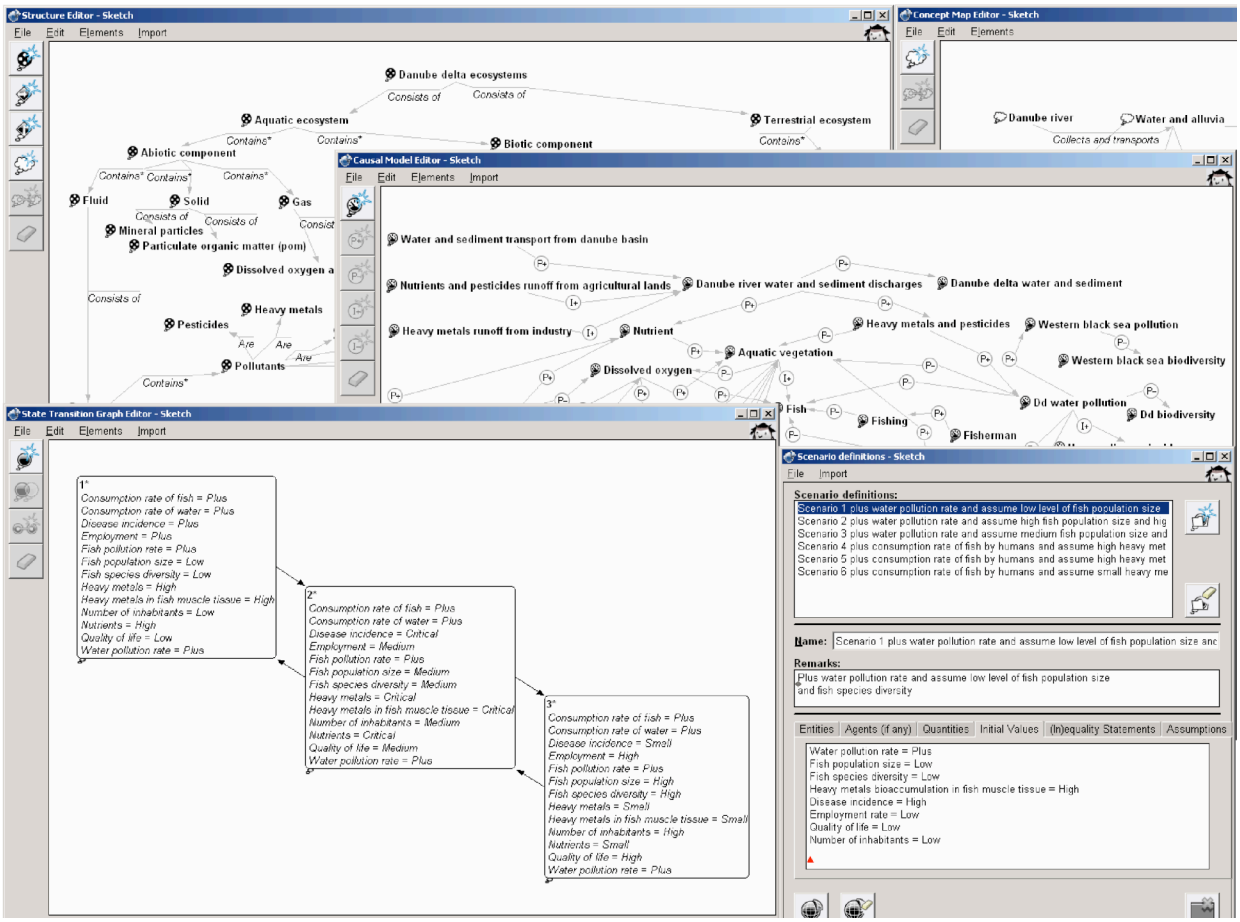


Figure 3: A screenshot of the Sketch environment.

four types of causal relationships: they are either direct or indirect, and either positive or negative [8]. Together, they provide an overview of the effects of the processes and actions defined in the previous two editors, and how these effects propagate through the system.

Finally, there are two editors that deal with specific behavior: the Scenario Definitions editor, and the Behavior Graph (or Expected Behaviors Map) editor. In the Scenario Definitions editor, scenarios can be specified to represent different initial situations of the system, which will be the starting points in the system's behavior. In this structured text based editor scenarios can be defined by specifying the entities, agents, quantities, initial values, (in)equality statements, and behavioral assumptions that pinpoint what is relevant in determining the behavior of the system.

In the Behavior Graph editor, the modeler can indicate how quantities and (in)equalities are expected to change over time given an initial scenario. The modeler creates the states, defined by a set of value and (in)equality statements, and possible transitions between them to represent the main aspects of the system's anticipated behavior. The value and (in)equality statements are displayed within the state nodes, to present a clear overview of the content of the possible behaviors. A screenshot of several of the Sketch editors is shown in Figure 3. The contents of the figure are taken from a case study within the NaturNet-Redime project about the Danube Delta Biosphere Reserve [6].

To further support working through the structured modeling methodology (following Figure 2), it is possible to import certain parts from one Sketch into another, thereby enabling reuse and refinement of ideas. For example, concepts specified in the Concept Map editor can be imported (and refined into other types) in the other editors. Entities, agents, and assumptions specified in the Structural Model editor can be imported in the Process Definitions editor, the Actions and External Influences Definitions editor, and the Scenario Definitions editor.

Conclusions and Future Work

This paper presents new collaborative modeling features of the Garp3 qualitative reasoning and modeling workbench [2] to further facilitate the articulation of knowledge. Engineers of conceptual knowledge use Garp3 to construct qualitative models. Particularly, partners in the NaturNet-Redime project use the workbench to capture knowledge about river restoration ecology.

To prevent redoing of work within a community a central online model repository has been developed in which qualitative models (formalized in the Web Ontology Language) can be shared and searched for. Within Garp3 multiple model support and copy functionality have been

added so that model parts can be easily reused. This makes it possible to reuse parts of existing models, integrate models to create larger models, and create alternative representations of systems to share within communities.

To support synchronous collaborative modeling the Sketch environment has been developed. Sketch helps consensus building through explicit representations to focus discussions and solidify established consensus. Another role of the Sketch environment is to ease the transition from initial ideas to implementation of the model, following a structured approach to model building [3]. Because the Sketches provide a high-level description of the implemented model, inspecting the Sketches can also help modelers to determine if a particular model is useful for them, without having to inspect the details of the model implementation itself. This is another added value.

Future work will focus on three issues. First, using the Sketch representations to (partially) automate model construction. Because the representations used in the Sketch environment are less constrained than the definitive Garp3 format for model implementation, certain model ingredients from the Sketches (e.g., the structural model, the causal model, the processes, and scenarios) might be reused or refined into the final model. The State-Transition Graph Sketch that represents anticipated behaviors can be compared to the actual simulation results to find discrepancies that may be used to refine the model, or the expectations.

Second, reusing model parts can cause undesired behavior during simulation. Investigating what kinds of issues occur, and how results deviate from modeler's expectations will further the design of repair methods, and eventually, automated support for troubleshooting.

Third, studies with modelers are planned, in the context of the NaturNet-Redime project, to evaluate the new functionality.

Acknowledgements

The research presented here is co-funded by the European Commission within the 6th Framework Programme for Research and Development (2002-2006) (project NaturNet-Redime, number 004074, www.naturnet.org). We thank the reviewers and the participants of the NaturNet-Redime workshops in Sofia, Bulgaria (March 2006) and Birini, Latvia (September 2006) for their insightful feedback.

References

[1] Antoniou, G. and van Harmelen, F. A Semantic Web Primer, The MIT Press, Cambridge, Massachusetts, April 2004.

[2] Bredeweg, B., Bouwer, A., Jellema, J., Bertels, D., Linnebank, F. and Liem, J. Garp3 - A new Workbench for Qualitative Reasoning and Modelling. 20th International Workshop on Qualitative Reasoning (QR-06), C. Bailey-Kellogg and B. Kuipers (eds), pages 21-28, Hanover, New Hampshire, USA, 10-12 July, 2006.

[3] Bredeweg, B., Salles, P., Bouwer, A., Liem, J., Nuttle, T., Cioaca, E., Nakova, E., Noble, R., Caldas, A.L.R., Uzunov, Y., Varadinova, E. and Zitek, A. Towards a Structured Approach to Building Qualitative Reasoning Models and Simulations. *Ecological Informatics* (in press).

[4] Bredeweg, B. and Struss, P. (eds). 2003. Current Topics in Qualitative Reasoning. *AI Magazine (special issue)*, Volume 24, Number 4 (winter), pages 13-130.

[5] Bobrow, D., Falkenhainer, B., Farquhar, A., Fikes, R., Forbus, K., Gruber, T., Iwasaki, Y., and Kuipers, B. (1996). A Compositional Modeling Language. In Iwasaki, Y., and Farquhar, A., editors, Proceedings of the Tenth International Workshop for Qualitative Reasoning(QR-96), pages 12-21, AAAI Press, AAAI Technical Report WS-96-01, Menlo Park, California, USA.

[6] Cioaca, E., S. Covaliov, C. David, M. Tudor, L. Torok, O. Ibram, Textual description of the Danube Delta Biosphere Reserve case study focusing on basic biological, physical, and chemical processes related to the environment. Naturnet-Redime, STREP project co-funded by the European Commission within the Sixth Framework Programme (2002-2006), Project no. 004074, Project Deliverable Milestone M6.2.1.

[7] Falkenhainer, B. C. and K. D. Forbus. Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 51:95–143, 1991.

[8] Forbus, Kenneth D., Qualitative Process Theory, Artificial Intelligence, 24, p. 86-168, 1984 .

[9] Forbus, K., L. Ureel, Carney, K., and Sherin, B.. Qualitative modeling for middle-school students. In J. de Kleer and K. D. Forbus (eds.) Proceedings of QR 2004, 18th International Workshop on Qualitative Reasoning, Evanston, USA, August 2–4, 2004, pp. 81–88, 2004.

[10] Genesereth, M. R. and Fikes, R. E. (1992). Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Stanford University Logic Group.

[11] Hoekstra, R., Liem, J., Bredeweg, B. and Breuker, J. Requirements for Representing Situations. Proceedings of the OWLED'06 workshop on OWL: Experiences and Directions. In Bernardo Cuenca Grau and Pascal Hitzler and Conor Shankey and Evan Wallace (Eds.): *CEUR Workshop Proceedings 216*. Athens, Georgia, USA, November 10-11 2006.

[12] Leelawong, K., Y. Wang, G. Biswas, N. Vye, J. Bransford, and D. Schwartz. Qualitative reasoning techniques to support learning by teaching. In G. Biswas, (ed.), Proceedings of QR 2001, 15th International Workshop on Qualitative Reasoning, St. Mary's University, San Antonio, Texas, 17-18 May 2001, pp. 65–72, Stoughton, WI, 2001. The Printing House.

[13] Liem, J. and Bredeweg, B. OWL and qualitatative reasoning models. In C. Freksa, M. Kohlhase, and K. Schill (Eds.): *KI 2006, Lecture Notes in Artificial Intelligence 4314*, pp. 33-48. Springer-Verlag Berlin Heidelberg 2007. (to appear)

[14] Novak, J. D. and D. B. Gowin. Learning How to Learn. Cambridge University Press, New York, 1984.

Using Qualitative Representations and Analogical Mapping to Solve Problems from a Spatial Intelligence Test

Andrew Lovett Kenneth Forbus Jeffrey Usher

{andrew-lovett, forbus, usher}@northwestern.edu
Qualitative Reasoning Group, Northwestern University
2133 Sheridan Road, Evanston, IL 60208 USA

Abstract

We show how qualitative spatial representations can be used with the Structure-Mapping Engine (SME), a domain-general model of analogy, to solve a set of problems from the Raven's Progressive Matrices test. SME is used in a two-stage mapping process which we have previously shown to be effective for solving geometric analogy problems. Each problem is drawn in PowerPoint and input into sKEA, our sketch understanding system. sKEA automatically computes qualitative representations of the drawings, using a spatial representation scheme motivated by research on human perception. We demonstrate that the representations generated by sKEA can be used with SME to solve the Raven's Progressive Matrices problems, without using any processes specifically designed for the task.

1. Introduction

The processes of visual perception and spatial reasoning present a number of interesting challenges to the fields of Cognitive Science and Artificial Intelligence. The principle questions are, how do we as humans construct appropriate representations of our visual world and use these representations to solve problems, and how can computers be programmed to match this performance. We have argued previously that we believe one key to spatial reasoning is using qualitative representations (Forbus, Ferguson, & Usher 2001). Qualitative representations capture an appropriate level of abstraction for finding salient similarities and differences across visual stimuli while ignoring irrelevant dimensions, such as the exact length or orientation of a particular edge. We have built a sketch understanding system, sKEA (Forbus et al. 2004), which constructs qualitative spatial representations of sketches drawn by a user. These representations can be used as the input to spatial reasoning tasks.

Our goal is to match human performance on spatial reasoning tasks, in order to show that qualitative spatial representations are both sufficient for solving the tasks and sufficient for explaining human performance on the tasks. For example, in (Tomai et al. 2005), we used sKEA representations to solve Miller geometric analogy problems, of a type used on intelligence tests. These are problems of the form A : B :: C : ?, or "A is to B as C is to _?" We solved these problems using SME (Falkenhainer, Forbus, & Gentner 1989), a computational model of analogy based on Gentner's (1983) structure-mapping theory of analogy in humans. We demonstrated that our domain-general model of analogy, when used with the spatial information provided by sKEA, was sufficient for solving a set of 20 analogy problems used in Evans' (1968) classic work.

While an important step, the problems used by Evans are not calibrated in detail against human performance at different developmental levels. By contrast, the Raven's Progressive Matrices (RPM) has been heavily studied and used in evaluations in recent years. RPM is a nonverbal intelligence test which measures individuals' eductive ability, i.e., their ability to find patterns in the apparent chaos of a set of visual scenes (Raven et al. 1998). The matrices come in two forms: 2x2 and 3x3 (see Figure 1; the authors have made up these examples to protect the security of the test). In each problem, individuals are presented with a matrix in which the bottom right entry has been left blank. Test-takers must pick the image that correctly finishes the matrix from a set of 6 possible answers for the 2x2 matrices, or a set of 8 possible answers for the 3x3 matrices.

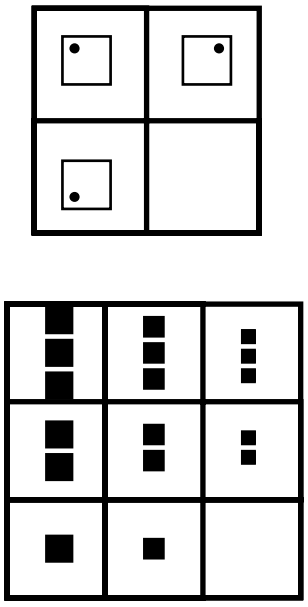


Figure 1: Examples of a 2x2 and 3x3 matrix problem.

We are particularly interested in the Standard Progressive Matrices (SPM) (Raven et al. 2000), which is divided into five sections, each progressively harder than the last. The A and B sections each contain 12 2x2 matrices, while the C, D, and E sections each contain 12 3x3 matrices. Section A involves simply filling in the missing part of an image. Later sections require more abstract reasoning.

Carpenter et al. (1990) modeled the performance of college students on the Advanced Progressive Matrices. Their system was able to match the performance of the most advanced students on these problems. However, their focus was on how students organized knowledge and applied problem-specific strategies. Among the shortcomings of their study, they mentioned: (1) they hand-coded the stimuli, rather than using an automatic perception system; (2) they identified the necessary rules beforehand and hand-coded them into the system, rather than having it discover the rules; and (3) their model might not capture the techniques and strategies used by younger students.

Our model addresses all three of these shortcomings. (1) We use sKEA to automatically compute visual representations. (2) We use only general processes to solve the Raven's problems; our system does not possess any prior knowledge about particular patterns or rules. (3) We are modeling performance by less advanced individuals; specifically, we want to see if domain-general processes and task-general spatial representations suffice for modeling performance on sections B and C, two relatively easy sections in the SPM.

We start by briefly reviewing SME and sKEA. The spatial representations automatically constructed by sKEA are discussed next, including some specifically motivated by the RPM task. We then describe how SME is used in a two-stage mapping process to solve these problems. Finally, we describe the model's performance and summarize some related work.

2. The Structure Mapping Engine

Structure-mapping defines analogy and similarity in terms of a comparison process operating over structured representations, i.e., entities, attributes, and relations, including higher-order relations between relations (Gentner 1983). A key bias affecting this process is *systematicity*, i.e., mappings involving systems of relations, including higher-order relations, are preferred by people. SME implements this comparison process (Falkenhainer, *et al*. 1989). Given base and target descriptions, SME computes one or more *mappings*. A mapping consists of *correspondences* that describe how items (entities, statements) in the base description align with items in the target, *candidate inferences* representing conjectures suggested by the correspondences, and a *structural evaluation score* indicting the structural quality of the match. Candidate inferences are based on non-aligned structure in the base or target which is rooted in the correspondences of the mapping.

3. Sketch Understanding

sKEA, the *sketching Knowledge Entry Associate*, is the first open-domain sketch understanding system (Forbus et al. 2004). Most sketch understanding systems are limited to a narrow domain and require extensive per-user training. sKEA works differently, by splitting the perceptual load with the user. Users manually segment a sketch into *glyphs*, pieces of digital ink that represent the different objects in the sketch. Users can label each glyph with one or more conceptual categories from sKEA's knowledge base. sKEA automatically computes various qualitative spatial relations between the glyphs in the sketch, including relative position & size, and the RCC8 set of topological relations (Cohn 1996). Sketches can be segmented into *layers*. Relations are only automatically computed between glyphs on the same layer. sKEA combines the conceptual knowledge provided by the user with the automatically computed spatial knowledge to produce a qualitative, structural representation of a sketch.

In cases where complex, precise spatial arrangements are required, drawing by hand can be too difficult. Moreover, certain types of figures simply cannot be sketched. For example, consider the solid black squares in Figure 1. While the four sides of a square can be sketched, there is no good way to sketch a solidly black object. Consequently, we have added a second method for adding digital ink to sKEA. Users can copy lines and polygons drawn in Microsoft applications such as PowerPoint and paste them into sKEA. sKEA interprets the Windows Metafile format, which consists of a set of draw commands, to create either lines or polygons in sKEA's digital ink format.

4. Spatial Representation

One of our primarily goals is to develop a qualitative spatial representation scheme that is powerful and flexible enough to match human performance in a variety of spatial reasoning tasks, provided it is paired with an appropriate spatial reasoning system. We see each task as an opportunity to evaluate our current representation scheme and, when necessary, add additional terms to our spatial vocabulary to meet the needs of the task. In this section, we summarize the representation scheme used to solve the RPM problems. While most components had been used for previous tasks, we were required to add some new features to the representation scheme, most notably a texture component. In the future, we hope to show that these new features, rather than being solely required for this task, will play useful roles in performing other tasks.

Because we are interested in explaining human performance, our spatial representation scheme is not simply an arbitrary collection of terms which have proven useful in reasoning tasks. Rather, the components are motivated by theories of human perception. Psychological evidence supporting the components below is described when available.

Basic Elements

What objects should make up the basic elements in a representation of visual structure? One possibility would be to create an entity for every edge in an image. However, Treisman and Patterson (1984) found evidence that humans detect closure at an early, pre-attentive stage in the visual pipeline, allowing them to quickly reason about triangles at the attentive level without, apparently, having to resort to

analyzing the triangles' individual edges. Consequently, we use closed shapes as entities in our spatial representation. Edges that do not form a closed shape are treated as separate entities. For example, a right triangle is represented as a single entity, while two edges forming a right angle are represented as two distinct entities.

We also include two basic attributes of closed shapes: their fill color, for solidly colored shapes, and their outline color. Ordinarily, every shape will have an outline color; a solidly black square will also have the outline color **black**.

Spatial Relations

Our representation contains two basic positional relations between elements: **left-of** and **above**. sKEA computes these relationships under certain conditions for pairs of glyphs that are disconnected and adjacent, i.e., there is no third glyph between them. Adjacency is determined by using a Voronoi diagram (Forbus et al. 2003). Positional relations are only asserted when one glyph is directly above or directly beside an adjacent glyph; mixtures of positional relations between a pair of glyphs are not allowed.

When one object is spatially located within another, a different set of relationships must be applied. In studying participants' memories for visual scenes in which a dot was located somewhere inside a circle, Huttenlocher et al. (1991) found that the memory contained two components. One was a quantitative component consisting of the dot's actual location inside the circle. The second was a qualitative component which encoded which of the circle's four quadrants the dot had been located in. When asked to recreate the scenes, participants tended to allow their memory of the qualitative component to bias their memory of the exact location, moving the dot closer to the center of the quadrant in which it had been located. We interpret Huttenlocher *et al*.'s results (1991) as indicating that closed shapes are capable of setting up a frame of reference. This frame consists of x- and y-axes running through the center of the shape. For simplicity, we currently assume that these axes align with the axes of the global reference frame, although this is not always true. When one visual object is located inside a closed shape, we first assert a topological relation stating that one glyph is inside the other. Then, the interior glyph's position relative to the axes of the exterior glyph's frame of reference is computed and encoded.

Shape Comparison

Clearly people are capable of distinguishing between different shapes based on detailed properties even when they are the same general type (e.g., two triangles). So while we treat closed shapes as entities, properties of their edges are still used in comparisons. People can compare novel complex shapes that are presented at different orientations. For tasks like the RPM, detecting what differences (rotation, reflection, and/or scale change) there are between two instances of the same shape is crucial.

Previous work on mental rotation provides valuable constraints. Shepard and Metzler (1971) demonstrated that when participants were shown drawings of arbitrary shapes and asked to determine whether one shape was a rotation of the other, the time required to identify the shapes as identical was proportional to the angle between them. This result seems to suggest that shape representations are orientation-specific, and that participants cannot compare two shapes to determine whether they are the same without first rotating their representation of one of the shapes to align it with the other. However, in the many studies that followed the original (see Shepard and Cooper 1982, for an overview), one fairly consistent finding was that, whether the shapes being compared were 2D or 3D, the time required tended to be proportionate to the degrees of rotation along the shortest possible axis of rotation. How could participants know the shortest axis of rotation before they knew whether the shapes were the same? Shepard and Cooper (1982) suggested that, before mental rotating one of the representations, people identify corresponding parts on the two images and these corresponding parts guide the rotation. If this is true, then people must have access to some orientation-invariant representation that can be used to identify corresponding parts.

Thus, we propose that people use two representations for objects being compared, whether the objects are simple closed shapes or complex, three-dimensional structures. The first representation is a qualitative, rotation-invariant representation that describes how an object's parts relate to each other. If the object is a closed, 2D shape, this representation describes the shape's edges. It includes relations which specify the types of angles that exist between connected edges, as well as the relative lengths and orientations of the edges.

When comparing two shapes, we claim that people begin by using a structure-mapping process to align the shapes' qualitative representations, identifying the corresponding edges in each shape. Once corresponding edges have been identified, people cannot immediately conclude that the shapes are the same. The qualitative representations are relatively sparse, lacking specific information about the length and orientation of each edge. To determine that the shapes are identical, their quantitative representations must be compared. These representations are orientation-specific, so they cannot be compared without first mental rotating one of them to line up its edges with the corresponding edges in the other representation. Because corresponding edges are known, people can quickly compare one pair of corresponding edges to determine the shortest axis of rotation between them. They must then rotate all the other edges together along this axis, using some mental process that can take linear time.

Implementation Our shape comparison process starts by decomposing the shape into its component edges by identifying significant discontinuities in the curvature of its outline. We then build a qualitative representation where each component edge becomes an entity. We compare the qualitative representations using SME. The correspondences

for the mappings it finds are used to compute the quantitative difference. That is, we iterate over every pair of corresponding edges to ascertain whether the rotational difference between every pair is the same. This comparison process is an approximation of the mental rotation process described above, since we compare each pair of edges in isolation, rather than rotating the entire set of edges together over a common axis. When SME returns multiple mappings (representing multiple possible rotations between the shapes), we pick the shortest possible rotation.

Our comparison process can detect reflections and scale changes as well as rotations. We detect reflections over the x- or y-axis by reversing the order of edges in one of the representations, finding the corresponding edges, and checking whether a reflection over the appropriate axis would explain the orientation for every pair of corresponding edges. We also check whether one shape is longer or taller than the other along the x and y axes. This size comparison is facilitated by the previous orientation comparison because, if one shape has been rotated about 90 or 270 degrees, we know to switch that shape's x and y dimensions before comparing their sizes.

Encoding The choice of when to compare shapes and how to encode the results of that comparison is somewhat task-specific. Given an RPM problem, our system begins by comparing shapes across all figures in the problem, creating a shape equivalency class for each shape. Membership in this equivalence class is used as an attribute that is encoded for every member of that class. For example, all rectangles in a sketched problem would be placed in the same shape class. (The attribute used is arbitrary, having no meaning outside that problem.) This enables objects to be aligned based on having similar shapes.

Transformations (i.e., rotations, reflections, and scale changes) are computed when images are compared. In RPM, this includes comparing two entries in the matrix or an entry with a possible answer. When comparing two images, the system begins by iterating over all shape equivalency classes. For each class, it selects the first instance of that class it finds and uses it as a reference shape. It compares all other instances of the class in both images to the reference shape to identify any transformations. All transformations are included in the representation of the appropriate image as an attribute of that object. This process is performed on edges which are stand-alone entities, as well as closed shapes. Firstly, all such edges are assigned to one of the following shape classes: **CurvedEdge** and **StraightEdge**. Then, one straight edge is chosen as a reference, and all other straight edges are compared to its orientation and length.

Textures

Several RPM problems require distinguishing textures, so we implemented a rudimentary representation of textures in our system. This is the least psychologically constrained component of our model. Textures are detected by looking for parallel lines that are not part of a closed shape. When enough such lines are found, they are grouped together to form a *texture patch*. The outline of a texture patch is the total area covered by the set of parallel lines. This outline is scrutinized to see if there is a closed shape whose outline matches it. If such a shape is found, it is added to the texture patch to create a border for it; otherwise the patch is marked as a closed shape without a border. Texture patches can play the same roles as any other closed shape in the spatial relationships described above.

Encoding In RPM problems, there are cases where two shapes should align simply because they both possess a texture, and other cases where two shapes should align specifically because they possess the same texture, i.e., the lines that make up their textures are parallel. We captured these distinctions by including two attributes for any closed shape possessing a texture. The first is the **TexturedObject** attribute, assigned to all shapes with a texture. The second attribute is a texture class attribute, similar to the shape class attribute described above. That is, for a given RPM problem, all shapes possessing textures with parallel lines are placed into a texture equivalency class and assigned an arbitrary texture name for that class. Thus, in a comparison between images, any two textured shapes will share at least one common attribute, but two shapes with the same texture will share two common attributes.

5. The Two-Stage Mapping Process

We use a variation of the two-stage mapping process from (Tomai et al. 2005) to solve RPM problems. We first describe the process for 2x2 problems and then show how it is generalized to solve the more complex 3x3 problems.

Solving 2x2 Matrices

Recall that geometric analogy problems take the form A : B :: C : ?. This form can easily be applied to a 2x2 Raven's matrix by focusing on either the rows or columns. For example, consider the columns in the 2x2 matrix in Figure 1. We could solve the problem posed by this matrix by asking "The top-left entry is to the bottom-left entry as the top-right entry is to _?". By posing the question in this way, we are ignoring some of the information provided in the matrix, specifically the relationship between the two entries in the top row. However, one of our key insights about both the 2x2 matrix problems and the 3x3 matrix problems is that they include a great deal of redundant information. Much of the time, these problems can be solved while ignoring some, or even most of the information provided in the matrices.

Our strategy is to run two stages of comparisons (see Figure 2). The first stage compares individual entries in the matrix. SME is used to compare the top-left and bottom-left entries in the matrix. Its mapping contains candidate inferences for expressions in the base that fail to align with the target and reverse candidate inferences for expressions

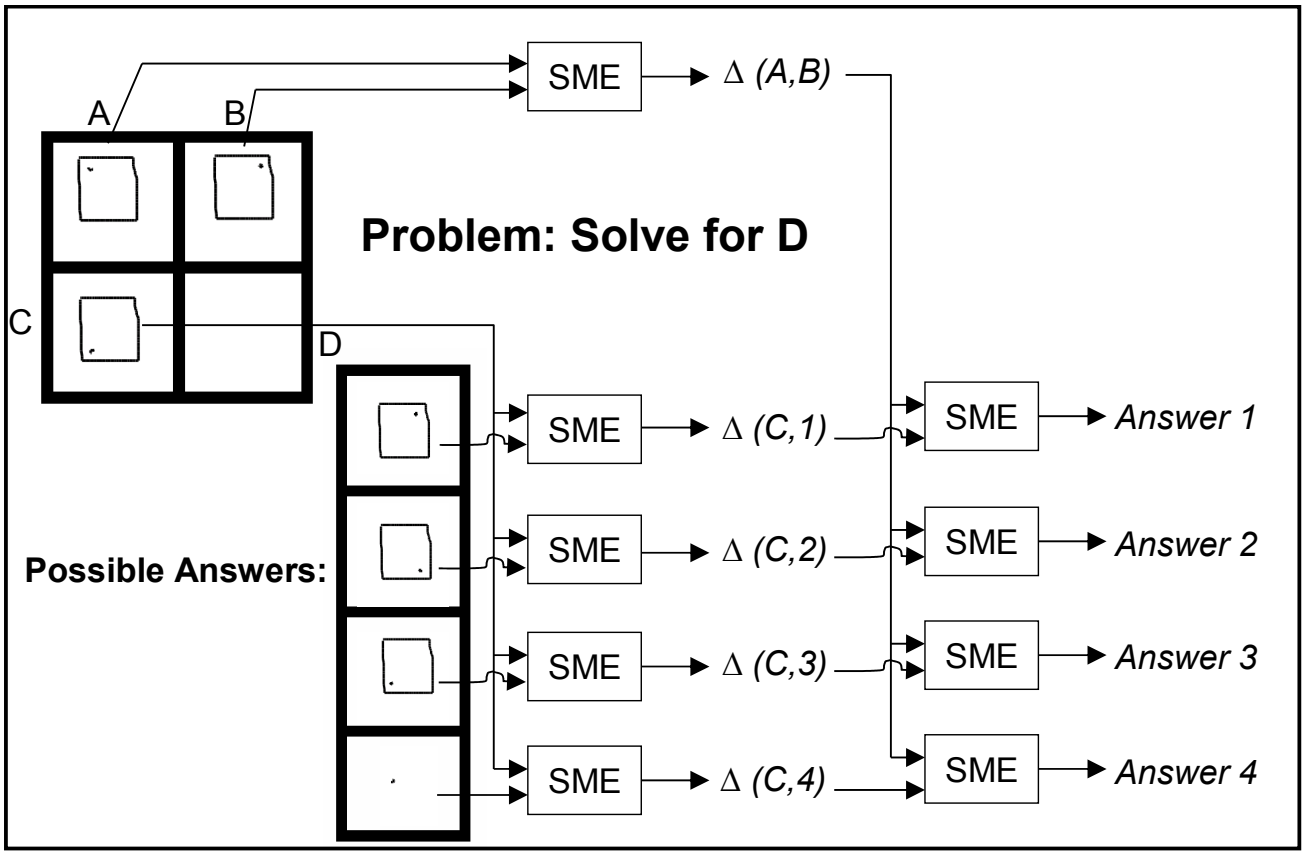


Figure 2: The Two-Stage Structure-Mapping process, using rows to solve a problem

in the target that fail to align with the base. In the current example, it produces one candidate inference saying that the dot is in the upper half of the square and one reverse candidate inference saying that the dot is in the lower half of the square. These inferences are used to construct a new representation, a representation of the differences between the two entries in the sketch. We refer to sets of differences as $\Delta(e_1,e_2)$, where e_1, e_2 are matrix entries or possible answers. Similarly, we use SME to compare the upper-right entry in the matrix to each of the six possible answers. Each of these comparisons also produces a Δ.

The second stage mapping process uses SME to compare the Δs found in the first stage. Thus Δ(*upper-left*, *lower-left*) is compared to the Δ(*upper-right*, *answer*) for each of the six possible answers. The correct answer should be the one whose Δ is most similar to Δ(*upper-left*, *lower-left*).

Scoring the similarity of Δs requires taking into account both those elements that align and those that fail to align. In our example, the correct answer would be a square with a dot located in its bottom right corner. Thus, both the Δs would involve a dot being in the upper half of the square versus the lower half. An answer that contained additional differences, e.g. where the dot also differed in its size or shape, would be less correct. An answer that contained fewer differences, such as the dot being in the same location, would also be less correct. Consequently, we measure similarity by calculating both the percentage of expressions in the base case (the differences between two matrix entries) that align with the target case and the percentage of expressions in the target case (the differences between a matrix entry and a possible answer) that align with the base case. We use the average of these two percentages as our similarity measure.

The entire process described above can also be computed based on the rows of the matrix. Because of redundancy, scoring the answers based on either rows or columns is usually sufficient. However, to ensure maximum accuracy, the system picks an answer based on the average of the scores computed based on rows and columns.

Solving 3x3 Matrices

It may initially seem that the two-stage mapping process described above is insufficient for solving 3x3 matrices. After all, the 3x3 matrices involve understanding a row of three entries, so a system based on comparing only pairs of entries should be unable to solve it. However, as noted above, Raven's matrices contain a high degree of redundancy. We have found that 3x3 problems can be solved by the same two-stage mapping process.

We solve 3x3 matrices by dividing them into four separate geometric analogy problems. As before, each of these problems involves finding differences between two matrix entries and comparing them to the differences between one matrix entry and each of the possible answers. The problems can be formulated as follows:

1) Row: middle : middle-right :: bottom-middle : ?
2) Column: middle : bottom-middle :: middle-right : ?
3) Row-Progressive: bottom-left : bottom-middle :: bottom-middle : ?
4) Column-Progressive: top-right : middle-right :: middle-right : ?

The first two questions are identical to the two used in the 2x2 matrix. That is, they ignore the first entry in their respective rows or columns and consider only the transformation between the second and third entries. The last two questions focus on a single row or column. They look at the change between the first and second entries and compare that to the changes between the second entry and the eight possible answers.

We have concluded that all of the problems in section C of the RPM can be solved by asking one of the four questions given above. Many of them can be solved by asking more than one. For example, our example 3x3 matrix in Figure 1 could actually be solved by asking any of these questions. However, most problems are not quite as easy as this one. Often, the correct answer will actually receive a poor score using the Row-Progressive and Column-Progressive questions. Therefore, it does not make sense to score each answer based on its average score across the four. Instead, we take the maximum score across the questions, assuming that the correct answer should receive a perfect or near-perfect score on at least one of the questions.

6. Performance on the SPM

Because the problems in the Standard Progressive Matrices are precisely rendered, we drew the figures in PowerPoint and pasted them into sKEA. The figures in each entry of the matrices were pasted into separate layers, and the layers were named so that sKEA could easily retrieve the set of glyphs and relations for each entry in the matrix. Each answer was also given its own layer.

We followed the normal sKEA strategy of relying on the user to segment a sketch into shapes. Each of the closed shapes was drawn in PowerPoint as a separate polygon. Lines that were not part of a closed shape were each drawn separately. Thus, when they were pasted into sKEA, they had already been segmented into the appropriate entities. sKEA was still required to segment the closed shapes into edges, so that they could be compared to the other shapes, but because PowerPoint had been used to draw perfectly straight lines, this task was relatively easy. sKEA's only other task before building its representations was to group parallel lines together to form textured closed shapes.

Note that we do not consider the problem of segmenting a scene into objects to be a trivial part of visual processing. In fact, we have previously explored automatic methods for decomposing the ink in a single glyph into edges and closed shapes (Lovett et al. 2007). However, for the present study, we are focusing on other perception problems.

Results

Our system was tested on the 12 problems in section B and 12 problems in section C of the SPM. Chance performance would be 2 correct answers in section B and 1.5 correct answers in section C. Our system correctly answered all 12 problems in section B, and 10 out of 12 problems in section C. Using the norms available on the SPM, we can compare our system's performance to human test-takers. According to the 1979 norms found in Table SPM2 (Raven *et al*., 2000), subjects who scored a perfect score on section B generally scored 52 or higher on the overall test (out of a total score of 60). Subjects who scored a 10 on section C generally scored between 49 and 52. This suggests that, within those sections, our system is performing at the level of test-takers who scored around 52 on the overall test, though we are not claiming our system could score as high on the other sections. According to tables SPM 9 and SPM 10 (Raven *et al*., 2000), a score of 52 is in the 50th percentile for individuals from the United States between the ages of 18 and 45. Thus, our system's performance on sections B and C appears to match the performance of the average American adult.

Discussion

While our system's performance was similar to the performance of typical adults on the SPM, the two problems that it missed were not the most difficult problems in section C, by human standards. The problems were from the middle third of the section, whereas the test is designed to steadily increase in difficulty throughout each section (Raven *et al.*, 2000). Thus, we believe the mistakes made by the system are based on limitations in our spatial representation scheme. By considering the cause of these mistakes, we can gain insights into how the representation scheme can be improved to better model human perception.

The first mistake involves a single closed shape that differs in number of parts between entries. Unfortunately, our current shape comparison algorithm does not handle partial shape matches: it only identifies shapes with the exact same number of edges at approximately the same relative orientations. In the future we plan to use a more forgiving comparison algorithm, which can align some of the edges in two shapes and, based on that alignment, identify differences such as the addition or removal of other edges. The other mistake involves textures; here, the general **TexturedObject** attribute playes a part in misleading the system into choosing the incorrect answer. This result indicates that the system's simplified texture component needs to be refined.

7. Related Work

Ferguson's (Ferguson and Forbus 1999) GeoRep constructed qualitative representations of line drawings. As with our own system, these representations could be used to compare two drawings; they were also used to compare a drawing to itself to look for symmetry. Museros and Escrig (2004) built a system that made qualitative representations

for closed shapes and could be used to compare two such shapes. We believe our system is the first to combine shape comparison with comparing full drawings to solve a complex task.

Other current models of analogical matching include Mitchell's (1993) Copycat program and French's (1995) TableTop. However, these systems were designed primarily to work in a single domain, letter-strings for Copycat and table settings for TableTop. SME, in contrast, works on a general vocabulary that allows it to be used in variety of different domains (Forbus et al. 1997). LISA (Hummel & Holoyak 1997) is a general model of analogy, but to our knowledge it has not been used with automatically generated representations, and could not process representations of the size needed to handle this task.

8. Conclusion

This simulation provides evidence for two important points. First, it demonstrates that our spatial representation scheme encodes sufficient information for solving 22 out of 24 SPM problems that constitute two sections of the entire exam. This, along with independent evidence motivating them, suggests that our representations capture some important properties of human visual representations. Most aspects of these representations have been used in prior simulations (e.g., Tomai et al. 2005), with only two (frames of references inside objects and texture patches) added for this simulation. While further development is needed, especially in texture patches, this result suggests that the representations are on the right track. Second, we have shown that SME, in a two-stage mapping process, can be used to solve easy to mid-level SPM problems. No special-purpose mechanisms are required, lending support to the claim that SME models general processes of structural alignment in human cognition.

While our simulation overcomes the three limitations of the Carpenter *et al* (1990) model, it has a complementary limitation: it is not clear that our model can explain the strategies used by more skilled individuals on more advanced problems. Modeling performance on the advanced problems would have the dual advantages of demonstrating the generalizability of our model and allowing us to determine whether the model can accurately predict human errors, since humans make considerably more mistakes on these problems. We believe it may be possible to approximate the task-specific problem-solving strategies laid out by Carpenter using general analogical processes, although additional comparisons and mapping stages will be needed to deal with the more abstract spatial relationships. In the future, we plan to test out this approach.

Acknowledgments

This work was supported by NSF SLC Grant SBE-0541957, the Spatial Intelligence and Learning Center (SILC).

References

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices test. *Psychological Review, 97*, 404-431.

Cohn, A. (1996). Calculi for qualitative spatial reasoning. In J. Calmet, J. A. Campbell, and J. Pfalzgraf (Eds.), *Artificial Intelligence and Symbolic Mathematical Computation, LNCS 1138*. Springer Verlag, 124-143.

Evans, T. (1968). A program for the solution of a class of geometric-analogy intelligence-test questions. In M. Minsky (Ed.), *Semantic Information Processing*. Cambridge, MA: MIT Press.

Falkenhainer, B., Forbus, K. and Gentner, D. (1986). The Structure-Mapping Engine. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 272-277). San Francisco, CA: Morgan Kaufmann.

Ferguson, R. W., and Forbus, K. D. (1999). GeoRep: A flexible tool for spatial representations of line drawings. In *Proceedings of the 13th International Workshop on Qualitative Reasoning (QR'99)*, 84-91. Loch Awe, Scotland.

Forbus, K., Ferguson, R., and Usher, J. (2001). Towards a computational model of sketching. *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI'01)*. Santa Fe, NM.

Forbus, K., Gentner, D., Markman, A. and Ferguson, R. (1997). Analogy just looks like high-level perception: Why a domain-general approach to analogical mapping is right. *Journal of Experimental and Theoretical Artificial Intelligence* (JETI), 4, 185-211.

Forbus, K., Lockwood, K., Klenk, M., Tomai, E., and Usher, J. (2004). Open-domain sketch understanding: The nuSketch approach. In *AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural*. Washington, DC.

Forbus, K., Tomai, E., and Usher, J. (2003). Qualitative spatial reasoning for visual grouping in sketches. In *Proceedings of the 17th International Workshop on Qualitative Reasoning*. Brasilia, Brazil.

French, R. (1995). *The subtlety of sameness*. Cambridge, MA: MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*, 155-170.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review, 104*, 427-466.

Huttenlocher, J., Hedges, L. V., and Duncan, S. (1991). Categories and particulars: Prototype effects in estimating spatial location. *Psychological Review, 98*(3), 352-376.

Lovett, A., Dehghani, M., and Forbus, K. (2007). Incremental Learning of Perceptual Categories for Open-Domain Sketch Recognition. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. Hyderabad, India.

Mitchell, M. (1993). *Analogy-making as perception: A computer model*. Cambridge, MA: MIT Press.

Museros, L., & Escrig, M. T. 2004. A qualitative theory for shape representations and matching. In *Proceedings of the 18th International Workshop on Qualitative Reasoning (QR'04)*. Evanston, IL.

Raven, J., Raven, J. C., and Court, J. H. (1998). *Manual for the Raven's Progressive Matrices and Vocabulary Scales. Section 1: General Overview*. Oxford: OPP Limited.

Raven, J., Raven, J. C., and Court, J. H. (2000). *Manual for the Raven's Progressive Matrices and Vocabulary Scales. Section 3: The Standard Progressive Matrices.* Oxford: OPP Limited.

Shepard, R. N., & Cooper, L. A. (1982). *Mental images and their transformations*. Cambridge: MIT Press.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science, 171*, 701-703.

Tomai, E., Lovett, A., Forbus, K., and Usher, J. (2005). A structure mapping model for solving geometric analogy problems. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, Stresa, Italy, 2190-2195.

Treisman, A., and Paterson, P. (1984). Emergent features, attention, and object perception. *Journal of Experimental Psychology: Human Perception and Performance, 10*(1), 12-31.

A Garp3 model of environmental sustainability in the River Mesta (Bulgaria)

Elena Nakova[1], Bert Bredeweg[2], Paulo Salles[3], and Tim Nuttle[4]

[1] Central Laboratory of General Ecology, Bulgarian Academy of Sciences, Sofia, Bulgaria.
[2] Human Computer Studies Laboratory, University of Amsterdam, The Netherlands.
[3] Institute of Biological Sciences, University of Brasília, Brasília, Brazil
[4] Institute of Ecology, Friedrich Schiller University, Jena, Germany and Dept. Biological Sciences, University of Pittsburgh, USA.

Abstract

We present a qualitative model of sustainable development issues in the River Mesta, Bulgaria. Following a standardized framework for conceptual description of QR case studies, we have organized our expert knowledge about biological and physical processes in the stream as well as impacts of external influences like pollution, erosion, and water abstraction. We present essential background about the modelled system, and describe how available knowledge was encapsulated into QR knowledge structures including model fragments and scenarios. Finally, we present simulation output based on this knowledge and discuss how this output contributes to understanding factors affecting sustainability of the River Mesta system.

Introduction

To realize the European Union's Strategey for Sustainable Development (SSD; European Commission 2001), citizens must become more educated about factors that affect sustainable development (SD). Qualitative reasoning (QR) has proven effective in educational settings as a means to educate about cause and effect ({Bredeweg, 2003 #4184}). This paper contributes to the objectives of the SSD, in the context of the NaturNet-Redime project, by presenting a QR model about an SD case study. This QR model will become part of a curriculum aimed at teaching concepts of sustainability, including the impacts of biological, physical, and chemical processes on human well-being (Nuttle et al. 2006).

The basic objective of this modelling project is to transfer expert knowledge (contained in a QR model) about processes affecting sustainability to stakeholders, decision makers, and citizens. This paper builds from Uzunov (2006), which presented preliminary progress in organizing qualitative knowledge about the River Mesta system using a "structured approach to qualitative modelling" (Bredeweg et al., in press). Here, we update that information based on the actual content of the implemented model. We begin by presenting essential background on the River Mesta system and a list of model goals. Next, we describe the most and insightful model fragments and scenarios that contribute to fulfilling these goals. We then present simulation results based on these scenarios, highlighting the most relevant behaviour paths. Finally, we discuss how these simulation results contribute to our model goals, in the context of supporting the SSD's educational objectives.

Model System

Varadinova (2006) describes the basic features of the River Mesta. The region is recognized as economically under-developed, with high unemployment. Regional development plans focus on intensifying economic activities based mostly on natural features of the region. This includes further development and diversification of tourism; modernizing and intensifying agriculture and forestry; increasing energy production from hydropower; construction of new roads and streets, and enhancing infrastructure like sewage systems, wastewater treatment plants, and domestic waste landfills.

All of these activities need more water than the River Mesta watershed can supply, potentially leading to conflicts between users. State and local authorities are faced with difficult solutions how meet these competing demands. Reconciliation of these conflicts requires finding of sustainable solutions and appropriate environmental and/or ecosystem health indicators, in addition to the economic and/or social ones usually taken into account.

One of the indicative parameters of aquatic ecosystem health is the amount of dissolved oxygen (DO) in the water. Oxygen is an essential component for all living organisms in the aquatic ecosystem. All water bodies contain some amount of DO due to diffusion from the atmosphere. Normally there is a dynamic equilibrium between inputs and outputs of DO due to the biological

processes of oxygen production and consumption. Water pollution, abstraction, erosion and other human activities can disrupt this balance, worsening ecosystem health and decreasing sustainable uses of ecosystem services.

Based on these factors, being able to discriminate between anthropogenic and natural fluctuations of DO is potentially of great importance for decision making about sustainable and integrated management of aquatic ecosystems. QR provides a modelling paradigm that allows explicit representation of the various processes that interact in a water body to affect DO (Bredeweg and Struss 2003). Furthermore, the ontology provided by QR facilitates education about these processes, which will be useful for explanation to decision makers and stakeholders—those people who have a vested interest in the outcome of sustainable decisions.

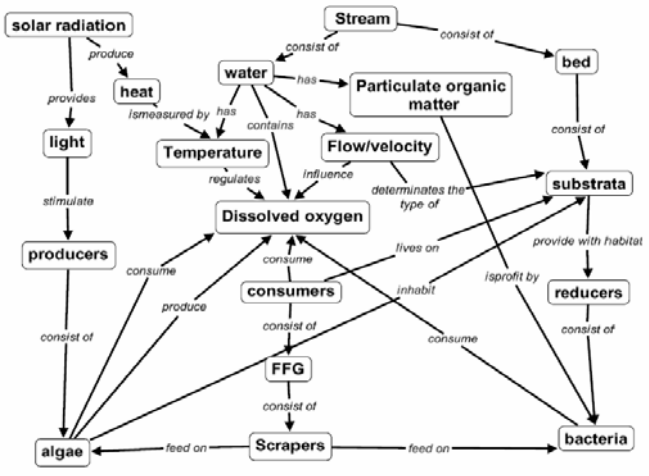


Figure 1. River Mesta concept map

Model Specification

Main Model Goals

We have identified the following modelling goals to focus and narrow the scope of our model. The model should:

1. Describe the behaviour of DO under different conditions (hydro-morphological, physico-chemical and biological).
2. Examine mechanisms of change in ecological functions anthropogenic influences of organic pollution, erosion (due to agriculture and deforestation), and water abstraction.
3. Be useful for scientific and management purposes to explain cause and effect processes to decision makers and stakeholders.

Although the focus of the model is the River Mesta system, the processes should be generalisable to any riverine aquatic ecosystem.

Concept Map

We begin with a concept map that helps identify, clarify, and focus our knowledge about the system of interest (Figure 1). Two main groups of processes influence DO. Physical processes involve solar radiation which provides light and heat, as well as water itself which modify the hydro-morphology of the channel (depth, width, bottom substrata, etc.), thus providing living organisms with habitats. Biological processes involve three groups of organisms responsible for oxygen production (producers: algae) and oxygen consumption (consumers and reducers). All aquatic organisms consume oxygen for their respiration.

Global Behaviour

Here, we identify and describe the main causal processes and how these combine to form the full causal model of the system as well as describe typical scenarios and expected outcomes. These textual descriptions help organize our knowledge for later implementation using QR dependencies.

Main Internal Processes

Oxygen diffusion is a physical process that involves the entity *water*, which has quantities *Temperature*, *DO*, *Light intensity*, and *Heat amount.* This describes the dependence of DO on the water temperature. The lower the temperature, the more oxygen can be held by the water. The process is always active while any water body is above freezing (between 0 and 100 C), which is always true for the River Mesta. The oxygen content within a small, turbulent stream is approximately at or near saturation. DO decreases following warming of water downstream and during summer. Discharges of thermal pollution (effluents of cooling waters from thermal plants for energy production and other industries) may reduce DO substantially in streams and rivers.

Aeration is a physical process that involves the entity *water* and the quantities *DO* and *Flow velocity.* Diffusion of oxygen from the air is facilitated by the turbulent movement of water. This turbulence mixes air and water and thus increases the amount of oxygen dissolved from these mixture. Turbulence is higher when flow velocity is higher and also in shallow water.

River bed substrata is a physical process that involves the entities *Stream*, *Water*, *River Bed*, and *Substrata* with the quantities *Flow velocity* and *Size of substrata particles* (stones/gravel). The kinetic energy of running water modifies the river bed's composition. The higher current velocity, the larger sized particles form the bottom bed. Larger particles (like stones and gravel) provide more surface to be inhabited by living organisms.

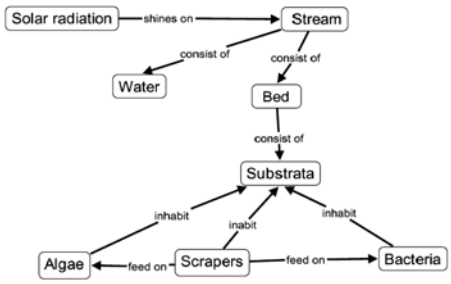


Figure 2: Structural model of the stream system

Oxygen Production is a biological process that involves entities *Light* and *Algae* and quantities *Light intensity*, *Number of algae*, *Photosynthetic rate*, and *DO*. Light from solar radiation is the primary factor for oxygen production through the process of photosynthesis by algae. Pollution and erosion due to effluents of organic and/or inorganic particles seriously reduce light penetration and thus the rate of photosynthesis.

Oxygen Consumption (respiration) is a biological process that involves all living entities (*Scrapers*, *Bacteria*, *Algae*) and the quantities *Amount of* living entities and *DO*. All aquatic organisms consume DO for their respiration thus decreasing its amount in water. Higher water temperatures accelerate the consumption rate.

Feeding (scraping/grazing) is a biological process that involves the entities *Algae*, *Bacteria*, and *Scrapers* as well the *Amount of* each. Scrapers are aquatic invertebrates that scrape (or graze) the thin layer of algae and bacteria (so called bio-film) on the surface substrata. The amount of scrapers depends on the amount and availability of their prey. The process is always active as long as food is available (algae and bacteria); it is assumed that scrapers will re-colonize as soon as food is available. Feeding is strongly and positively related to water temperature and rate of oxygen consumption. External impacts like pollution may negatively influence the process by changing the amount of the food.

Bacterial degradation is a process that involves the entities *Bacteria* and *Water* and the quantities *Amount of Bacteria*, *DO*, and *Amount of POM* (particulate organic matter). Bacterial degradation involves decomposition of organic matter from dead organisms and inputs from the watershed. The process decreases DO. The amount of bacteria depends strongly on the amount of POM in water bodies. Input of POM by urban and industrial wastewaters accelerates degradation until DO is completely exhausted.

External Influences

All pressures that originate outside the River Mesta are considered external influences. We consider three external influences that have the greatest impact on sustainability of the River Mesta.

Erosion increases the amount of suspended solids in the stream, decreasing light intensity. Erosion is created by deforestation and unsustainable agriculture. Sustainable agricultural and forestry practices reduce soil erosion.

Pollution increases POM in the stream. POM affects DO, accelerating oxygen consumption by bacteria, making it less available for scrapers and algae. The effects of pollution depend on the amount of wastewater discharge and water temperatures. POM generally arises from point sources, such as households, industries and other human activities. Hence, wastewater treatment prior to discharge into water bodies can reduce the amount of POM discharging river bodies.

Water abstraction reduces the amount of water in the stream. Humans need water for various purposes of their every-day life (drinking, washing, bathing) and many economic activities – agriculture (irrigation), industry (supply for technological processes and manufactured goods), etc. Decreasing water discharge in natural water bodies affects all physical and biological processes and thus may negatively affect ecosystem health of the River Mesta downstream the abstraction point.

Causal Model

The effects of internal and external processes are refined as causal dependencies following Qualitative Process Theory (Forbus 1984). The full causal model of the River Mesta is depicted in Figure 3. Model documentation fully describes each of the dependencies depicted in Figure 3 (Uzunov et al. 2006). We refer the reader to the textual descriptions above for explanations for each dependency.

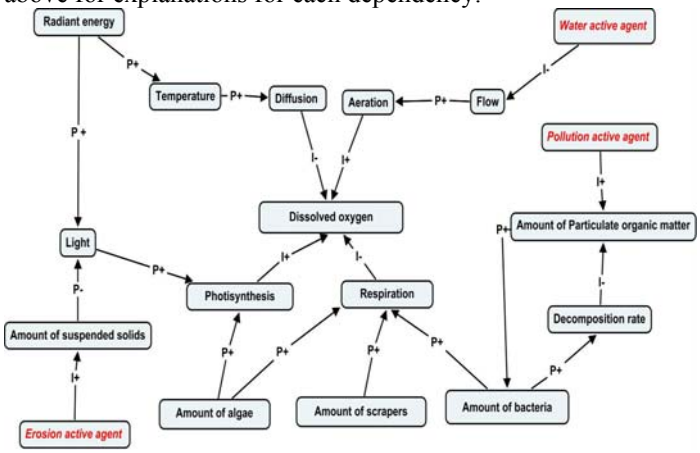


Figure 3. Causal model for the River Mesta, as implemented.

Model Implementation

A model has been implemented in the Garp3 workbench (Bredeweg et al. 2006). Implementation details contain the detailed description of the modelled system: *Entities*, *Attributes*, *Configurations* (structural relationships

between Entities), *Quantities* associated to each *Entity*, *Quantity Spaces* associated to *Quantities*, *Scenarios*, *Model Fragments*, *Agents* (External influences), and *Assumptions*. The implementation phase of model development helped us clarify our thinking on several of the processes described above. Therefore, there are slight differences between the textual descriptions above and how they were implemented. Nevertheless, the basic ideas remain the same, and the description of model fragments makes clear how the processes were modelled.

Entities, Configurations, and Agents

Based on the concept map, we identified the main entities involved in dissolved oxygen balance of the River Mesta. These are organized into a type hierarchy (Figure 4) and described in Table 1.

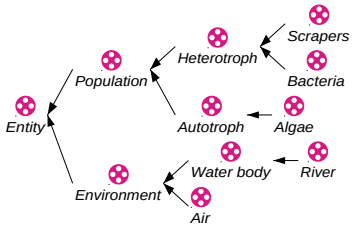


Figure 4. Entity hierarchy for the River Mesta model.

Table 1: Entities involved in the River Mesta QR model

Entity	Description
Stream	A natural water body which consists of some amount of running water and river bed/bottom.
Water	Part of the stream, a fluid that possibly contains dissolved gases and substances. Water has its flow/velocity and temperature.
River bed	The solid background within which the water runs downstream. The bed consists of different types of substrata, the size of particles of which depends on current velocity.
Algae	A kind of producer. Periphytic algae inhabit (live on) substrata (river bed).
Bacteria	Reduce the organic substances and bodies of dead organisms. They play important role in self-purification processes.
Scrapers	A functional feeding group of trophic structure of the community of bottom invertebrates (macro-zoobenthos), which feed on small-sized organ-isms like algae and bacteria, scrapping/grazing them from surface of the bottom substrata they used to live on.
Solar radiation	An environmental factor which is the main source of energy (light and heat) for the aquatic ecosystems.

Configurations are used to model relations between instances of entities and agents. Possible configurations include: River *consist of* Water body and River bed. Population *lives in* River bed.

External influences (described above) were organized into Agents. These include Erosion, Pollution and Water abstraction. The effects of these agents are described in the model fragments (below). Configurations specify whether a given agent is *active in* a water body.

Quantities and Quantity Spaces

We use four quantity spaces (QS) to describe the various quantities in the model. These are summarized in Table 2. The main quantity of interest, *Dissolved oxygen*, uses a QS consisting of five values. We felt this would be more insightful for demonstration and education with stakeholders, being that this is the main focal quantity. For example, the existence of riverine flora and fauna depend of DO concentration and survival rate of organisms decreases with lower DO concentration. Only some populations can survive in low DO concentration

Table 2: Quantity spaces and the quantities they are associated to (▌ indicates an interval value, ● indicates a point value).

Quantity space	Quantities	Associated Entity
‹mzp› ▌plus ●zero ▌min	All derivatives	
‹vllrhvh› ▌very high ●high ▌regular ●low ▌very low	*Dissolved oxygen*	Water body
‹zp› ▌plus ●zero	*Biomass,* *Photosynthesis;* *Production*; *Respiration;* *Diffusion rate,* *Aeration rate,* *Excretion rate;* *Decomposition rate;* *Soil particles;* *Pollution;* *Solar radiation;* *Water abstraction*	Population Algae Heterotroph Water body Bacteria Erosion (agent) Polluter (agent) Sun (agent) Water abstractor (agent)
‹interval› ▌interval	*Flow velocity, Light, Temperature, Heat, Particulate organic matter;* *Biomass*	Water body Population

(microaerobic conditions). Very low concentrations result in anaerobic conditions where only some bacterial populations can exist. Thus, being able to distinguish these qualitative values is useful for explaining consequences of certain behaviours to learners.

All "intermediate" quantities, whose behaviour is modelled mainly to show how they affect DO, but whose values are not directly of interest, use a simple QS; serving to reduce ambiguity in the simulation. Quantities that influence others via a direct influence (I+ or I-, Forbus 1984) require a known zero so it can be known whether the quantity is positive or negative. Hence, they use QS *‹zero, plus›*, or *‹zp›*. Quantities that influence via qualitative proportionalities (P+ or P-) don't require a known zero (their derivative carries the influence), so they have a QS consisting of a single interval, *‹interval›*. The exception to this rule is that quantities associated to agents use QS *‹zp›*, because we wanted to represent whether the process was active *‹plus›* or inactive *‹zero›*, even thought they act via qualitative proportionalities.

Model Fragments

As a guiding principle for organizing causal dependencies into model fragments, we viewed a water body to represent a dynamic equilibrium of positive and negative influences. For example, DO is in equilibrium between aeration and diffusion, whereas populations are in balance between production (or photosynthesis) and respiration. The prevalence of one of the processes can change the equilibrium. External influences, acting via agents, upset the equilibrium and may cause the system to establish a new lower equilibrium DO content or population size. These unsustainable and unfavourable conditions will persist unless the external influences are removed.

Another basic principle we employed is that values for quantities with only a single interval QS are set in model fragments rather than in scenarios (see e.g. Figure 5); this allows for sparser scenarios, which are clearer to communicate to stakeholders. Finally, we have constructed the model fragments using the "one concept, one model fragment" principle (Salles and Bredeweg 1997), which also helps in developing learning materials based on the model and makes the most of the capabilities of compositional modelling (Falkenhainer and Forbus 1991).

Aeration and Diffusion: These balancing processes are the main controls on DO content of a river. Figure 5 shows the *Diffusion* process model fragment (MF). It imports a model fragment *Water body* that specifies some basic information about all water bodies. The *Diffusion* MF uses a qualitative proportionality to model the fact that as *Temperature* increases, so does *Diffusion rate*. Diffusion reduces the amount of DO in a water body; thus there is a negative direct influence of *Diffusion rate* on *Dissolved oxygen*. Finally, there is a feedback (via P+) from *Dissolved oxygen* to *Diffusion rate*, because the more DO in the water, the more can diffuse out.

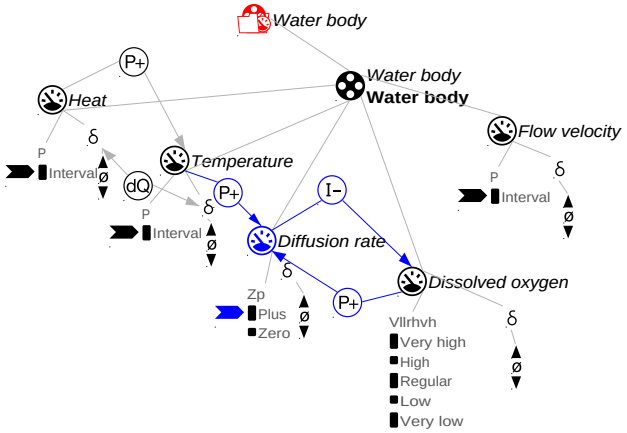


Figure 5. The diffusion process model fragment.

Aeration process MF is structurally similar to *Diffusion*. As *Flow velocity* increases, so does *Aeration rate* because the churning of turbulent water flow causes air to be incorporated into the water; this is modelled using a qualitative proportionality. Aeration causes an increase in the amount of DO; thus there is a positive direct influence (I+) from *Aeration rate* to *Dissolved oxygen.* In contrast to *Diffusion*, *Dissolved oxygen* does feed back on *Aeration.*

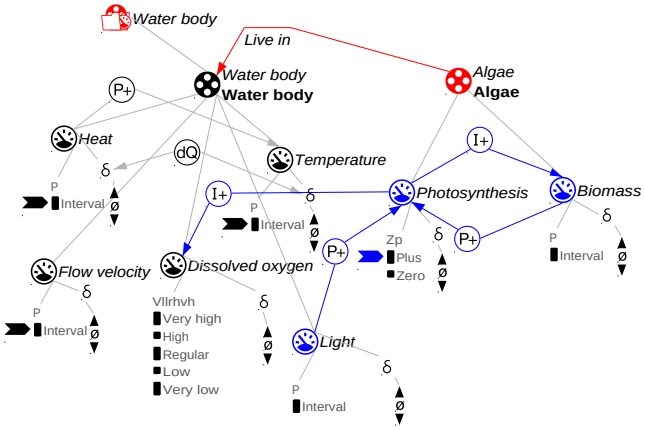


Figure 6. The photosynthesis process model fragment

Photosynthesis is another process that can add DO to a water body. The rate of *Photosynthesis* is positively proportional to both the amount of *Light* and *Biomass* of *Algae*: the more light and the more algae there are, the more photosynthesis happens. *Photosynthesis* rate increases (I+) the *Biomass* of the *Algae* and increases (I+) the amount of *Dissolved oxygen* in the water body.

Production and Respiration: The *Production* process is analogous to *Photosynthesis* except it applies to

Heterotrophs (including *Bacteria* and *Scrapers* but not *Algae*). So far, we don't specify the source of energy for the production rate, but in the future this will come from feeding on another population. Another difference is that *Production* rate does not increase *Dissolved oxygen*.

The *Respiration* process applies to any *Population* (including therefore *Algae* and *Heterotrophs*). *Respiration* rate reduces (I-) *Biomass* and Dissolved oxygen. There is a feedback (P+) from *Biomass* to *Respiration* rate.

Decomposition is a process whereby bacteria break down particulate organic matter (POM), using up DO in the process. *Bacteria* have a *Decomposition rate*, which decreases (I-) both the amount of *Particulate organic matter* and *Dissolved oxygen* in a *Water body*. There is a feedback (P+) from *Particulate organic matter* to *Decomposition rate*, so the more POM, the faster it is broken down and the faster DO decreases.

Excretion is the creation of POM from the dead bodies and excrement (in the case of scrapers) of each of the populations in a river. In one model fragment, the *Excretion rate* is positively proportional (P+) to the *Biomass* of each of these populations (*Algae, Bacteria,* and *Scrapers*). A process model fragment (*Excretion*) then specifies that *Excretion rate* increases *Particulate organic matter* in a *Water body*.

Agent model fragments: There are four agent model fragments: Erosion, Polluter, Sun, and Water abstractor. Normally, agents are implemented with rates that affect aspects of the model system via direct influences (I+ or I-). However, from our point of view, each of these processes is always active to some degree on any water body; hence, we are interested in showing how *changes* in the status quo propagate through the system to affect the DO balance of the stream. Hence, each of these agents is associated with a quantity that influences one or more of the endogenous system quantities via a qualitative proportionality. Effects of the agents are simulated when the configuration *active in(Water body, [Agent])* and the relevant quantity's derivative is specified in a scenario. The effects of the agents and their relevant quantities are:

P+(Water body: Light, Erosion: Soil particles)
P+(Water body: Particulate organic matter, Polluter: Pollution)
P+(Water body: Light, Sun: Solar radiation)
P+(Water body: Heat, Sun: Solar radiation)
P-(Water body: Flow velocity, Water abstraction: abstractor)

Scenarios

Scenarios present initial situations, including the configuration of the system of interest and starting values for quantities. We present two scenarios, each starting off with *dissolved oxygen* in the interval ‹*Regular*›. One (Scenario A) shows only how physical aspects of the system affect DO via their effects on aeration and diffusion rates (Figure 7). The other (Scenario B) shows both physical and biological effects on DO (Figure 8). In Scenario B, to reduce ambiguity, *production* and *respiration* rates start out equal for *scrapers* and *bacteria*; *photosynthesis* and *respiration* start out equal for *algae*, and *decomposition* and *excretion* rates as well as *aeration* and *diffusion* rates also start out equal.

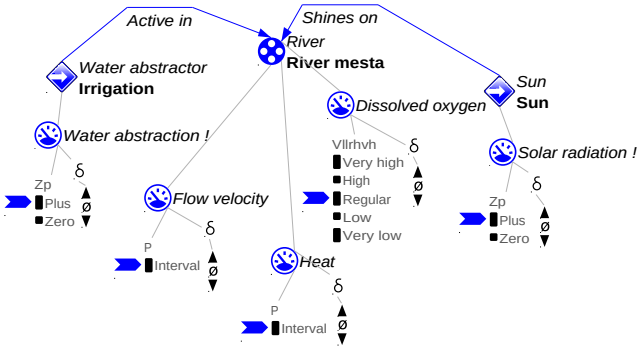


Figure 7. Scenario examining the interaction of physical processes in affecting DO of the River Mesta. Quantities associated to the agents Sun and Irrigation are set to be exogenously increasing (note exclamation marks).

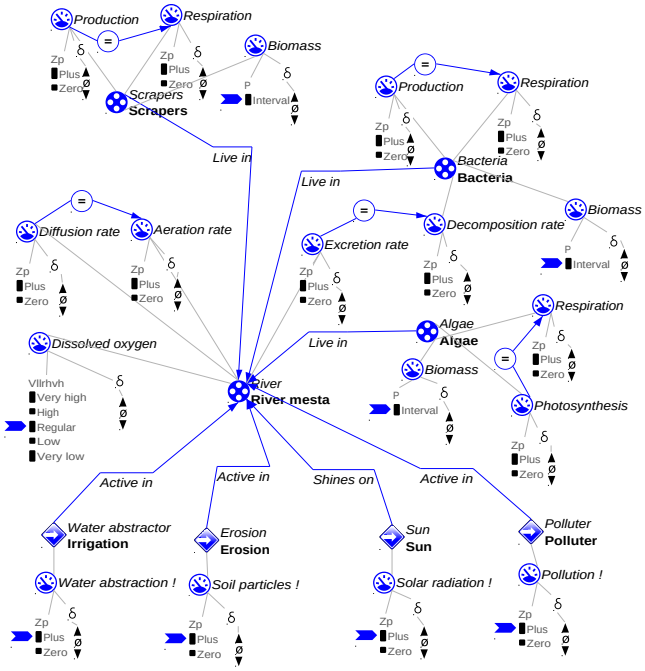


Figure 8. Scenario examining the interaction of physical and biological processes in affecting DO of the River Mesta. Quantities associated to the agent Polluter is to be exogenously increasing whereas agents Water abstracter, Sun, and Erosion are exogenously steady (note exclamation marks).

Simulation Results

Scenario A: Physical Processes

Simulation of this scenario results in five initial states (states 1 – 5, Figure 9) that represent possible interpretations of the net effects of aeration and diffusion on dissolved oxygen. Causal dependencies (Figure 10) give rise to a total of 20 possible states originating from these initial five. Figure 9 (right) presents value histories of three pathways that are representative of the behaviour that *Dissolved oxygen* may take (all states are shown in the three value histories; readers can therefore track each possible transition in the state graph). Note that for each of the value histories presented, the system may start out at any state ≤ 5 (e.g., the middle value history represents paths [5 → 9 → … → 10 → 11] as well as path [2 → 10 → 11]). In each path, *Dissolved oxygen* eventually reaches value *‹very low, -›,* even if it initially increases. The equation histories (Figure 9, bottom) make clear that this behaviour occurs due to the relative magnitudes of *Diffusion rate* and *Aeration rate*. The full causal model of all physical aspects of the system is depicted in Figure 10. Scenarios (not shown) where the agents are both exogenously decreasing result in the opposite behaviour (*Dissolved oxygen* eventually increases to *‹very high, +›*). When the agents take different behaviours (one increasing, the other decreasing or stable, etc.), then all behaviours are possible, and *Dissolved oxygen* may stabilize at any value in the QS.

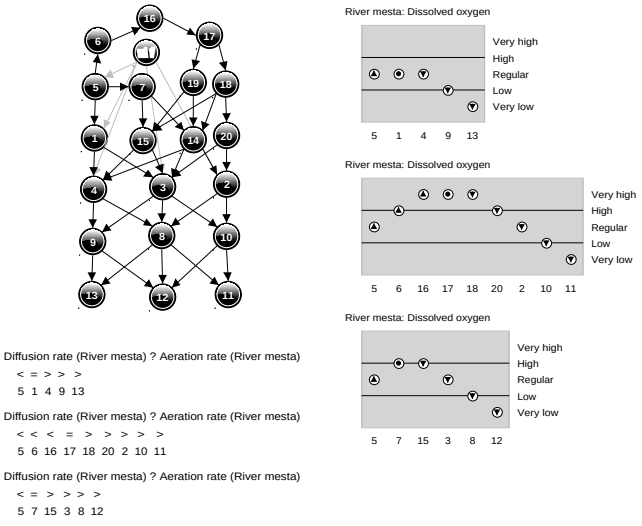


Figure 9. State graph (top left), value histories of selected paths (top right), and corresponding equation histories (bottom) for Scenario A.

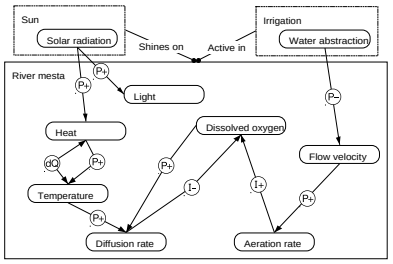


Figure 10. Causal model for Scenario A.

Scenario B: Biological and Physical Processes

Simulation of this scenario results in one initial state (Figure 11) with *Dissolved oxygen* decreasing. Causal dependencies (Figure 12) give rise to a total of 17 states in the full simulation, with 11 possible end states (Figure 11). In these end states, *Dissolved oxygen* either stabilizes at a value less than the starting value or continues to decrease in the interval *‹very low›* (see value histories, Figure 13). This happens because as *Pollution* increases, so does *Decomposition rate,* which has a negative influence on *Dissolved oxygen* (all other processes are held constant in the scenario). *Dissolved oxygen* may stabilize after initially decreasing because with less *Dissolved oxygen* in the river, *Diffusion rate* becomes less. Hence, *Dissolved oxygen* may stabilize when the combined negative influences of diffusion, decomposition, production, and respiration become equal to the combined positive influences of aeration and photosynthesis. Since none of these change except *Diffusion rate* and *Decomposition rate*, it all depends on the relative size of these two quantities.

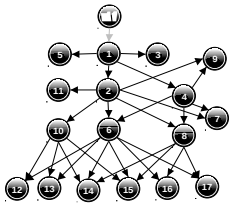


Figure 11. State graph for Scenario B.

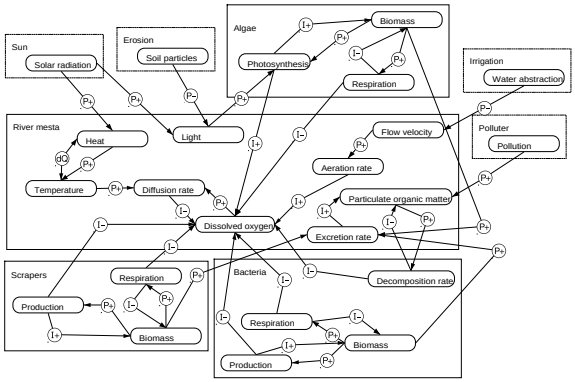


Figure 12. Causal model for Scenario B.

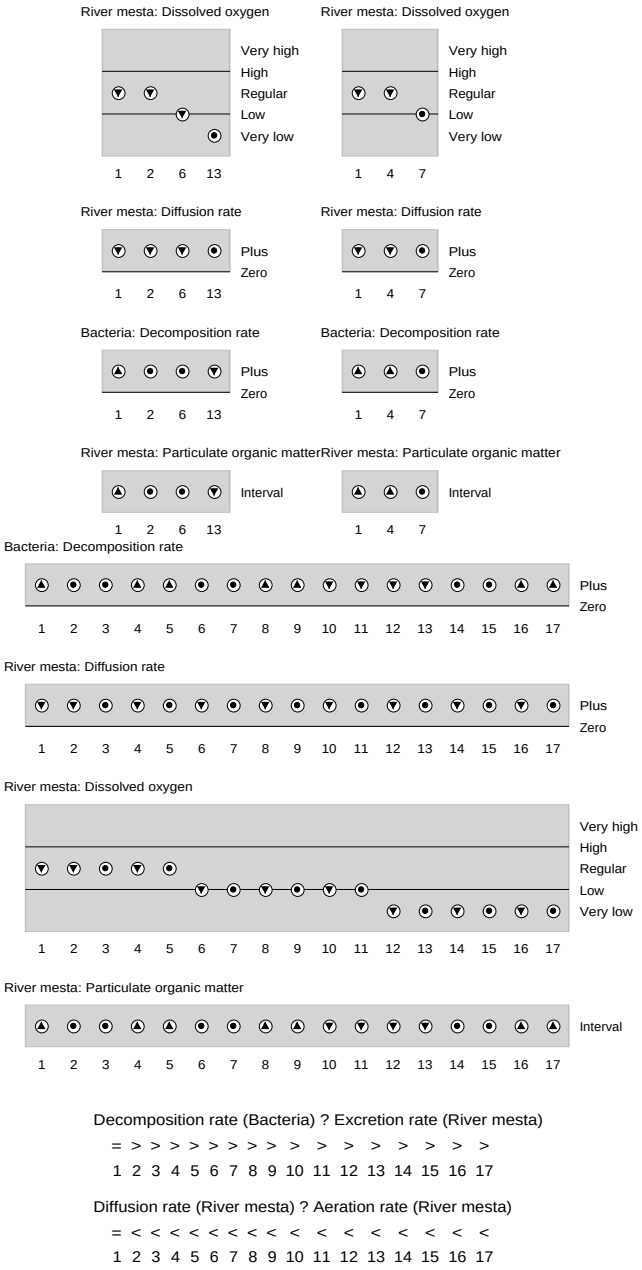


Figure 12. Value histories for two selected paths (top) and for all states in the full simulation of Scenario B (middle), showing quantities where the value may change during the simulation, as well as equation histories (bottom) for balancing processes. The two value histories at top show representative paths; other paths are variations and can be constructed from the state graph (Figure 11) and the corresponding values in the value history (middle).The value for *Pollution* is ‹*plus, +*› and all other values are either ‹*plus, 0*› or ‹*interval, 0*›, as appropriate for the quantity (see Table 2).

Conclusions

This paper has presented progress on a Garp3 model of sustainability issues in the River Mesta, Bulgaria. The model focuses on dissolved oxygen in the river as an indicator of healthy status of the river ecosystem. Since the initial model specification stage (see Nakova et al. 2006), many issues have been clarified and further refined as the ecological knowledge has been captured into the QR ontology using the Garp3 workbench.

In building this model, we employed three modelling principles that are interesting from the QR perspective. First, state variables (DO and population size) are all impacted by balanced negative and positive influences. Second, all intermediate rates are modelled with the simplest quantity space possible, either an open interval (for processes operating via qualitative proportionalities) or quantity with a point value for zero and a positive interval (for processes operating via direct influences). Finally, we employed a large quantity space for the focal quantity representing DO so that a broader range of dynamics can be visualised by users. Although these values are not assigned to quantitative landmarks, they are useful in explaining under what DO conditions certain groups of organisms can persist in the stream.

The model thus captures the most important processes directly and indirectly affecting dissolved oxygen in a river body. By employing a clear ontology for expressing balancing influences, we were able to capture the expert ecological knowledge into model fragments that are both relatively self-contained and insightful. These can be reused and assembled by the Garp3 reasoning engine to make more complex causal models of multiple populations. Further work on the model will better describe the trophic interactions among the three populations. Specifically, feeding relationships between scrapers and bacteria and between scrapers and algae need to be specified.

Once finalized, the model satisfies the first two modelling goals specified above. Concurrent work in the NaturNet-Redime project centres on developing educational materials from this and other models of other case studies to teach about issues concerning environmental sustainability (Nuttle et al. 2006) This will help us to satisfy the third goal for the model, namely to be used as a tool for decision makers and stakeholders to make more informed decisions concerning sustainable development of the River Mesta system.

Acknowledgements. This research was funded by the European Commission's Sixth Framework Programme for Research and Development (project number 004074, project acronym Naturnet-Redime). Information on the topics

being studied in this project can be found at http://www.naturnet.org.

References

Bredeweg, B., and P. Struss. 2003. Current topics in qualitative reasoning. AI Magazine **24**:13-16.

European Commission. 2001. Communication from the European Commission: a Sustainable Europe for a Better World: A European Union Strategy for Sustainable Development. Brussels, Belgium.

Falkenhainer, B., and K. Forbus. 1991. Compositional modelling: finding the right model for the job. Artificial Intelligence **51**:95-143.

Forbus, K. 1984. Qualitative process theory. Artificial Intelligence **24**:85-168.

Nuttle, T., P. Salles, A. Bouwer, B. Bredeweg, A. L. R. Caldas, E. Cioaca, S. Covaliov, J. Liem, E. Nakova, M. Neumann, R. Noble, E. Varadinova, Y. Uzunov, and A. Zitek. 2006. Navigating a landscape of sustainability concepts: towards progressive learning routes within and among qualitative reasoning models. Pages 2-1 - 2-10 *in* M. Sànchez-Marrè, M. Oprea, and F. Wotawa, editors. Proceedings of the 5th ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, part of the 17th European Conference on Artificial Intelligence, Riva del Garda, Italy.

Uzunov, Y., E. Nakova, and E. Varadinova. 2006. Textual description of River Mesta case study. STREP project co-funded by the European Commission within the Sixth Framework Programme (2002-2006), Project no. 004074, Project Deliverable Report D6.3.1.

Varadinova, E. 2006. Studies on the Functional Feeding Groups of the Macrozoobenthos along the Mesta River [in Bulgarian]. PhD Thesis. Bulgarian Academy of Sciences, Sofia, Bulgaria.

Integrative qualitative modelling of ecological and socio-economic aspects of river-rehabilitation in England

Richard Noble[1], Tim Nuttle[2] and Paulo Salles[3]

[1] University of Hull International Fisheries Institute, Hull, HU6 7RX, England.
[2] Institute of Ecology, Friedrich Schiller University, Jena, Germany and Dept. of Biological Sciences, University of Pittsburgh, USA.
[3] University of Brasilia, Brazil

Abstract

This poster present the planning phase of a QR model following the structured modelling approached described by Bredeweg *et al.* (2007). The model is aimed at exploring issues and outcomes for rehabilitation of Atlantic salmon stocks in the River Trent. These stocks have been lost due to impacts from the industrial revolution. Rehabilitation programmes within the context of the European Union's legislation for sustainable water management require that ecological status be balanced alongside socio-economic needs of society. This model uses the ecological processes of the salmon life cycle as a focus around which socio-economic factors of rehabilitation are also modelled. The completed model could be applied as a tool for environmental managers, scientific researchers and stakeholders to learn about rehabilitation programs and their potential outcomes.

Introduction

This paper describes development of a Qualitative Reasoning (QR) modelling case study which was developed as part of the EU funded NaturNet-Redime (NNR) project (Sixth Framework Program, Project no. 004074 http://www.naturnet.org). This project contributes to the European Union's Strategy for Sustainable Development (European Commission 2001) in the context that the NNR project revolves around developing new technologies (of which QR is one example) for use within a general educational setting, for exploration of sustainable development issues. One of the goals of the NNR project was to improve the Garp3 QR software tool and to construct guidelines for a standardised approach to QR modelling (Bredeweg *et al*. 2007). Within the QR component of the NNR project five case studies were developed, which contribute to creating programmes for education about sustainability (see Salles *et al.*, Cioaca *et al.*, Nakova *et al*., and Zitek *et al*. also submitted to this QR workshop). All these case studies focus on issues pertaining to aquatic ecology and sustainable management of freshwater ecosystems. The case study presented here focuses on the modelling of species life cycles within the context of river rehabilitation programmes. As such the case study addresses the issues of sustainability of aquatic resource management, an issue that is currently of great concern in the European Union. The European Water Framework Directive (WFD) (EU 2000) provides legislation for a standardised approach to the sustainable management of water resources in the European Union. As such management and rehabilitation of rivers under the WFD represents an example of sustainable development.

River rehabilitation projects in the UK are often targeted to economically valuable fish species (e.g. Atlantic salmon, *Salmo salar* L.). Conservation and management of these species is often based around quantitative life cycle models (e.g. Aprahamian, Wyatt & Shields, 2006) which examine the recruitment of individuals to each consecutive life stage to identify the factors that are impinging on, or limiting the size of, the population. Hence, planning of rehabilitation activities focuses on the key human activities that impact on the life cycle of the fish populations/community being managed. In this context, QR modelling of rehabilitation issues based around a species life-cycle model has the potential to provide a basis for environmental managers, researchers, stakeholders and students to investigate the potential outcomes and conflicts within a given rehabilitation programme. Indeed, QR modelling has been previously used to examine the functioning of Atlantic salmon redds (spawning areas) (Guerrin & Dumas 2001a,b). However, this model focused only on one phase during the life cycle. A QR model which encompasses a full life cycle, together with the socio-economic factors of sustainable management, may be an extremely valuable tool.

The goal of this paper is to describe the formulation of the case study following the "structured approach to qualitative modelling" as described by Bredeweg *et al*. (2007). The standard approach describes four main steps to be undertaken during the formulation and planning of a

QR model. These steps are designed to focus the modeller on: 1) describing the general *model system* and *model goals* within the context of the domain specific concepts of the case study (including concept maps using domain specific terminology); 2) identification and specification of the key components of the model system; 3) description of potential scenarios and general *global behaviours* expected, and 4) *detailed specification of model system structure* using Garp3 terminology. All these elements contribute to a detailed model plan prior to implementation in the Garp3 QR software. This paper presents the outputs from these fours steps; the model itself is currently being implemented in Garp3 and will be forthcoming.

Model System

Prior to the industrial revolution, the River Trent had diverse and prolific fish stocks, and supported good fisheries. However, with the expansion of industry, the fishery began to decline, and water quality reached its lowest level in the 1950s. Long stretches of the Trent suffered from a lack of dissolved oxygen and were devoid of fish until the 1970s. Since the 1970's improvements in waste water treatment and management have led to great improvements in water quality to the point where these are no longer considered to be limiting factors to ecological recovery (Cowx, 1986; Cowx & O'Grady, 1995a, b; Sykes, 2004). In the past the River Trent was undoubtedly an important salmon river. The decline in catches from the 1880s onwards was almost certainly because of pollution and obstructions to the passage of migrating adult salmon (e.g., weirs used for water regulation; Cowx, 1986; Cowx & O'Grady, 1995a, b; Sykes, 2004). In recent times much consideration has been given to the potential to rehabilitate some of these semi-redundant weirs and to rehabilitate the salmon populations of the Trent (Cowx, 1986; Cowx & O'Grady, 1995a, b; Sykes, 2004).

Much of the promotion of the potential rehabilitation of salmon populations lies around the fact that beyond their ecological and conservation value, salmon fisheries may have a high social and economic value. However, whilst there are potential socio-economic benefits of re-establishing salmon populations to the Trent catchment, alterations to the longitudinal barriers (i.e., weirs) on the Trent may detrimentally affect other user groups and impede continued development of other resources in the Trent catchment. If salmon are re-introduced, these conflicts between users will have to be resolved through negotiation with the various user groups. In addition, future proposals on the river will have to be evaluated to prevent degradation of the catchment and permit the sustained development of the salmon stocks.

Main Model Goals

A QR model for this case study could provide a tool that:

1. enables stakeholders, environmental managers, researchers and students to investigate factors limiting salmon stocks in the River Trent, to learn what activities are required to re-establish a salmon fishery in the River

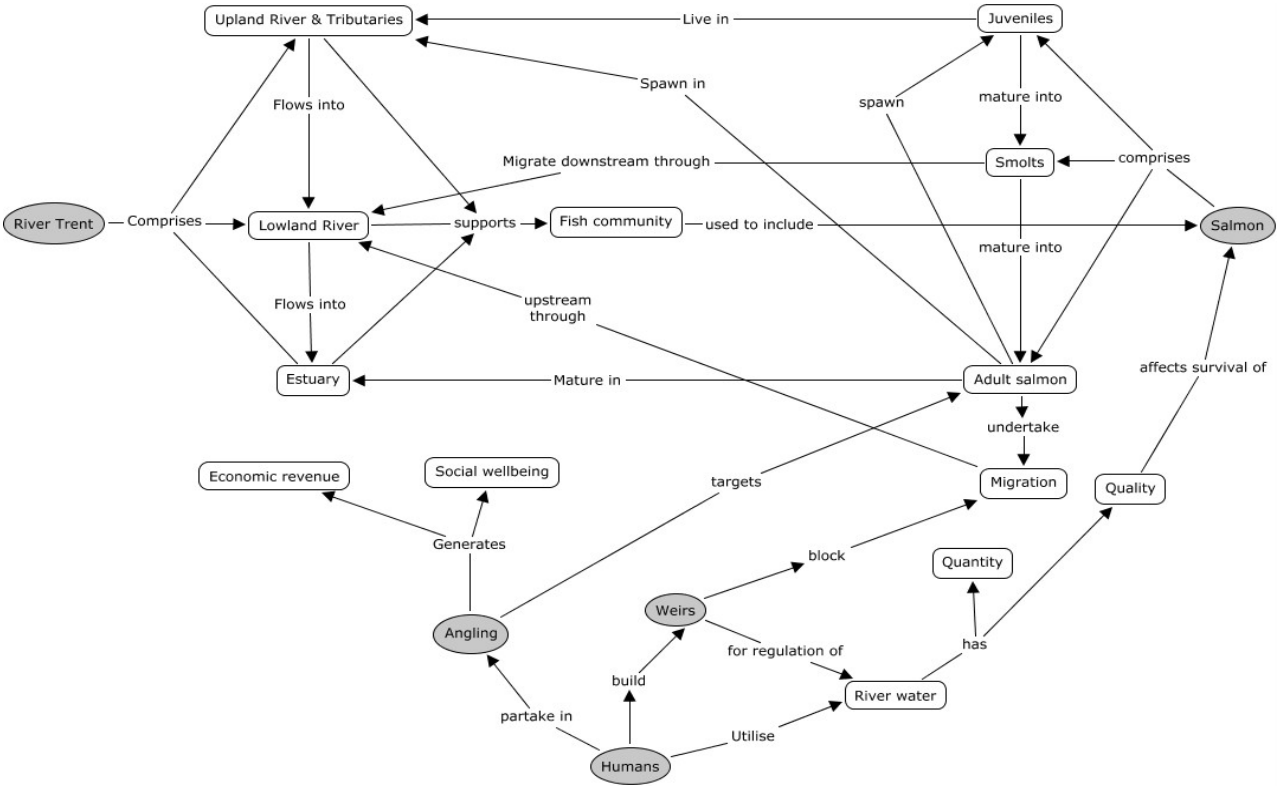


Figure 1. Concept map for salmon rehabilitation in the River Trent.

Trent, to describe the potential outcomes of rehabilitation activities and to identify potential stakeholder conflicts;

2. shows potential socio-economic benefits of re-establishing a salmon fishery, together with the socio-economic costs of rehabilitation both to the fishery and other river users.

Concept Map

The first stage in the structured modelling approach was to construct a concept map that helps identify, clarify, and focus knowledge about the system of interest (Figure 1). This concept map captures the domain knowledge that Atlantic salmon is a migratory species that reproduces in the upper reaches of rivers, but grows to maturity in the North Atlantic Ocean. Adult salmon spawn (lay eggs) into redds (nests), which the adult salmon dig in suitable gravel substrates of the river bed. Juvenile salmon remain resident in fresh water until they attain a size at which, as "smolts", they can migrate down river to the estuary and open ocean. The concept map indicates the key life stages in a salmon population, the different habitats they utilize, the way humans directly exploit salmon and how human activities can damage the salmon population. This concept map also helps define the *entities* in the model system (physical objects or abstract concepts that play a role within the system) and the contextual relations between them (termed *configurations* in Garp3).

System structure

The entity hierarchy and system structure (the way entities in the model relate to each other, and the configurations between them) for the model are designed to account for the different life stages of a salmon population, the different habitats in a river and the different sections of human society that may influence, or be influenced by, rehabilitation projects. Essentially, the model has four entities: human, salmon, river and weirs. These are divided into a number of sub-entities as follows:

- Human
  - General Population
  - Environmental manager
  - Stakeholder
  - Anglers
- River
  - Catchment
  - Upland river
    - Spawning habitats
    - Juvenile habitats
  - Lowland river
  - Water
- Fish
  - Salmon
    - Egg
    - Juvenile
    - Smolt
    - Returning adult
    - Spawning adult
- Weirs

Global Behaviour

Main Processes

The model for the rehabilitation of the Atlantic salmon populations of the River Trent is designed to capture life cycle processes that are integral in sustaining a viable salmon population. The life cycle processes that are generally considered within rehabilitation programmes are mortality/survival, natality, migration, individual growth/maturation, and immigration (from neighbouring populations). This model needs to consider different life stages of salmon populations as different human pressures affect different parts of a river and consequently may have different effects on different life stages of a migratory species.

Given that the case study objectives are for the socio-economic aspects of river rehabilitation to be captured in a QR model, there are a number of human activities that can be seen as integral processes. According to expert reviews of the Trent (Cowx, 1986; Cowx & O'Grady, 1995a, b; Sykes, 2004) the key human activities that affect the system and should be included are pollution, habitat degradation, river regulation (for a variety of purposes e.g. abstraction, hydropower generation etc.) and salmon angling (exploitation of returning adult salmon).

Rehabilitation of a salmon population in the River Trent requires a number of activities to ameliorate the impacts of human population utilisation of the river. The key aspects for rehabilitation that should be considered are mitigation or removal of barriers (weirs and fish passes), stocking of juvenile salmon, waste water treatment, appropriate management of abstraction and rehabilitation of habitat.

Causal Model

In the salmon life cycle model (Figure 2) which forms the focus of the River Trent case study, the numbers of each life stage are regulated by the natality rate, recruitment rates and mortality rates between each life stage. Each recruitment rate acts to increase (through positive directed influences, I+ in Garp3) the numbers of the next life stage whilst also reducing the numbers of the previous life stage (through negative directed influences, I- in Garp3). This reflects the maturation process. The activity of the recruitment rate is controlled by a table of allowable values between the numbers of the previous life stage and the

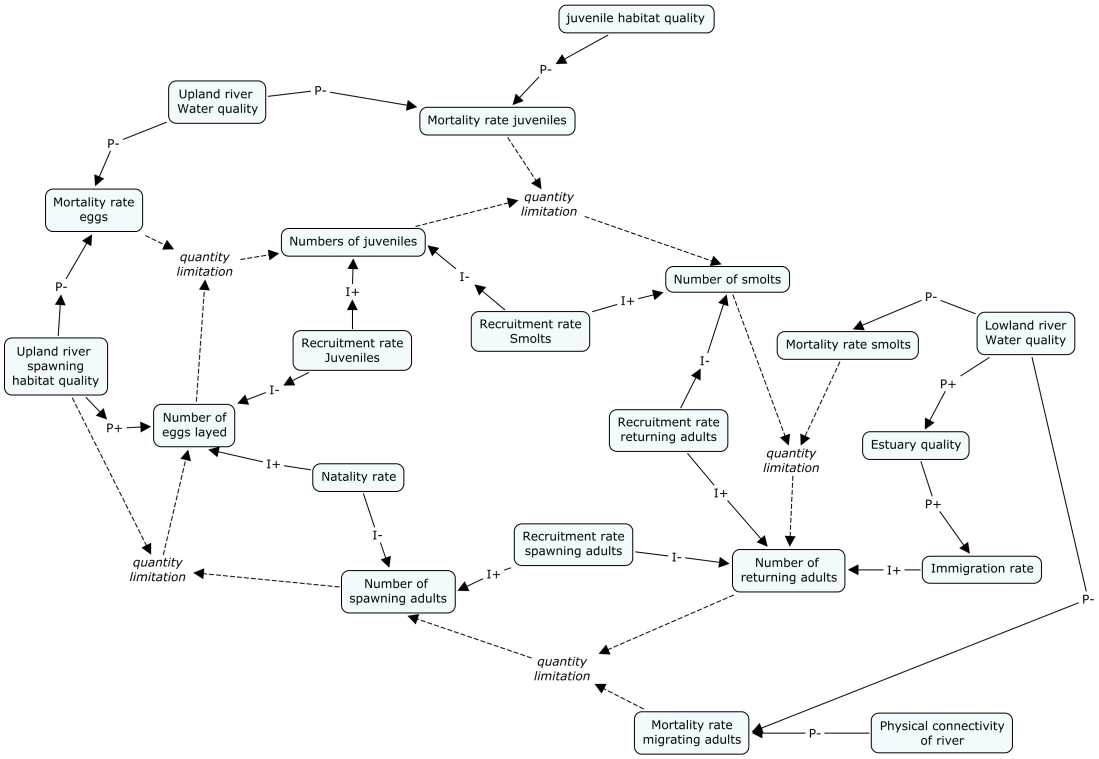


Figure 2. Causal model for the salmon life cycle which forms a focus for the overall QR model. "I" indicates a directed influence and "P" indicates a proportionality within the Garp3 software.

mortality rate of the previous life stage (e.g. Table 1). In this situation the magnitudes of each recruitment rate are equal to each other, and as such only act to increase or decrease the numbers of each life stage. The maximum numbers occurring at each life stage are regulated by the mortality rate of each life stage. This mortality rate controls how many of the previous life stage survive to recruit to the subsequent life stage. For each of the human activities, the intensity of the activity acts to alter the mortality rates between specific life stages. These human activities affect the mortality rates through positive (P+) and negative (P-) proportionalities in the Garp3 software.

Table 1 Matrix of allowable values for the number of life stage x+1 based on the combined values of number of life stage x and the mortality that is applied.

N life stage x+1 Matrix		Level of mortality / increase in mortality			
		Natural	Low	Medium	High
N life stage x	High	*High*	*High*	*Medium*	*Low*
	Medium	*Medium*	*Medium*	*Low*	*Very low*
	Low	*Low*	*Low*	*Very low*	*Very low*
	Very low	*Very low*	*Very low*	*Very low*	*Zero*
	Zero	*Zero*	*Zero*	*Zero*	*Zero*

Scenarios

There are two main scenarios and behaviours that can be explored with this model. Within these two main scenarios a whole suite of specific scenarios could be developed depending on the type of end-user and their existing knowledge. These scenarios are envisioned to be used by environmental managers to demonstrate to stakeholders or students the variety of critical bottlenecks to establishment of salmon populations and the likely outcomes of proposed rehabilitation measures.

1) Salmon population exists.
This type of scenario can be used to explore the effects of human activities in a system where a salmon population is present and is under threat from human pressures.
2) Salmon population absent due to physical barriers to migration (actual situation).
This scenario could have many variations depending upon how much rehabilitation was feasible based on socio-economic conditions and other external factors. Within this group of scenarios two particular scenarios/behaviours can be explored:

A. "Stocking only scenario" - In this scenario salmon are absent from the Trent but the environmental managers and fishery stakeholders attempt to re-introduce a salmon population by stocking juvenile salmon into suitable juvenile habitats in the Upper Trent catchment. In such a

situation the numbers of juveniles increases but this doesn't correspond to establishment of a self-sustaining population as management has not addressed the issue of barriers to migration of adults.

B. "Stocking plus rehabilitation of weirs" - This scenario is similar to A except that at the same time environmental managers and fishery stakeholders act to rehabilitate the connectivity of the river by increasing the passability of weirs through the construction of fish passes. This allows any returning adults a chance to return to spawn in the upper reaches of the river.

Detailed System Structure and Behaviour

The case study described above is being developed in the Garp3 collaborative QR modelling workbench. The software provides a structured approach to organizing causal dependencies into a series of model fragments (partial models which are composed of multiple ingredients capturing one concept about the model system). Within the structured approach to modelling the model planning phase includes definition of a detailed system structure and behaviour. One key element of this section is the textual description of model fragments in terms of the conditions for them to be active, the consequences of their activity, including dependencies and assignments of qualitative values (via correspondences, calculations, or inequalities). Textual descriptions of model fragments serve to guide implementation within the Garp3 workbench. For the purposes of this paper, only a few model fragments are described as examples.

Life-cycle transition fragments

Each life-cycle transition will have its own basic model fragment describing the relationships between numbers of each life stage and mortality and maturation/recruitment. For example:

Recruitment egg to juvenile

- *Condition*
  - if there is a salmon population and number of eggs > zero
- *Consequence*
  - then there are entities "eggs" and "juveniles" and there is a quantity "recruitment rate" which is "plus" and a quantity "mortality rate eggs"
  - there is positive influence I+ from "recruitment rate" to number of juveniles and a negative influence I- to the number of eggs.
  - Then "mortality rate eggs" has value correspondences with number of juveniles to limit how many juveniles come from any number of eggs.
  - There are correspondences between these "number of" quantities that set limits for the number of juveniles a certain number of eggs could produce.

These recruitment fragments are required so that conceptually there is a process which increases the numbers of life stage x+1 whilst at the same time reducing numbers at life stage x. This is equivalent to maturation from x to x+1. As all these rates are equal, the numbers of each life stage will not change unless altered by changes in mortality rates which limit how many of life stage x+1 can survive from a number of life stage x under a certain mortality rate.

Mortality relationship fragments

This set of fragments will be required to model the relationship between the "habitat", "water" and other river quantities and the mortality rates at each stage. These relationships are effectively proportionalities and value correspondences. For example:

Egg mortality is influenced by spawning habitat quality and upland water quality

- *Condition*
  - if there is a river and a salmon population
- *Consequence*
  - then there are quantities "upland river habitat quality", "water quality" and "mortality rate eggs".
  - There is a table of allowable values which relates the water and habitat quality to the mortality rate of eggs. In general there is a P- relationship from these "qualities" to the mortality rate. Value correspondences are used to make the calculation of the resultant mortality rate.

These model fragments determine how many individuals of life stage x+1 can be produced from a number of individuals in life stage x, based on the mortality rate. In this context, mortality rate is linked to the quality and quantity of habitats available for each life stage and as such reflects the ecological concept of carrying capacity of the ecosystem.

Conclusions

This case study represents a challenging field of ecological modelling where both ecological processes and socio-economic factors are modelled in a single integrative model. One of the biggest challenges in terms of modelling the ecology is to model the recruitment and mortality processes in a way that gives the required population behaviours. One of the challenges here is to capture the true nature of transitions from one stage to another in qualitative terms through the use of informative quantity

spaces and value correspondences. Only once these can be modelled properly can the socio-economic factors be assessed, and the effects of rehabilitation activities be properly judged. If these can all be built into a single integrative model then it could prove to be an extremely useful and flexible tool for education, research and project planning, including identification of potential conflicts.

As the model is being built around some general ecological principles, it could be adapted for other rehabilitation situations and species. For example this model is also being developed for use in a case study of lowland floodplain river rehabilitation using common bream as an indicator species. Transferability of a standard model between systems may provide scope for common assessment of sustainable rehabilitation activities and comparisons between systems and different rehabilitation activities.

One of the issues of using qualitative reasoning modelling for investigations which involve a range of end-users is that to understand the outputs the user needs to understand both the domain expertise and the terminology and concepts within QRM. One of the key benefits of the standardised approach to modelling proposed by Bredeweg *et al.* (2007) is that the structured textual descriptions give the modeller a framework to use to translate domain knowledge into QR vocabulary and settings. The transition of domain concepts into a Garp3 model is not always an easy process and the construction of a causal model using QR terminology gives a useful framework to build on when the model is implemented. However, the main difficulty in developing a causal model prior to implementation is that it is not always clear whether relationships should be qualitative proportionalities (P+/P-) or direct influences (I+/I-) or indeed some form of correspondence. In such a situation it would be useful to initially develop a causal model where the relationships are described only as positive, negative and/or some form of value limitation. Once this "domain" causal model was described fully the process of developing it into a full QR causal model could be undertaken. The transition from domain causal model to QR causal model would also act as a modelling phase that identified where one quantity was acted on by more than one other quantity. This process would facilitate the determination of the most appropriate choice of I's and P's and how to combine the influences of a number of quantities.

The documentation then also allows end users to understand both the domain knowledge and how it has been represented into a QR model. Such a structured approach is also imperative for the evaluation of models from the point of view of both domain experts and other QR modellers. Within the NatureNet-Redime project a range of end-users (students, researchers etc.) will use both the textual description of the model and the final model (once fully implemented in Garp3) itself to evaluate the models domain content, QR interpretation and potential application.

Acknowledgements. This research was funded by the European Commission's Sixth Framework Programme for Research and Development (project number 004074, project acronym Naturnet-Redime - http://www.naturnet.org, see Newsletter ISSN 1801 6480).

References

Aprahamian M.W., Wyatt R.J. & Shields B.A. (2006) Use of biological reference points for the conservation of Atlantic salmon, *Salmo salar*, in the River Lune, North West England. *Fisheries Management and Ecology* **13**, 21-30.

Bredeweg B., Salles P., Bouwer A., Liem J., Nuttle T., Cioaca E., Nakova E., Noble R., Rios Caldas A.L., Uzanov Y., Varadinova E. & Zitek A. (2007, in press) Towards a Structured Approach to Building Qualitative Reasoning Models and Simulations. *Ecological Informatics*.

Cowx I.G. (1986). *The case for the re-introduction of salmon into the River Derwent*. Derbyshire County Angling Club Report 54 pp.

Cowx I.G. & O'Grady K.T. (1995a) *The reintroduction of salmon in the River Dove via the River Trent*. Report to the National River Authority Severn Trent Region Nottingham 98 pp.

Cowx I.G. & O'Grady K.T. (1995b) *Feasibility of upstream movement of salmon at Holme sluices on the River Trent at Nottingham*, Report to the National River Authority Severn Trent Region Nottingham 12 pp.

European Commission. 2001. Communication from the European Commission: a Sustainable Europe for a Better World: A European Union Strategy for Sustainable Development. Brussels, Belgium.

EU (2000) *Directive 2000/60/EC of the European Parliament and the Council of 23 October 2000 establishing a framework for community action in the field of water policy*. The European Parliament and the Council of the European Union, 72 pp.

Guerrin F. & Dumas J. (2001a) Knowledge representation and qualitative simulation of salmon redd functioning. Part I: qualitative modelling and simulation. *Biosystems* **59**, 75-84.

Guerrin F. & Dumas J. (2001b) Knowledge representation and qualitative simulation of salmon redd functioning. Part II: qualitative model of redds. *Biosystems* **59**, 85-108.

Sykes A.J.A (2004) Feasibility study for the revival of migratory salmonid stocks within the rivers of the Humber catchment. MSc Thesis, University of Hull, 141pp.

USING AGGREGATION FOR AUTOMATIC IDENTIFICATION OF WORK PROCESSES IN THE MANAGERIAL HIERARCHY

Bahadır Kaan Özütam, Cem Say
Boğaziçi University, Computer Engineering Department
İstanbul, Turkey
E-mail: ozutam@yahoo.com, say@boun.edu.tr

KEYWORDS

Aggregation, Modeling Organizations.

ABSTRACT

Aggregation is a qualitative reasoning technique which replaces repetitive cycles of process instances with a higher level description of a single continuous process. We investigate how this AI method can be used in the modeling and simulation of social organizations. Different levels of an organization such as first level employees, middle management, and top management view the available information using different degrees of abstraction. These different levels thus have different ontologies. Our approach involves starting with descriptions of low-level work processes and using the automatic abstraction mechanism to come up with higher level process descriptions employing ontologies with fewer details. The output of our aggregator provides a suggestion for how a hierarchy of tasks might be constructed in that organization.

1. INTRODUCTION

The choice of the abstraction level is a critical decision in modeling any system. A low-level, high-resolution model is more realistic than a more abstract one, but dealing with such models may be both computationally expensive, and, more importantly, cognitively inappropriate. Some abstraction is inevitable. A social organization such as a large company is a multilevel structure in which each upper level is an abstraction of the lower ones. Different levels view the available information using different "ontologies." Finding the most suitable design for such hierarchies and determining the appropriate ontology for each level is an important problem in organization theory (Daft 2001).

Computer modeling and simulation are used with increasing frequency in the study of organizations (Davis et al. 2007). Agent-based and equation based models, system dynamics, cellular automata and social network models are some of the dominant trends. Among the early significant contributions to this field, one can list the Garbage Can Model of Cohen, March and Olsen (1972), the evolutionary theory of economic change by Nelson and Winter (1982), and March's work (1991) on exploration and exploitation in organizational learning. More recent studies include the model of organizational demography and culture by Carroll and Harrison (1998), Gavetti and Levinthal's (2000) work on cognitive and experiential search, and the modeling of organization structure in unpredictable environments by Davis, Eisenhardt and Bingham (2006).

We apply the qualitative reasoning technique of aggregation of processes introduced by Weld (1986) to organizational modeling. The basic form of aggregation as explained in Weld's paper detects repeating cycles of discrete processes using the simulation history structure, and replaces them with a higher level of description, a continuous process which performs the same action as the processes in the repeating cycle. The process descriptions have two components; preconditions and changes. The preconditions define the conditions to be satisfied so that the process can be active. The changes are the effects that the process will create over other objects in the simulation when it is active.

In this ongoing work, we implemented and used the basic aggregation algorithm for automatic identification of higher-level work processes in an organization setup, improved the method so that it can be used more conveniently and effectively in this domain, and identified new aggregation types that seem to be required for organizational modeling.

2. MOTIVATION

Identifying the abstraction degrees of the information viewed by the different managerial levels, and designing the managerial hierarchy which is appropriate for a given organization are important problems in the study of organizations. The aggregation technique seems to be a promising method for these purposes. Having obtained the different level models, it is possible to simulate all of them and see how the predictions of the more abstract ones diverge from those of the lowest level. In this regard, we are interested in the following questions: Are the upper level ontologies suggested by the program indeed the ones seen in real-life multi-level organizations? Do the simulation results of the different level models differ, and if they do, how do they differ? Does this mean that one or more of the simulations are wrong? In which way do discrete and continuous simulations differ? Can we see the difference between discrete and continuous simulations when we run a simulation with aggregated, disaggregated and partially aggregated processes? We hope to gain new insight about identification of different level processes in organizations, and, indeed, about multiple-level modeling in any domain, from the answers of such questions.

3. WHAT IS AGGREGATION?

When a sequence of processes is repeated, we obtain a cycle. Aggregation is a qualitative reasoning technique which replaces such a cycle of discrete processes with a higher level of description, a continuous process which has the same effects as the processes in the repeating cycle.

Discrete processes make atomic effects over the objects they affect, and the amount of this change is known in advance. If necessary, one can repeat the discrete process multiple times until the desired total amount of change is obtained. This effect can also be realized by a single instance of a continuous process, as will be demonstrated below.

The history structure is a structure which keeps the history of the changes in the system state. Weld's (1986) algorithm of aggregation detects the repetitive cycles occurring in the history, and aggregates them. The algorithm includes three phases;

- *The repetition recognition phase* detects two similar endpoints, which means two similar instances of the same process, in the history.
- *The candidate cycle extraction phase* finds the sequences which connects these two endpoints.
- *The cycle verification phase* checks to ensure that all the process instances in the sequence can repeat.

Weld gives examples of the use of the algorithm in causal simulation to replace cycles of discrete processes with higher level continuous processes.

Let us consider an example from our organizational model to illustrate what aggregation does:

Example 1: Consider three processes to produce a Product using some Raw Material:

SeizeMachine:
Pre: the production machine is idle
Ch: reserves the machine.

ProduceHalfProduct:
Pre: the machine has been reserved,
a sufficient amount of raw material exists.
Ch: converts a specific amount of raw material to a corresponding amount of half-product.

ProduceProduct;
Pre: the machine is reserved,
a sufficient amount of half-product exists.
Ch: converts a specific amount of half-product to a finished product, and release the machine.

If we have some amount of raw material at the beginning, the sequence "SeizeMachine, ProduceHalfProduct, ProduceProduct" will repeat until almost all the raw material is transformed to products. The aggregation of a prefix of the simulation history generates a continuous process:

CP1
Pre: RawMaterial exists,
productOrder>0
The machine is idle
Ch: convert RawMaterial to Product (Assuming one unit of product is produced by one unit of Raw Material)

CP1 runs when production machine is idle and there exists raw materials. It converts all the raw material to products.

The process replacing the cycle must be a continuous process. If a discrete process is used to replace the cycle in an unsophisticated manner, the new process will just make a greater, but fixed amount of change every time it is activated. The duration and the total effect of a continuous process, on the other hand, are not determined before it runs. The effects it creates are defined in terms of change rates per unit time and the total effects of each different instance of the aggregated continuous process are different from each other in general, since they depend on the model environment.

The processes that can be abstracted by aggregation need not all be discrete. Continuous processes can also be replaced. This allows the aggregator to find nested cycles, and make multiple replacements over the same set of processes.

Let us add a new process to our model: Deliver(*n*) runs when *n* products exist, and delivers *n* products to the customer. In this new setup, CP1 (aggregated from SeizeMachine, ProduceHalfProduct, and ProduceProduct, as before,) does not run alone. An instance of Deliver(*n*) runs each time a new batch of *n* products have been produced. The sequence "CP1, Deliver" is repeated until all the raw material is converted to products and delivered. In this case, the aggregator finds a new cycle "CP1, Deliver" and replaces it with a new continuous process CP2, which converts the raw materials to products and delivers them. CP2 represents the operations of a Production Department in this organization.

4. USING AGGREGATION IN ORGANIZATIONAL MODELING

Different levels of an organization such as first level employees, middle management, and top management are actually different abstraction levels, and the information requirements of each level differ from the others, depending on their interests. So the focus of the modeler must be the different levels of an organization and the ways these levels view things. The different levels have different ontologies. In the higher levels, some groups of processes of the lower levels are represented with more abstract processes, some low level decisions are not seen, but only the cumulative results of these decisions are visible, some variables of the lower levels are not represented, and individuals are viewed collectively as groups having aggregated characteristics. A consequence of the point of view that the higher level views are basically summaries of the more detailed lower levels is that we can obtain the whole model of an organization by repeated abstractions once we construct

PurchaseRawMaterial
Pre: *RawMaterialAlternatives exists*
Ch: *add RawMaterial, decrease rawMaterialorder, increase accounts_payable, remove RawMaterialAlternatives*

FindRawMaterials
Pre: *rawMaterialorder>0*
Ch: *add RawMaterialAlternatives*

SearchForCustomer
Pre: *marketingBudget>0*
Ch: *add RequestingCustomers*

NegotiateWithCustomer
Pre: *RequestingCustomers exists*
Ch: *remove RequestingCustomers, add CommittedRequests, decrease marketingBudget*

AcceptBestOrder
Pre: *CommittedRequests exists*
Ch: *increase productionOrder, increase rawMaterialOrder, remove CommittedRequests*

SeizeMachine
Pre: *machine is idle, productionOrder>0, RawMaterial exists*
Ch: *machine is reserved*

ProduceHalfProducts
Pre: *machine is reserved, RawMaterial exists, HalfProduct doesn't exist*
Ch: *add HalfProductStock, remove RawMaterial*

ProduceProducts
Pre: *HalfProduct exists, machine is reserved*
Ch: *remove HalfProduct, add Product, idle the machine*

Deliver
Pre: *at least N Product exists, productionOrder >= N*
Ch: *remove N Products, decrease productionOrder by N, increase accounts_receivable*

CollectMoney
Pre: *accounts_receivable>0*
Ch: *decrease accounts_receivable, increase money*

PayMoney:
Pre: *accounts_payable>0, money>0*
Ch: *decrease accounts_payable, decrease money*

Research
Pre: *money>0, researchBudget>0*
Ch: *increase researchDone*

ProposeNewProduct
Pre: *researchDone > X*
Ch: *add ProductProposed*

N = delivery amount, assumed constant. X = research necessary for a new product, assumed constant.
Y = research necessary for finding a new technology to use, assumed constant.
Variable names start with lower case (i.e. money), object names start with upper case (i.e. RawMaterial)

Figure 1 : The Processes in the Organization Model

the lowest level of the model. Our aim is to realize this kind of modeling by an automatic aggregation process, which takes as its input a hand-made description of the lowest level.

The following working example illustrates the use of the aggregator described in the previous section to find such "natural" abstractions of the lowest level work processes. The higher level process descriptions output by the aggregator can be seen as a suggestion about how a hierarchy of tasks might be constructed in an organization with these low-level processes.

Example 2: Consider a production company, whose lowest-level processes are defined as shown in Figure 1. The model is simulated with a scenario in which a customer is found and a production order of a certain amount is received from that customer. Taking the simulation history as input, the aggregator gave the replacements in the model shown in Figure 2.

An examination of the processes resulting after the second-level aggregation leads one to say that this organization can have five departments; CP1 = Purchasing, CP6 = Marketing, CP7 = Production, CP4 = Accounting, CP5 = Research and Development. Note that the aggregator output is not just a suggestion of a possible organizational structure. The descriptions of the high-level processes also indicate which ontologies, i.e. subsets of the set of objects and variables used in the lowest level model, are used by the respective "managers" of the departments when they talk to their own superiors. For instance, the object type "Half-Product" is not mentioned when the Production Department's manager communicates with top management.

5. NEED FOR FURTHER AGGREGATION METHODS

We took Weld's algorithm (Weld 1986) as a starting point in constructing the aggregator whose runs were exemplified in Sections 3 and 4. Even in the simple case of Example 2, Weld's algorithm has problems in identifying cycles, since multiple "departments" are supposed to run in parallel. The improvements we incorporated to the algorithm to handle such cases are explained in Section 5.1.

The full realization of the goal of automatic identification of work processes in the managerial hierarchy requires more than even the improved version of the cycle aggregation method we considered until now. Consider Example 2. The highest aggregated level, which is the furthest one can go and obtain sensible results using this method, includes five continuous processes. This is not the most abstract level one would hope to obtain. Imagine the owner of the company, who is not concerned with the internal operational details of the company, but is only interested in the performance measures of the company, rather than how they are achieved. There must be a highest level process which defines the entire organization in terms of the changes it caused, and maybe more levels above the output of Example 2 under this top level. We therefore need further methods of aggregation to complete the construction of the managerial hierarchy of the organization. One such method that we examined is explained in Section 5.2, including a working example.

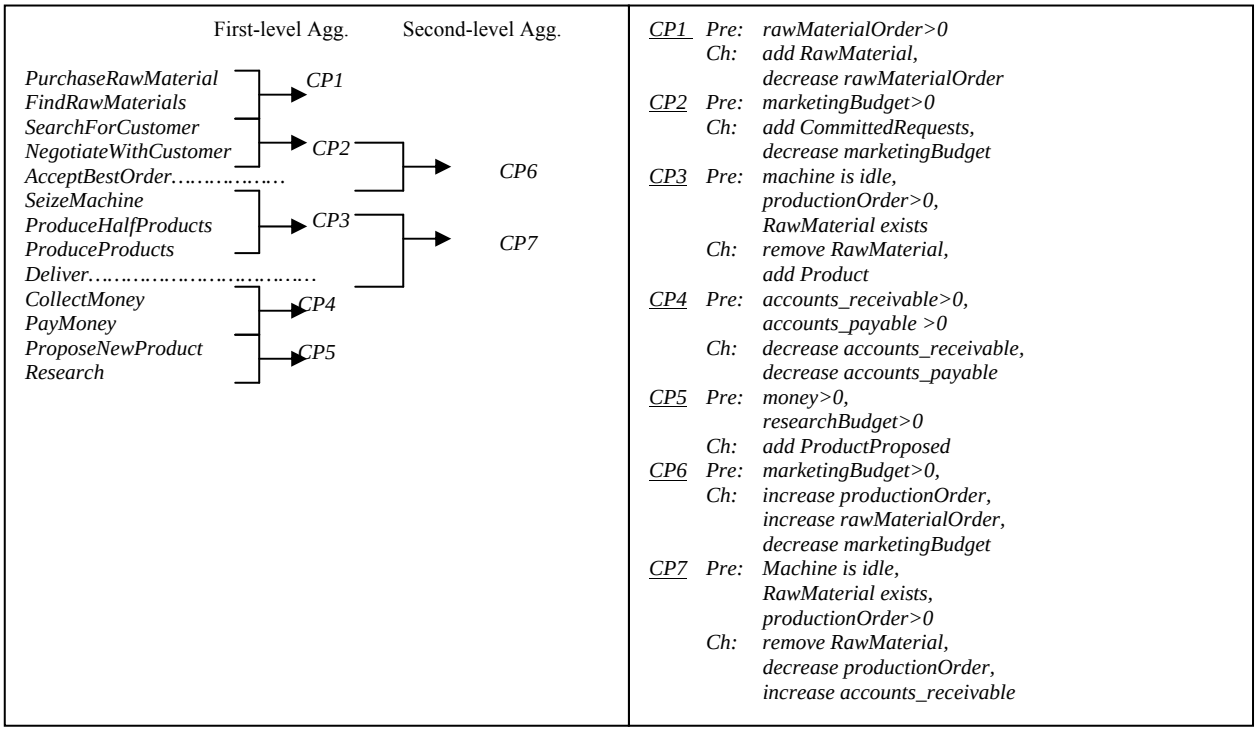


Figure 2 : First and Second Level Aggregation of the processes in the organization model

Other, as yet unimplemented, ideas that we plan to look at, are explained in Section 6.

5.1. Improvements in the algorithm

In many cases, Weld's algorithm may detect multiple alternative cycles as aggregation candidates in the simulation history. To be able to prefer better alternatives, the aggregator can be improved to choose the cycle according to certain principles. A company has several departments. Usually, mutually irrelevant tasks take place in different departments. Descriptions of such mutually irrelevant process instances which run in the same time will be printed out close to each other in the output of a sequential simulation (like the ones carried out by our process simulator) of the entire company, such that they may appear as candidates for being aggregated together to a naïve algorithm.

Consider the sequence CollectMoney, PayMoney, CollectMoney, PayMoney, … and a separate sequence FindRawMaterials, PurchaseRawMaterial, FindRawMaterials, PurchaseRawMaterial, … Although the two departments perform mutually irrelevant tasks, they operate simultaneously. When the aggregator analyses the history, it may find several repetitions of the sequence "CollectMoney, FindRawMaterials, PayMoney, PurchaseRawMaterial", and try to replace this "cycle" of four with a single continuous process, and this is clearly not what we want it to do. (This predicament of distinguishing irrelevant processes is referred to as the local evolution problem (Forbus 1993)). How can the aggregator distinguish these mutually irrelevant processes? Let us define what we mean by "mutually irrelevant" in terms of the aggregator's input.

Definition : A process A is a *change predecessor* of a process B if at least one of the changes of A affects a variable referenced in the preconditions of B.

Definition : Two processes are *mutually irrelevant* if neither of them is a change predecessor of the other one.

(Note the similarity of our definition of change predecessor with that of the notion of threats in nonlinear planning (Russell and Norvig 1995)).

The main idea is that the processes in a cycle must trigger each other so that they form a meaningful cycle. Irrelevant processes may occur successively in time by chance, in which case they do not form a meaningful cycle.

What if a nonzero duration of time passes before the start of B after its change predecessor A terminates? Can we still say that A triggers B, and that they are good candidates for a cycle? If A triggers B, changing a variable which is referenced in the preconditions of B, B must be activated just when A changes that variable. If there is a gap between these two events, there must be another event occurring in this gap which triggers B. Thus, if this event had not occurred, B would not be active. A does not trigger B, and they are not good candidates for a cycle.

In our simulator, time proceeds in ticks, and the amount of time between two successive time ticks is the unit time (smallest amount of time that must be considered)

```
ProduceNewProduct
      Pre:  RawMaterial exists,
            newProductOrder > 0
      Ch:   convert RawMaterial to  NewProduct
DeliverNewProduct
      Pre:  at least N NewProduct exists,
            newProductOrder >= N
      Ch:   remove N NewProducts,
            decrease  newProductionOrder by N,
            increase  accounts_receivable
N = delivery amount
```

Figure 3 : The processes added in Example 3

for the simulation. The starting and ending times of processes correspond to starting and ending of time units since there can be no smaller time in the simulation (A discrete process lasts one tick). Thus, if a process triggers another process, the triggered process must start just at the tick at which it is triggered.

Definition : An instance of a process A is a *time predecessor* of an instance of another process B, if the instance of B starts just when the instance of A terminates.

If the aggregator only accepts the instances of processes which trigger each other, it will avoid including irrelevant processes appearing in one cycle and will form meaningful cycles.

Definition : An instance A_i of a process A *triggers* an instance B_i of another process B, if A is a change predecessor of B, A_i is a time predecessor of B_i and there doesn't exist a process instance C_i that undoes the effects of A_i on the variables referenced in the preconditions of B_i.

Must the last process of a cycle trigger the first process of the cycle? Considering the meaning of the word "cycle," one is tempted to say "yes". But consider the case where we produce products from raw materials. The process will continue as long as we have raw materials and a nonzero production order, despite the fact that the act of finishing a product does not trigger the start of another production. Thus, we have decided to accept sequences in which every process except the first is triggered by the previous one as valid cycle iterations, even when the last process of such a sequence does not trigger the first process. Example 2 was run on our implementation of an improved aggregator in which these augmented criteria for cycle detection enabled the identification of the sensible hierarchy seen in Figure 2.

5.2. Superclass Aggregation

The aggregator described above recognizes a cycle only if multiple instances of the same process are seen to be repeating. In some cases, abstraction of two non-identical but sufficiently similar process descriptions to a "superclass" process may help. Consider the sequence given below for a cleaner who is working in a building:

Cleaning the floor,
Walking down the stairs one floor,
Cleaning the floor,
Going down one floor by elevator,
Cleaning the floor, …

No proper repetition can be found. But if we abstract "Walking down the stairs one floor" and "Going down one floor by elevator", to the new process "Going down one floor", we can obtain the new sequence

Cleaning the floor,
Going down one floor,
Cleaning the floor,
Going down one floor,
Cleaning the floor, …
in which a repetition can be detected easily.

Forbus and Falkenhainer give a good example of how to compare processes to find out their similarity in their work on analogical processing with the Structure-Mapping Engine (Falkenhainer et al. 1990). The idea is to accept two processes as subclasses of a superclass if a significant proportion, rather than all, of their properties are identical. This operation can of course create problems if the non-identical features of the low-level processes, which are abstracted away, play important and different roles in the actual system. This method of abstraction can therefore sometimes produce incorrect higher-level models. The allowed error can in fact be tuned by the user, since the aggregation algorithm decides whether to create a superclass process for two given low-level processes according to a similarity function which measures the match between two processes. Processes whose similarity degree exceeds a user-defined constant are aggregated. Keeping the required similarity degree high will avoid the kind of error explained above, with the cost of a narrow scope for aggregation. Keeping it low will allow more processes to be aggregated together, and the divergence of the higher-level model predictions from the lowest level ones will increase. Upon receiving the input set of process definitions before beginning the simulation, the aggregator first tries to identify superclass processes among these. Simulation with the aim of cycle detection is then performed with the updated process list. Since common superclasses can be detected even among the newly abstracted processes during the higher-level aggregations, the aggregator runs this similarity detection procedure again whenever new process descriptions are added to the list.

Example 3: Consider adding two new processes which are similar to some already present processes to the model of Example 2: The company has obtained a new machine of type "NewMachine", which uses the same type and amount of raw materials, but produces a different product of type "NewProduct" directly, without going through the half-product stage. The new processes are shown in Figure 3.

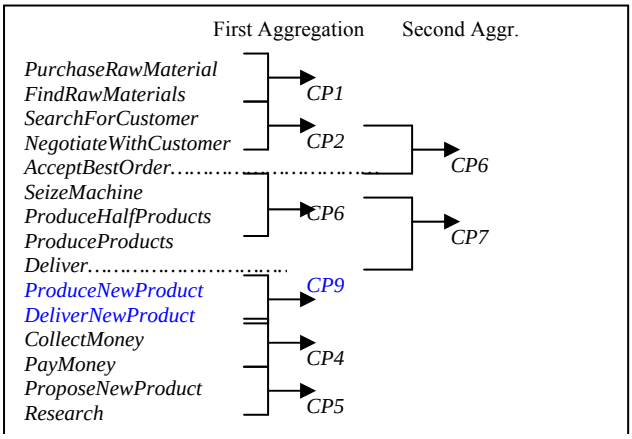


Figure 4 : Aggregation in Example 3 when the required similarity degree is 0.9

We submitted this model to the aggregator with a scenario of receiving orders for Product and NewProduct randomly with the same probability. Two different runs, with required similarity degrees of 0.9 and 0.6, respectively, were performed.

In the first run with a required similarity degree of 0.9, shown in Figure 4, the cycles the aggregator replaced in the first-level aggregation were the same as in Example 2, and ProduceNewProduct and DeliverNewProduct were also aggregated to a new process. The second-level aggregation had the same results as in Example 2, and the new production process was not aggregated any further.

In the second run, with a required similarity degree of 0.6, the aggregator first found out that Deliver and DeliverNewProduct are similar processes. Similarly, after generating CP3 from "SeizeMachine, ProduceHalfProduct, ProduceProduct", it also recognized CP3 and ProduceNewProduct as similar processes. Consequently, the program made the aggregation shown in Figure 5, with the process definitions given in Figure 1 and the production "department" process of the organization came out to be isomorphic to the one in the Example 2, (Figure 2)

As seen in Figure 5, the Deliver and DeliverNewProduct processes were replaced by the more general process CP10 and ProduceNewProduct and the new process CP3, which was produced by the aggregation of SeizeMachine, ProduceHalfProduct and ProduceProduct were replaced by the more general process CP11.

The aggregator found the similarity of Deliver and DeliverNewProduct to be 0.67, and the similarity of CP3 and ProduceNewProduct to be 0.72, causing both pairs not to be abstracted together in the first run, and to be abstracted together in the second run. Both results may be preferable depending on the requirements of the context. When analysis considering the sales of the products individually is required, seeing the products as

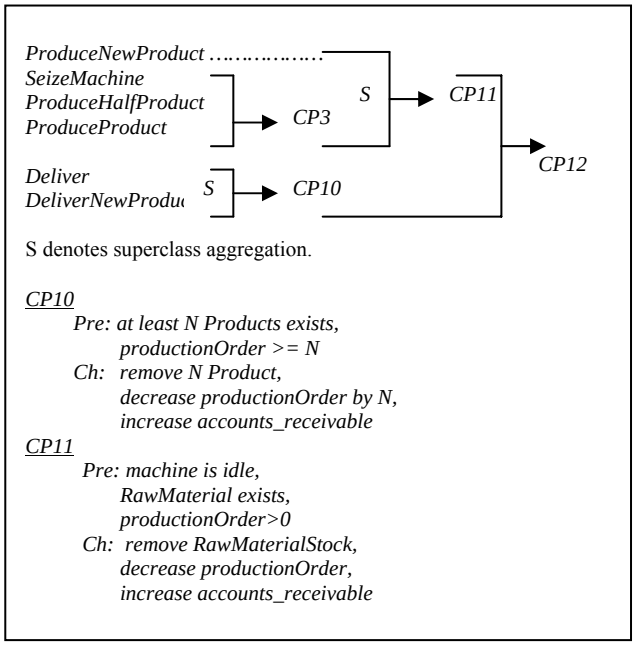


Figure 5 : Aggregation in Example 3 when the required similarity degree is 0.6

separate units and getting the results accordingly will help. In another case, for example, when the total sales of the company, or only the total amount of money made by the company is required, seeing the products as equivalent and looking only at the totals will help.

6. FUTURE WORK

More tools are necessary for a complete and sensible identification of and reasoning about the managerial hierarchy of an organization. In this ongoing study, we plan to implement the following additional methods.

6.1. Parallel Aggregation

Weld (Weld 1986) defines parallel aggregation as "the replacement of multiple instances of the same process occurring simultaneously", without including the working principles or any example. This idea promises to serve as wide a range of possibilities as serial aggregation does, so we plan to realize it. While serial aggregation makes a vertical replacement in the history of the simulation since it replaces instances spread over time, parallel aggregation makes a horizontal replacement, since it replaces process instances occurring in the same time, but spreading over the actors of the simulation.

In parallel aggregation, as well as the starting time, we know the ending time since only one iteration occurs. But we do not know the total effects, since we do not know how many parallel processes will participate in this parallel processing in different moments of the simulation. The definition of parallel aggregation requires defining a new type of process, since neither the discrete nor the continuous process definition formats we use meet the requirements of the process generated as a

result of a parallel aggregation. This new type of process occurs atomically, and its termination time is determined when it starts. But unlike discrete processes, its total effects can not be known before simulation, since it depends on the number of parallel process instances participating which can be different in different moments of the simulation. As a result of this, the total effects of the process change in different instances.

6.2. Reverse Aggregation

Consider the case where we are given the abstract description of an organization, or some components of the organization, and we attempt to obtain the descriptions of the lower level processes. For instance, we may try to guess the invisible structure of a rival company from the partial information that is available about that company. What we need to do is to "reverse aggregate" the high-level models.

At first sight, reverse aggregation seems to be a hopeless task, since it involves creating a more detailed description than its input. But if we also have a library of common lowest-level process descriptions, we might at least guess a candidate input process list which would, when aggregated, result in the higher-level models that have been given to us.

6.3. Mixed Levels of Abstraction

Abstraction is meaningful if we do not need the detailed information we lost, or we have the ability to reconstruct it in case we need it later. Consider a manager who manages several working teams including several workers. The manager would not care about the workers individually, and would not want to know the details of each worker. In his daily operation, he would rather know the output of the teams, leaving the details of what happens within the teams to team leaders. The point of view of this manager is an aggregated level, which hides the first level employees inside teams. But one day, in case a problem about an individual first level employee occurs and affects the operation of the team, or even the whole organization, the manager would want to understand the case and need the details about the operation of the first level employee. This would require the manager to adopt, for the purposes of this case, a "mixed-level" model, which does not necessarily include all the complexity of the lower level processes which are irrelevant in this case, but which contains the details necessary for the present reasoning task. We believe that the infrastructure we are preparing for the application of the methods described above can easily be adapted for examining such mixed-level modeling tasks.

6.4 Aggregation of Probabilistic Models

One of the simplistic aspects of Example 2 is that all processes are deterministic; e.g. once its preconditions are satisfied, ProduceProducts does its job, assuming that the machine will never fail. This, of course, is unrealistic. We plan to consider an alternative process format which supports probabilistic models like one where the machine in example 3 can be stipulated to have a specific fault probability, and the results of such eventualities can be described separately in the process descriptions. An aggregation algorithm for such probabilistic models may produce higher-level models which are themselves probabilistic.

7. CONCLUSION

The hierarchical structure of an organization such as a large company provides a suitable domain for investigating aggregation, since the hierarchical levels composed of various positions (first level employee, middle management, CEO, etc.) in the organization correspond to different aggregation levels. Our experiments so far with our improved version of Weld's aggregator have helped us identify some other tools that are necessary for reasoning about these issues in this domain. We are actively working on the design and development of these tools.

REFERENCES

Carroll, G. and Harrison, J. R. 1998. "Organizational Demography and Culture: Insights from a Formal Model and Simulation", *Administrative Science Quarterly*, 43: 637-667.

Cohen, M. D., March, J., and Olsen, J. P. A. 1972. "Garbage Can Model of Organizational Choice", *Administrative Science Quarterly*, 17(1): 1-25.

Daft R. L. 2001. *Essentials of Organization Theory and Design,* Ohio: South-Western College.

Davis, J., Eisenhardt, K. and Bingham, C. 2006. "Complexity Theory, Market Dynamism, and the Strategy of Simple Rules", Stanford Technology Ventures Program - Working Paper.

Davis, J., Eisenhardt, K. and Bingham, C. 2007. "Developing Theory Through Simulation Methods", *Academy of Management Review,* 32(2): 480-499

Falkenhainer, B., Forbus, Kenneth D. and Gentner, D. 1990. "The structure-mapping engine: algorithm and examples", *Artificial Intelligence,* 41: 1-63.

Forbus, K. D. 1993. "Qualitative Process Theory", *Artificial Intelligence*, 59: 115-123.

Gavetti, G., and Levinthal, D. 2000. "Looking Forward and Looking Backward: Cognitive and Experiential Search", *Administrative Science Quarterly*, 45: 113-137.

March, J. G. 1991. "Exploration and Exploitation in Organizational Learning", *Organization Science*, 2(1): 71-87.

Nelson, R. R. and Winter, S. G. 1982. *An Evolutionary Theory of Economic Change*, Cambridge, Massachusetts: Belknap - Harvard University Press.

Prietula, Michael J., Kathleen M. C. and Gasser, L. (ed) 1998., *Simulating Organizations: Computational Models of Institutions and Groups,* AAAI Press /MIT Press.

Russell, S. and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach,* New Jersey: Prentice Hall. Inc.

Weld, D. S. 1986. "The Use of Aggregation in Causal Simulation", *Artificial Intelligence*, 30: 1-17.

Advanced Experiments for Learning Qualitative Compartment Models

Wei Pang and **George M. Coghill**
Department of Computing Science
University of Aberdeen
Scotland, United Kingdom, AB24 3UE

Abstract

In this paper, the learning of qualitative two-compartment metabolic models is studied under the conditions of different types and numbers of hidden variables. For each condition, all the experiments, each of which takes one of the subsets of the complete qualitative states as training data, are tested one by one. In order to conduct the experiments more efficiently, a backtracking algorithm with forward checking is introduced to search out all the well-posed qualitative models as candidate solutions. Then these candidate solutions are verified by a fuzzy qualitative engine JMorven to find the target models. Finally the learning reliability and kernel set under different conditions is calculated and analyzed.

1. Introduction

Qualitative Model Learning (QML), as a branch of system identification, plays an important role in the fields of biology and physics. It involves extracting the qualitative structures (namely Qualitative Differential Equations, QDEs) of systems from given qualitative data, which are often incomplete and imprecise. So it can be viewed as the inverse of Qualitative Simulation (such as QSIM (Kuipers 1994)).

Some related research in this field has been done during the last two decades, such as GENMODEL (Hau & Coiera 1993), MISQ (Richards, Kraan, & Kuipers 1992), QSI (Say & Kuru 1996) and more recently, QSI-ILP (Coghill *et al.* 2004). All these systems are based on QSIM representation. However, the above systems have different limitations. GENMODEL can not introduce hidden variables and perform dimensional analysis. QSI often generate over-constrained models. None of these systems except QSI-ILP performs systematical experiments which include conditions of all the subsets of complete data. None of these systems have analyzed the influence of different hidden variables on the learning reliability of the system.

2. Model Representation

2.1 JMorven

In this paper, a more flexible qualitative reasoning engine, JMorven (Bruce & Coghill 2005), is used to represent and verify qualitative models. JMorven, a Java implementation of the Morven framework (Coghill 1996), possesses all the benefits of QSIM and introduces many new features. The introduction of differential planes (Wiegand 1991) and vector envisionment (Morgan 1988) make it possible to reason about more than two derivatives. By introducing fuzzy theory, JMorven uses fuzzy quantity spaces to specify the variables, and can perform fuzzy vector envisionment (Coghill 1996), which enables it to deal with fuzzy qualitative data. In addition, the utilization of parallel techniques makes JMorven more efficient. All the above advantages make JMorven a better choice as a model representation and verification component in our work.

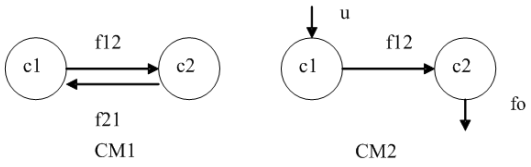


Figure 1: Two-Compartment Metabolic Models

QDE	JMorven Differential Plane 0
f12=M+(c1)	func (dt 0 f12) (dt 0 c1)
fo=M+(c2)	func (dt 0 fo) (dt 0 c2)
q1=u - f12	sub(dt 0 q1)(dt 0 u)(dt 0 f12)
q2=f12-fo	sub(dt 0 q2)(dt 0 f12)(dt 0 fo)
c1'=M+(q1)	func(dt1 c1) (dt0 q1)
c2'=M+(q2)	func(dt1 c2) (dt0 q2)

Table 1: QDE and JMorven Description for CM2

2.2 Compartmental Models of Metabolic Systems

Metabolic systems are often modeled by two-compartment models (See Figure 1). In the two-compartment model, if input *u* and output *fo* do not exist, the model becomes a coupled closed system, denoted as model CM1 in this paper. CM2 is defined in a similar way. Table 1 shows QDE and JMorven representation (0th differential plane) for CM2.

The "func" symbol in Table 1 denotes the Function con-

Increase Mappings	
neg	neg
zer	zer
pos	pos

Decrease Mappings	
neg	pos
zer	zer
pos	neg

Table 2: Increase and Decrease mappings

Quantity Name	a	b	alpha	beta
neg	-100	-1	0.5	0.5
zer	-1	1	0.5	0.5
pos	1	100	0.5	0.5

Table 3: Quantity Space

straint in JMorven. JMorven extends the M+ and M- constraint in QSIM by introducing a more general function constraint, in which two variables can have arbitrary mappings.

In order to simplify the problem and compare our work with previous research, some assumptions, similar to those in (Coghill *et al.* 2004), are imposed upon the models. That is, the compartment models in our work are linear systems with constant coefficients in the functional relationships. The fuzzy quantity space of any variable includes only three values: negative, zero and positive. One reasonable quantity space is shown in Table 3. The meanings of the variables "a","b","alpha" and "beta" are described in (Shen & Leitch 1993). For all the observed variables, only the zero derivative (magnitude) and the first derivative can be measured qualitatively. All the function relationships have the corresponding value (zer, zer), and there are only two kinds of function mappings as shown in Table 2. The models can be causally ordered, and are in canonical form as described in (Iwasaki & Simon 1986). For simplicity and clarity, the rest of this paper will refer to *Inc* and *Dec* as the function constraints which have the increase and decrease mappings in Table 2.

2.3 One and a Half Differential Plane

The concept of *differential plane* in qualitative context was first proposed in (Wiegand 1991). The zeroth differential plane constains the constraints, which can construct a model used for numerical simulation. The constraints in a higher differential plane are obtained by differentiating the corresponding constraints in the preceding differential plane.

Based on the assumptions in previous section, a JMorven representation with "one and a half" differential planes is adopted to represent the models. Here "one and a half" means only the constraints in the 0th differential plane and part of the constraints in the 1st differential plane can be used to represent the model. In the 1st differential plane, the constraints which contain the 2nd derivative of a variable can not be used, because only the information about zero and first derivative of a variable are available. This form is equivalent to QSIM description for the purpose of comparison. Notice that M+ and M- are implemented by two function constraints in different differential planes: the corresponding values can be obtained from the mappings of the corresponding function constraint in the 0th differential plane, and the function constraints in the 1st differential plane determine the monotonically increasing or decreasing relation between two variables.

Differential Plane 0
C1: Inc (dt 0 f12)(dt 0 c1)
C2: Inc (dt 0 fo)(dt 0 c2)
C3: sub (dt 0 q1)(dt 0 u) (dt 0 f12)
C4: sub (dt 0 q2)(dt 0 f12)(dt 0 fo)
C5: Inc (dt 1 c1)(dt0 q1)
C6: Inc (dt 1 c2)(dt0 q2)

Differential Plane 1
C7: Inc (dt 1 f12)(dt 1 c1)
C8: Inc (dt 1 fo) (dt 1 c2)
C9: sub(dt 1 q1)(dt 1 u)(dt 1 f12)
C10:sub(dt 1 q2)(dt 1 f12)(dt 1 fo)

Table 4: JMorven Model for CM2

For example, the CM2 model is described in Table 4. In this description, Constraint C1 in the 0th differential plane and C7 in the 1st differential plane are equivalent to the constraint M+(c1 f12) in QSIM (Note the position difference). The following two constraints in the 1st differential plane are abandoned for the above mentioned reason:

C11: Inc (dt 2 c1)(dt1 q1)
C12: Inc (dt 2 c2)(dt1 q2)

3. Background Knowledge

Before introducing the algorithm, some preliminary knowledge has to be described in detail. The constraints involved in this section are only in the 0th differential plane.

3.0 Some Concepts about Qualitative Modeling

The state variables in a causally ordered system are the variables that are directly effected by the integration operation, and is usually the output of the integrator.(Wiegand 1991) Simply speaking, in a model with canonical form, only the state variable can have first derivative. The magnitude of a state variable can not appear on the left side of any equation in the model. The exogenous variables are those variables determined from outside the model. All the non-exogenous variables are also called system variables.

In an experiment, the hidden variables are the unmeasured variables which lose both range and dimensional information. The number of hidden variables is often unknown, but it is reasonable to specify a maximum number of possible hidden variables. If the maximum number is less than the number of actual hidden variables, only "shallow" models will be induced; otherwise, unnecessarilly "deep" models may be found.

The model size in this paper is referred to as the number of the constraints in the model. The specification of model

size is another factor that can influence the learning of the models.

3.1 Inconsistent Constraints

An inconsistent constraint is a constraint that is inconsistent with the training data and consequently fails to pass the consistency check. The consistency check module we employed here is the same as the one in JMorven, which uses the fuzzy interval algebraic operations. For example, constraint *X=Y-Z* is an inconsistent constraint when current training data include the following qualitative state: (X,Y,Z)=(pos, neg, pos),The quantities of "pos" and "neg" are taken from table 3.

3.2 Conflict Constraints

Two qualitative constraints *C1* and *C2* are conflicting if they are logically or dimensionally inconsistent, or redundant so that a simpler qualitative constraint can be derived from these two constraints by algebraic operations. We use $C1 \bowtie C2$ to represent that *C1* and *C2* are conflicting. The details about conflict constraints will be illustrated by section 3.2.1 $\sim$ 3.2.4.

3.2.1 Conflict between two function constraints

Two function constraints *C1* and *C2* are conflicting if they satisfy any of the following two conditions:

- a. Logical Conflict: They have the same variable/derivatives in the corresponding positions of the constraints but they have different function mappings, i.e., one is *Inc* and the other is *Dec*
- b. Redundancy: They have the same variable/derivatives and same function mappings but the variables/ derivatives appear in different positions of the constraints.

In condition a, these two constraints are actually logically inconsistent, because in a physical or biological system, the relation between two variables can not have different mappings. For example, the following two constraints are conflicting:

C3.1: Inc (dt 1 X) (dt0 Y)
C3.2: Dec (dt 1 X) (dt0 Y)

C3.1 means the first derivative of variable X and zero derivative of Y has increasing relationship, but C3.2 means these two variables has decreasing relation, this is contradictory in logic.

In condition b, these two constraints are in fact the same constraint. Considering the following two constraints:

C3.3: Inc (dt 0 X) (dt0 Y)
C3.4: Inc (dt 0 Y) (dt0 X)

C3.3 and C3.4 actually describe the same relation if the causal ordering is ignored. It will be redundant if both of them appear in one model, and also the system cannot be causally ordered. So C3.3 and C3.4 are conflicting constraints.

3.2.2 Conflict between subtract constraints

The detection of a conflict between two subtract constraints is more complicated than that in function constraints. After generalization, the following seven conditions are listed without considering the dimensional consistency:

a: a=b-c , c=b-x (x can be any variables in the system)
b: a=b-c, d=b-c (a, d can be any variables in the system)
c: a=b-c, d=c-b (a, d can be any variables in the system)
d: a=b-c, b=a-x (x can be any variables in the system)
e: a=b-c , c=a-b
f: a=b-c , b=c-x (x can be any variables in the system)
g: a=b-c , c=x-a (x can be any variables in the system)

The constraints in the above conditions are either contradictory or can be replaced by a simpler constraint.

Apart from the above conditions, the dimensional conflict may occur between two subtract constraints when there exist variables with undefined dimension, such as hidden variables. The following condition is an instance of dimensional conflict:

h: Hid0=a-b, c=Hid0-d

Suppose both of these two constraints are dimensionally consistent individually, and the dimension of a and b is different from that of c and d. Hid0 is a hidden variable with undefined dimension. The conflict occurs because Hid0 can only have one dimension, either the same as a and b, or c and d.

3.2.3 Conflict Set of a Constraint

After the preprocessing phase of the algorithm, which we will introduce later, a candidate constraint set is obtained, denoted as ***FCS***. The conflict set for a constraint C1 is defined as:

$$ConflictSet(C1) = \{C_i | C_i \in \boldsymbol{FCS}, C1 \bowtie C_i\}$$

As we have introduced before, this conflict relation is binary which only involves two constraint.

3.2.4 Conflict involved more than two constraints

The conflict may involve more than two constraints, for example,

Inc(Hid0, Hid1), Hid0=a-b, a=Hid1-d.

Here the hidden variables Hid0 and Hid1 have the same dimension derived from the second and third constraint, resulting in no physical meaning for the first function constraint. Because the corresponding equation for the first constraint is:

Hid0= k* Hid1

In this equation, k must have a dimension if we make the assumption that there is no gain or amplifier in the system under study. So the dimension of Hid0 and Hid1 can not be the same.

3.3 Defining Constraints and Search Space Partition

3.3.1 Defining Constraint

The defining constraint for a variable with specified derivative (or variable/derivative for short) is the constraint in which the variable/derivative appears in the leftmost position.

For instance, constraint sub (dt1 X) (dt0 Y) (dt0 Z) is one defining constraint for the first derivative of variable X. All derivatives of an exogenous variable and zero derivative of a state variable do not have defining constraints.

3.3.2 Referring Constraint

The referring constraint of a variable/derivative is the constraint in which the variable/derivative appears in any position except the leftmost position.

For example, Sub(dt0 Y)(dt0 X)(dt0 Z) is a referring constraint for both zero derivative of variable X and zero derivative of variable Z.

3.3.3 Dependency Set of a Constraint

For a certain variable/derivative, all its referring constraints depend on its defining constraints in causal ordering context. If constraint C1 depend on C2, then this relation is denoted as follows:

$$C1 \rightarrow C2$$

Suppose the candidate constraint set is ***FCS***, the dependency set for a constraint C1 is defined as:

$$Dependency(C1) = \{C_i | C_i \in \textbf{\textit{FCS}}, C1 \rightarrow C_i\}$$

For example, constraint sub (dt0 X)(dt0 Y)(dt0 Z), the dependency set of this constraint may contain the following constraints:

Inc (dt0 Y)(dt0 A)

Dec (dt0 Z)(dt0 B)

In a causally ordered model, a constraint can not appear before any of its dependency constraints, because only after the defining constraint of a variable/derivative appears, can other constraints refers to this variable/derivative.

Theorem 3.1

Based on all the assumptions we have made upon the models, in the 0th differential plane, a well-posed model defined in (Coghill *et al.* 2004) must include one and only one defining constraint for each of the system variables (zero or first derivative).

Proof: Suppose X is a non-exogenous variable in the model. If X is a state variable, according to the definition of state variable, there must be a defining constraint for the first derivative of X. If X is not a state variable, and the model does not include any defining constraint for the zero derivative of X, then no referring constraints for X can be included in the model, resulting in the exclusion of X from the model. This is contradictory considering the completeness principle of well-posed models, stating that the model must include all the system variables. So a well-posed model must include at least one defining constraint for each of the system variables.

On the other hand, if a model includes more than one defining constraint for the same variable, it also can not be causally ordered. Consequently Theorem 3.1 is sound.

Corollary 3.1

The model size of a target model equals to the number of system variables (including hidden variables) in the model.

4. Algorithm Description

First we introduce the preprocessing phase of the algorithm, this includes four modules: Constraint Generation, Constraint Filtering, Pre-Calculation and Constraint Set Partition.

4.1 Constraint Generation

Constraint generation is similar to GENMODEL (Hau & Coiera 1993) except that it performs an additional dimensional check(Bhaskhar & Nigam 1990). In this phase, given all the observed variables, maximum number of possible hidden variables, maximum number of derivatives for each variable (2 in our problem), range and dimension (if available) for each derivative, and all possible constraint types (Subtract, Inc and Dec in this paper), the constraint generator will generate all the possible constraints, denoted as Initial Candidate Constraint Set (***ICCS***).

4.2 Constraint Filtering

Second, all the constraints in ***ICCS*** will be checked for consistency by the constraint filter. The inconsistent constraints defined in Section 3.1 will be filtered out. After this phase, a filtered constraint set (***FCS***) is obtained. Given complete behaviors of the systems, ***FCS*** will have the minimum size; otherwise, the size of ***FCS*** may be very large.

4.3 Calculation of Conflict Set and Dependency Set

In this phase, for each constraint in ***FCS***, we calculate the conflict set (Section 3.2.3) and dependency set (Section 3.3.3) and store the result into two matrixes: *ConflictMatrix* and *DependencyMatrix*. They will be used for later backtracking search algorithm.

4.4 Constraint Set Partition

FCS is divided into several subsets, each of these subsets contains all the defining constraints for the same variable. If S_i is the subset containing the defining constraints for a hidden variable, an "empty" constraint ϕ is appended on this subset: $S_i = S_i \bigcup \{\phi\}$. ***DS*** is a set that takes each of these subsets as an element, denoted as ***DS***= $\{S_n\}$ (n=1 to N) N is the number of variables (including hidden variables). For any two elements in ***DS***, $|Si| \leq |Sj|$ if $i \leq j$. For example, in the CM2 model, a subset for variable f12 may contain the following constraints:

Inc(dt0 f12) (dt0 c1)
Dec(dt1 f12) (dt0 c2)
Sub (dt0 f12) (dt0 fo)(dt0 u)

4.5 Backtracking Algorithm

The basic idea of the algorithm is for each subset S_i in ***DS***, selecting only one constraint, thus to construct a model, then checking the validity of this model. The correctness of this selection is guaranteed by Theorem 3.1.

For efficiency reasons, a backtracking tree search algorithm is adopted. The algorithm continuously adds constraints from different subsets in ***DS***, once a new constraint from a subset is added, the current partial model will be checked for validity by model checking algorithms. If this partial model fails to pass the check, we will abandon it, backtrack to previous node, and select the next node, in order to avoid searching hopeless nodes.

Notice that for each S_i which contains the defining constraints for a hidden variable, there is also an "empty" constraint ϕ in it. The current partial model will not change

if an "empty" constraint is added upon it. The empty constraint introduced here is to deal with the redundant hidden variables. When the number of maximum possible hidden variables is greater than that of the hidden variables the system actually has, some generated hidden variables can not be introduced to the system. For example, a system has two hidden variables, but the maximum number of possible hidden variables is 3, resulting in the generation of three subsets in ***DS*** for these three hidden variable. The target model in fact choose constraints from only two of these subsets; for another subset, the target model will select the empty constraint.

An auxiliary forward checking method (Russell & Norvig 2003) is also performed; that is, when a new constraint is added, all the constraints in ***FCS*** that are conflicting with this new constraint will be ignored in the later search process. In order to prune more sub-trees, the exploration order of the subsets in ***DS*** is determined by the number of *legal constraints* in these subsets: the subset which has the minimum number of legal constraints will be explored first. The legal constraints are all the constraints in ***FCS*** that do not conflict with any constraint in the current partial model.

4.6 Pseudo Code of The Tree Search Algorithm

Step1: **Preprocessing**
Step 1-1: Constraint Generation, get ***ICCS***
Step 1-2: Constraint Filtering, get ***FCS***
Step 1-3: Calculating Conflict Set and Dependency Set, get *ConflictMatrix* and *DependencyMatrix*.
Step 1-4: Partition ***FCS***, get ***DS***
specify model size *ModelSize*
Step2: **Backtracking Search**
Begin
PartialModel=null; // current partial model.
ExploredSubset=empty;
// record the subsets that have been explored;
LegalSet:= ***FCS***;
Backtracking_FC(PartialModel, ExploredSubset, LegalSet)
End
The Function Backtracking_FC is defined as follows:
Backtracking_FC (PartialModel, ExploredSubset, LegalSet)
a. if size of PartialModel $\geq$ *ModelSize*
then return; // the exit of the recursion
b. Select a subset $S_i \in$ *DS-ExploredSubset*,
St. min($|S_i \bigcap LegalSet|$)
c. if $S_i \bigcap LegalSet == \emptyset$ return;
// no legal constraint, exit.
d. ExploredSubset:=ExploredSubset+S_i;
e.For each constraint C1 $\in S_i \bigcap LegalSet$
Begin
CS_C1:=Conflict Set of C1;
LegalSet:= LegalSet- CS_C1;
PartialModel:=PartialModel+ C1;
if *CheckPartialModel*(PartialModel)==true then
Begin
if size of PartialModel $<$ *ModelSize*
Backtracking_FC(PartialModel, ExploredSubset,LegalSet)
Else
CheckCompleteModel(PartialModel);
End // if CheckPartialModel()==true
LegalSet:= LegalSet + CS_C1;
//Restore the legalSet
PartialModel:=PartialModel- C1;
End //for each constraint
f. ExploredSubset:=ExploredSubset- S_i;
//restore exploredSubset
End Function

4.7 Model Checking Algorithm

The model checking algorithm is divided into two parts, every time a new constraint is added into the current partial model, the partial model checking function (*CheckPartialModel()* in pseudo code) will be performed on the current partial model. This checking algorithm will quickly check whether this partial model is consistent. There are two submodules in this function:

1. Contradictory Check: Checking whether current partial model contains conflict constraints based on the *ConflictMatrix*.

2. Dimensional Consistency Check: Checking the situation in section 3.2.4, in which the conflict involves more than two constraints.

Another model checking module, *CheckCompleteModel()*, checks the other properties of well-posed models, as stated in (Coghill *et al.* 2004), including model language, model connection, model completeness, singularity, connection, causal ordering and model coverage, will be only performed on "full models" which attain the pre-specified model size.

The causal ordering check is based on the *DependencyMatrix*. The JMorven package is tailored and slightly modified as an embedded module to support the model coverage test. Because of the relatively expensive computational cost of JMorven simulation, the model coverage test is arranged in the final stage, only well-posed models which pass all the other model checking modules are allowed to be simulated by JMorven.

5. Experimental Results

Using the above efficient algorithm, the learning of CM1 and CM2 models are throughout tested. We focus on the influence of hidden variables and incomplete training data on the learning reliability.

5.1 Experimental Methodology

For each model, We will start from the easiest experiment, which is given maximum number of variables and complete data. If it succeeds, we will conduct more difficult experiments, categorized by the following conditions: losing nonderivative variables, partial or not specifying the state variables, and losing the derivative variables.

For each of the above conditions, first complete training data, obtained from JMorven's complete envisionment, will be provided. If our algorithm can find the target model, the

Experiment ID	Hidden Variable	Known State Variable	Success
CM1-E1	qx	c1, c2	Yes
CM1-E2	qx, f12	c1,c2	Yes
CM1-E3.a	qx,c1	c2	No
CM1-E3.b	qx,c1	c2	No
CM1-E4.a	qx,c1	c2,Hidden	No
CM1-E4.b	qx,c1	c2,Hidden	Yes *
CM1-E5	qx, f12,f21	c1,c2	Yes
CM1-E6	qx,f12, c1	c2,Hidden	No
CM1-E7	qx	None	Yes
CM1-E8	qx,f12	None	No

* under additional domain-specific knowledge

Table 5: Experimental Conditions for CM1

experiment will be tested by providing all the elements in the power set of the training data, and get the learning reliability from these result.

For CM1, there are 6 qualitative states, so there will be 2^6 =64 experiments in each different condition, the computation cost is tolerable. For CM2 there are 14 qualitative states in the complete envisionment, as shown in Appendix B. So there will be 2^{14} =16,384 experiments. This will be very computational expensive.

In order to accelerate the calculation, we take the following approach: instead of finding the well-posed models for different training data in each experiment separately, we will first search all the well-posed models only once, discarding the training data. Then in each experiment, the set of well-posed models will be narrowed by different training data. So the search for well-posed models are only executed once and the results are used 2^n times (n is the number of qualitative states). This approach can be done easily by modifying the original tree search algorithm:

a. First disabling the constraint filtering function (step 1-2) in the preprocessing phase and the coverage test in the *CheckCompleteModel()* function. After searching, all the well-posed models found in the algorithm will be stored for later use.

b. Then for each experiment with different training data, all the previously found well-posed models which contain the inconsistent constraints will be filtered out.

c. Finally the remaining well-posed models will be tested for coverage by JMorven one by one, and the final results are obtained.

5.2 Experiments for CM1

The target model and complete envisionment obtained from JMorven are shown in Appendix A. Table 5 shows the set of experiments under different conditions.

Table 6 shows the performance of the search algorithm discarding the training data. The learning reliability of successful ones is shown in Figure 2. The same learning curve is obtained in CM1-E1,E2,E5 and E7. CM1-E4.b has different curve which will be explained later.

CM1-E1 is the easiest one, which has only one hidden variable qx and fully specified state variables. There exists

Experiment ID	Search Space	Well-posed Models	Running Time (Milli-sec)
CM1-E1	614,061	2,520	13,496
CM1-E2	961,875	5,176	28,271
CM1-E3.a	2,975,625	1,672	64,732
CM1-E3.b	2,975,625	23,136	59,450
CM1-E4.a	1071,225	8,192	34,367
CM1-E4.b	1,071,225	5,848	31,421
CM1-E5	1,975,509	7,128	37,672
CM1-E6	1,358,127	7,336	57,295
CM1-E7	2,736,741	34,272	52,046
CM1-E8	6,281,875	80,616	99,322

Table 6: Performance for learning CM1

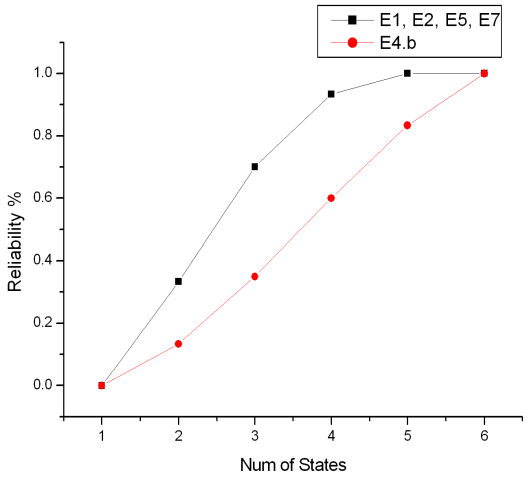


Figure 2: Learning Reliability of CM1 Experiment 1,2,5 and 7.

a kernel set which is defined in (Coghill *et al.* 2004). The elements in the kernel set are all pairs:

(1,2) (1,3) (1,5) (2,3) (2,4)

The number in the pairs stands for the State ID in the complete envisionment, which is shown in Appendix A. This means for learning CM1, the above pairs and all subsets including these pairs can successfully learn the right model. CM1-E2 removes variable f12 and CM1-E5 removes both f12 and f21. The same learning curve and kernel set as CM1-E1 are obtained. In CM1-E7, no state variables are specified, but only qx is hidden variable, we also can successfully learn the model and get the same learning curve and kernel set as CM1-E1. In addition, in CM1-E7 we found another kind of "correct" model:

Sub (dt0 Hid0) (dt0 c2) (dt0 c1)
Inc (dt0 c1) (f12)
Inc (dt0 c2) (f21)
Inc (dt1 f12) (Hid0)
Dec (dt1 f21) (Hid0)

This model can cover exactly the complete data, but it has different physical meaning: it is a flow-based system, in which it is the change of flow that causes the change of the concentration. On the contrary, the target model is a concentration-based system. The algorithm can not discriminate these two models because of missing state variable information.

In CM1-E3.a, the state variable c1 becomes hidden variable. Given complete data, 24 models which can cover exactly the complete data are found, of which only one is equivalent to target model. 112 over-generalized models which can cover not only the complete data but some other data are also found.

In CM1-E4.a, we make it easier than CM1-E3 by "telling" the system that there is a hidden variable and this hidden variable is a state variable. We found 8 over-generalized models and the target model. Then we go into the details of the non-target models in the result, and find out that all the over-generalized models have the following two constraints (or symmetric constraints in which the positions of Hid0 and Hid1 are swapped):

C5.1 Inc (dt1 Hid1) (dt0 Hid0)
C5.2 Dec (dt0 Hid0) (dt0 Hid1)

Hid0 and Hid1 are two hidden variables. We can add additional hypothesis upon the model to filter out the over-generalized models which contain the above two constraints. One possible hypothesis is that there is no hidden relation in the target model.

Another hypothesis can be no "redundant" hidden variables in the model. Hid0 is a redundant hidden variable because we can induce the following constraint from C5.1 and C5.2:

C5.3 Dec (dt1 Hid1) (dt0 Hid1)

Hid0 becomes logically "redundant", although Hid0 may have physical meaning in a real system. In CM1-E4.b we add this hypothesis and the experiment is successful. But we got different result from other successful experiments. The kernel set becomes smaller and is a subset of CM1-E1's:

(1,3) (2,3)

The learning reliability decreases as shown in Figure 2.

Experiment ID	Hidden Variable	Known State Variable	Success
CM2-E1	q1,q2	c1, c2	Yes
CM2-E2	Q1,q2	None	Yes
CM2-E3	q1,q2,f12	c1,c2	Yes
CM2-E4	q1,q2,f12	None	No
CM2-E5	q1,q2,f12,fo	c1,c2	Yes
CM2-E6	q1, q2,c1	c2,Hidden	Yes
CM2-E7	q1, q2,c1,f12	c2,Hidden	No

* No Model Connection Check

Table 7: Experimental Conditions for CM2

Experiment ID	Search Space	Well-posed Models	Running Time (Milli-sec)
CM2-E1	29,430,625	14,748	416,471
CM2-E2	216,825,625	657,785	1,660,930
CM2-E3.a	52,521,875	12,216	667,982
CM2-E3.b	52,521,875	41,784	517,114
CM2-E4	669,921,875	954,754	3,804,080
CM2-E5	113,358,609	10,080	1,172,365
CM2-E6	3,051,209	47,696	1,009,223
CM2-E7	95,918,823	38,748	1,709,191

Table 8: Performance for learning CM2 (No training data is provided)

Based on CM1-E4.b, in CM1-E6, f12 becomes hidden variable. In this experiment, the envisionment only includes 5 qualitative states. 808 models are found in this experiment. A detailed investigation in these models indicates that no simple hypothesis can be added upon the problem domain to discriminate the target model from the others.

Finally, the last CM1-E8 is based on and harder than CM1-E7 in the sense of hiding both qx and f12 and not specifying any state variables. In this experiment, our algorithm finds 380 models under complete data, of which only 12 models are equivalent to target models.

5.3 Experiments for CM2

The target model of CM2 has been given in the previous section, this model is equivalent to the cascaded tanks model in (Coghill *et al.* 2004) and the set of experiments under different conditions are listed in Table 7. Suppose the inflow u={pos, zer} , there are 14 qualitative states in the complete envisionment, shown in Appendix B.

Like the experiment for CM1, the performance of the search algorithm are listed in Table 8. The learning reliability is illustrated in Figure 3.

The kernel set of the easiest situation CM2-E1 is

(0,2,5) (0,2,7) (0,2,11) (0,2,13) (0,4,5) (0,4,7) (0,4,11) (0,4,13)

The result is the same as that obtained in (Coghill *et al.* 2004).

In CM2-E2, all the conditions are the same as CM2-E1 except the state variables are not specified. Similarly as CM1-E7, we found two different kinds of models that can

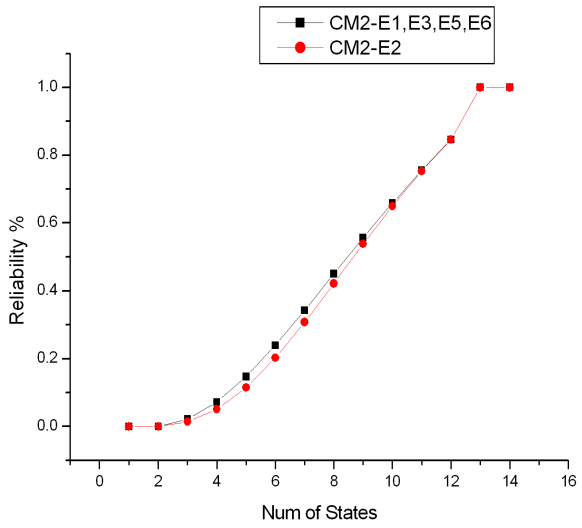


Figure 3: Learning Reliability of CM1 Experiment 1,2,5 and 7.

cover exactly the complete data: one is the target model and the other is shown as follows:

Inc (dt0 c2) (dt0 fo)
Inc (dt0 f12) (dt0 c1)
Sub (dt0 Hid1) (dt0 c1)(dt0 c2)
Sub (dt0 Hid0) (dt0 u) (dt0 f12)
Inc (dt1 c1) (dt0 hid0)
Inc (dt1 fo) (dt0 Hid1)

In this model, c1 is correctly identified as state variable, while fo is wrongly treated as state variable. This model can be seen as a "mixture" of concentration-based and flow-based system. The kernel set found in CM2-E2 is smaller than that in CM2-E1:

(0,2,5) (0,2,7) (0,2,11) (0,2,13) (0,4,7)
(0,4,5,6) (0,4,6,11) (0,4,6,13)

In this kernel set, to become an element of the kernel set, (0,4,5), (0,4,11) and (0,4,13) must be accompanied by an additional state 6. The learning reliability also slightly decreases, as shown in Figure 3.

In CM2-E3, f12 becomes hidden variable, we can still learn the target model in this experiment. Then based on CM2-E3, in CM2-E4 we do not specify state variables, resulting in an unsuccessful experiment. Again based on CM2-E3, in CM2-E5 fo becomes hidden variable, this experiment is similar to CM1-E5, only state variables are known, all the other variables become hidden ones. The target model is successfully learned in this experiment. In CM2-E6, the state variable c1 becomes hidden variable, but one hidden variable is specified as state variable. In this experiment, we succeed in descriminating the target model from other well-posed models. Based on CM2-E6,in CM2-E7, f12 is removed, and we find there exist non-target models in the learning result. In all the successful experiments except CM2-E2, the kernel set and learning reliability are the same.

Some initial conclusion can be drawn from the above experiments:

1. The state variables which have more than one derivative in the 0th differential plane are very important for learning, learning task will become difficult if some of them become hidden variables (CM1-E3, CM1-E4, CM1-E6, CM2-E6, CM2-E7).

2. If a learning task can not be successfully accomplished, we can add more domain specific information to guide the learning (CM1-E4.b). It is still possible to learn the target models. But the kernel set and learning reliability may change.

3. The specification of state variables is also a factor which can influence the learning. Partially or not specifying the state variables will result in a large search space and may lead to failed experiments (CM1-E3,E4,E8 and CM2 E4, E7).

4. Given the right model size, number of hidden variables, and fully specifying the state variables, the non-state variable has the least influence on learning (CM1-E1, E2, E5 and CM2-E1, E3, E5).

5. If state variables are not specified, too many hidden variables can lead to unsuccessful learning (CM1-E7, E8 and CM2-E2, E4).

6. Conclusions and Future Work

The work presented in this paper continues the previous work in (Coghill, Garrett, & King 2004) and (Coghill *et al.* 2004). The contributions of our work are:

First, a more flexible model representation JMorven is adopted. This is not only an alternative representation method, but also has the potential ability to deal with fuzzy data and reason about more than two derivatives.

Second, an problem-specific backtracking tree search algorithm is proposed, which can find out all the well-posed models efficiently.

Third, the learning of qualitative models under the conditions of different hidden variables and specified state variables are systematically tested and the influence of the hidden and state variables are analyzed.

Last, for speeding up the experiment and avoiding repeated calculation, the same routine work for searching the well-posed models is performed only once in each condition, and the result can be used by all the experiments.

Future work will involve the following aspects: First, the precision of the model can be improved by adding more quantities in the quantity space. The learning task will become more challenging. Second, more complex qualitative models, which have more training data and system variables, can be analyzed. Another direction is using the evolutionary computation to learn the qualitative models with huge size, in the field of which traditional methods do not perform well.

References

Bhaskhar, R., and Nigam, A. 1990. Qualitative physics using dimensional analysis. *Artificial Intelligence* 45:73–111.

Bruce, A. M., and Coghill, G. M. 2005. Parallel fuzzy qualitative reasoning. In *Proceedings of the 19th International Workshop on Qualitative Reasoning*, 110–116.

Coghill, G. M.; Garrett, S. M.; Srinivasan, A.; and King, R. D. 2004. Qualitative system identification from imperfect data. Technique Report AUCS/TR0501, Department of Computing Science, University of Aberdeen.

Coghill, G. M.; Garrett, S.; and King, R. D. 2004. Learning qualitative metabolic models. In *European Conference on Artificial Intelligence (ECAI'04)*, 445–449.

Coghill, G. M. 1996. *Mycroft: A Framework for Constraint based Fuzzy Qualitative Reasoning*. Ph.D. Dissertation, Heriot-Watt University.

Hau, D. T., and Coiera, E. W. 1993. Learning qualitative models of dynamic systems. *Machine Learning* 26:177–211.

Iwasaki, Y., and Simon, H. A. 1986. Causality in device behavior. *Artificial Intelligence* 29:3–32.

Kuipers, B. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge, MA: MIT Press.

Morgan, A. 1988. *Qualitative behaviour of Dynamic Physical Systems*. Ph.D. Dissertation, University of Cambridge.

Richards, B. L.; Kraan, I.; and Kuipers, B. 1992. Automatic abduction of qualitative models. In *National Conference on Artificial Intelligence*, 723–728.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach 2nd Edition*. Prentice Hall. chapter 5, 142–146.

Say, A. C., and Kuru, S. 1996. Qualitative system identification: deriving structure from behavior. *Artificial Intelligence* 83:75–141.

Shen, Q., and Leitch, R. 1993. Fuzzy qualitative simulation. *IEEE Transactions on Systems, Man, and Cybernetics* 23(4):1038–1061.

Wiegand, M. 1991. *Constructive Qualitative Simulation of Continuous Dynamic Systems*. Ph.D. Dissertation, Heriot-Watt university.

Appendix A

The JMorven Model for CM1:

Differential Plane 0
C1: Inc (dt 0 f12)(dt 0 c1)
C2: Inc (dt 0 f21)(dt 0 c2)
C3: sub (dt 0 qx)(dt 0 f12) (dt 0 f21)
C4: Inc (dt 1 c1)(dt0 qx)
C5: Dec (dt 1 c2)(dt0 qx)

Differential Plane 1
C6: Inc (dt 1 f12)(dt 1 c1)
C7: Inc (dt 1 f21)(dt 1 c2)
C8: sub(dt 1 qx)(dt 1 f12)(dt 1 f21)

Complete Fuzzy Vector Envisionment for CM1. c1={pos, neg} means that the zero derivative of c1 is "positive" while the first derivative of c1 is "negative".

State ID	c1	c2	f12	f21
0	{zer , zer}	{zer , zer}	{zer , zer}	{zer , zer}
1	{zer , pos}	{pos , neg}	{zer , pos}	{pos , neg}
2	{pos , neg}	{zer , pos}	{pos , neg}	{zer , pos}
3	{pos , zer}	{pos , zer}	{pos , zer}	{pos , zer}
4	{pos , pos}	{pos , neg}	{pos , pos}	{pos , neg}
5	{pos , neg}	{pos , pos}	{pos , neg}	{pos , pos}

Appendix B

Compete Envisionment for CM2, supposing inflow u={pos, zer}

State ID	c1	c2	f12	fo
0	{zer , pos}	{zer , zer}	{zer , pos}	{zer , zer}
1	{zer , pos}	{pos , neg}	{zer , pos}	{pos , neg}
2	{pos , zer}	{zer , pos}	{pos , zer}	{zer , pos}
3	{pos , pos}	{zer , pos}	{pos , pos}	{zer , pos}
4	{pos , neg}	{zer , pos}	{pos , neg}	{zer , pos}
5	{pos , zer}	{pos , zer}	{pos , zer}	{pos , zer}
6	{pos , zer}	{pos , pos}	{pos , zer}	{pos , pos}
7	{pos , zer}	{pos , neg}	{pos , zer}	{pos , neg}
8	{pos , pos}	{pos , zer}	{pos , pos}	{pos , zer}
9	{pos , pos}	{pos , pos}	{pos , pos}	{pos , pos}
10	{pos , pos}	{pos , neg}	{pos , pos}	{pos , neg}
11	{pos , neg}	{pos , zer}	{pos , neg}	{pos , zer}
12	{pos , neg}	{pos , pos}	{pos , neg}	{pos , pos}
13	{pos , neg}	{pos , neg}	{pos , neg}	{pos , neg}

From Whiteboard to Model: A Preliminary Analysis

Praveen Paritosh[1] and Will Bridewell[2]

[1] Qualitative Reasoning Group, Northwestern University, Evanston, IL, USA

[2] Computational Learning Laboratory, Center for the Study of Language and Information, Stanford University, Stanford, CA 94305 USA

{paritosh@cs.northwestern.edu, willb@csli.stanford.edu}

Abstract

Building models of a complex system such as an ecosystem or a chemical plant is an arduous task that can take several person months to complete. One rarely knows the scope of the model, its assumptions and claims, at the outset of the task, let alone how to state those in a formal language. To make this task manageable, modelers start at the whiteboard – by making free-form drawings that capture their current understanding of the studied system. These drawings need not conform to any particular ontology and may lack internal coherency or consistency. Nevertheless, such drawings can help organize one's thoughts and can capture key participants and relationships in the dynamic system. We argue that these free-form drawings facilitate the modeling process, based on evidence from modeling in practice. We analyze the relationship between free-form drawings and formally encoded models. We then suggest how to exploit these relationships to develop a modeling environment that supports a tighter integration between conceptual and detailed modeling.

1 Introduction

Model building is a common and vital task in the sciences. The resulting artifacts of thought let us test the implications of our theories, help us better understand complex systems, and support effective communication and education. Moreover, models specified in a computable language support prediction and diagnosis, thereby enabling discussions about things such as the migration of killer bees and the efficacy of carbon sequestration. Unlike theories, models describe specific situations, real or imagined. For example, the theory might discuss the migration of a general population across a landmass, whereas a model would make specific claims about the movement of *Apis mellifera scutellata* into south Texas. Given this model, one might predict that residents of El Paso will be knee deep in bee-riddled corpses by March 1979. In comparison, hypotheses are singular, testable statements such as "building a border fence 50 meters high will protect us all from the bee invasion." With an appropriate model, one could gauge the plausibility of this hypothesis before committing the required resources.

Despite the ubiquity of modeling, there have been relatively few analyses of the task of modeling. More often, researchers emphasize the effectiveness of an encoding tool [e.g., Bridewell et al., 2006] or the use of the resulting model. Recent work in chemical engineering [Foss, Lohmann and Marquardt, 1998] and in educational settings [Sins, Savelsbergh and van Joolingen, 2005] exploring the process of modeling itself are more the exception than the rule. One general finding is that the modelers benefit from multiple representations and that each one has its own merits [Lohner, van Joolingen, and Savelsbergh, 2003]. These results suggest that modeling environments should, in principle, support different views of the artifact and that these views should map onto each other.

In the qualitative reasoning community, Bredeweg and colleagues have recently described a framework for building qualitative models [Bredeweg et al., in press]. One of their findings is that the use of loosely constrained conceptual models provided considerable, if not necessary, support for the development of formal qualitative models. Bredeweg used a concept map to capture the earlier stage of modeling. We believe that the simplicity and popularity of concept maps make them a suitable interface for describing the pre-formal free-form modeling that happens at the whiteboard, and the rest of this paper will stick to this assumption[1].

The natural extension of these findings is the development of a modeling environment that provides a tighter integration between building the concept map and the detailed model. In this paper, we describe how such a system might be designed. The next section discusses the modeling process in more detail, with special emphasis on the utility of pre-formal conceptual models. We then describe how concept maps facilitate building detailed models. Next we present a catalog of ontological relationships and mapping operations between elements in the concept map and the detailed model. Finally, we conclude with questions and future research issues that this discussion generates.

[1] This is a simplifying assumption, the whiteboard provides a much richer interface, perhaps more like the sketching systems, e.g., sKEA [Forbus and Usher, 2002].

2 The Modeling Process

Model building forms a part of larger tasks such as design and scientific investigation. The task goals influence the trade-offs among generality, realism, and precision of the model [Levins 1966]. Models built for communicating the relationships in a complex system tend to be more general, while models built for process control emphasize precision, and so on. One's available knowledge and data also influence the modeling task. Domains such as chemical engineering and circuit design are knowledge-rich, which enables a realistic expression of the entities and relationships within the modeled system. Other domains, such as ecology, are less theory-driven, and the amount and type of data will influence one's modeling decisions. Models range from being purely descriptive and explanatory, e.g., in political and social domains to being predictive, e.g., in engineering.

Foss and colleagues [1998] performed a field study of the modeling process in the domain of chemical engineering, in which they interviewed sixteen modeling practitioners with an average modeling experience of over ten years. These interviews followed a case study wherein the modelers described a realistic modeling experience. On the basis of these interviews, Foss et al. identified six distinct activities: 1) problem understanding, specification, initial data collection, 2) conceptual modeling and model representation, 3) implementation and verification, 4) initialization and debugging, 5) validation and 6) documentation. Notably, the modeling process is not linear. That is, the modeler may freely move among these six activities without any fixed pattern. However, as model refinement progresses, the modeler moves through the chunks sequentially as a moving window capturing more than one chunk at a time. As the modeling process goes by, the degree of back-steppings to the earlier chunks diminishes in favor of forward-steppings to the later chunks. Nevertheless, there exist numerous iterations between the chunks rendering a highly intertwined and complex modeling process.

Foss's study offers important insights for builders of modeling environments. First, modeling is not a strictly progressive refinement from conceptual to detailed models. This finding suggests that environments should support links among the tasks so that modelers need not shift to external media as they work. Second, the modeling environment must provide tight integration between the various modeling activities. Being able to work with several representations of a model becomes problematic when they are unsynchronized. The environment should treat each representation as an index into the others so that the modeler can move about freely with ease. And third, no matter the richness of knowledge or data about an environment, conceptual modeling remains important. Thus, builders of modeling tools should consider including various level of representation – including those that permit inconsistency as is inevitable as one begins to model, into their software.

One of the first stages in modeling a complex system involves the identification of the model's scope, which includes the relative entities and relationships expressed at a high level. This task fits well into Rittel and Webber's (1973) notion of a *wicked problem*. In particular, the problem definition is usually vague and evolving, proposed

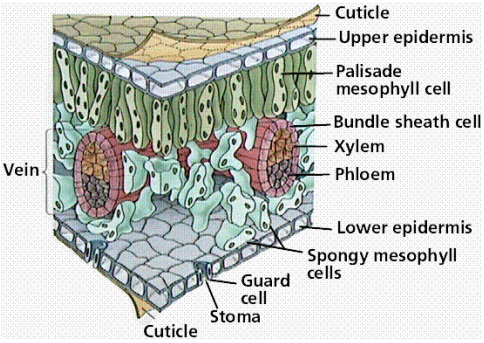


Figure 1. A visual diagram of the cross section of a leaf, reprinted from Farabee [2001]

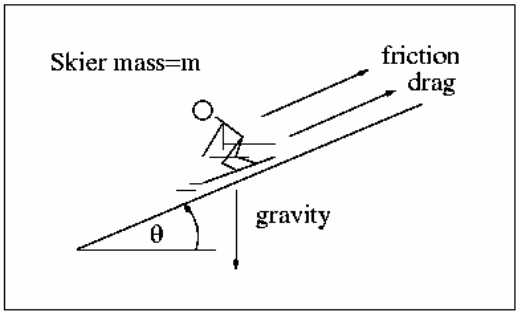


Figure 2. A spatial free-body diagram of a skier.

solutions can create new problems, and multiple solutions may exist with no obvious measures of preference. To begin, one often represents the target system with free-form text descriptions and drawings on paper or whiteboard. As with the entire modeling process, the goal is to make one's knowledge explicit, but at this stage issues of syntax and semantics can serve as barriers and interfere with one's creativity. So, when working with pen and paper one introduces objects and relationships without concern for incomplete specifications, consistent typology, or formal correctness.

Most of the quantitative modeling environments today (e.g., AspenTech's HYSYS, ASCEND, SPEEDUP, STELLA, Simulink), primarily focus on the formal encoding of models without much support for the free-form conceptual modeling that takes place on a whiteboard. On the other hand, qualitative modeling environments like VModel [Forbus *et al.*, 2004], Betty's Brain [Leelawong 2005], Garp[2], among others, provide richer support for less detailed models. However, these are

[2] Downloadable from http://hcs.science.uva.nl/QRM/software/

not unconstrained enough for capturing the possible inconsistency and ambiguity of the whiteboard drawing. Underlying each of these environments is a modeling ontology that constrains and restricts what can be drawn, which is precisely what gives these environments power to reason with the models built using them. There are a large number of software tools available as "mind-mapping tools" [Buzan 1991] that support pre-formal unconstrained drawing. The end result, however, in mind mapping is the drawing that is produced. There is very little work on elaborating or fleshing the output of mind map into a model that can be reasoned with.

Lets look at the different types of drawings that are built while modeling. We classify free-form diagrams into the following three categories:

1. *Visual drawings* are faithful to the salient spatial relationships and bear apparent resemblance to the object or system being drawn. Figure 1 shows a visual diagram of a leaf.
2. *Spatial drawings* use the spatial layout of the drawing medium. Examples include course of action diagrams and free body diagrams in classical mechanics. In this representation, one introduces abstractions and metaphorical conventions such as arrows that convey spatial direction. Figure 2 shows a free body diagram of a skier.
3. *Abstract drawings*, such as UML diagrams, organizational charts, and concept maps ignore the implicit spatial dimension of the drawing medium. In these figures, the relative location of two objects does not necessarily communicate a real spatial relationship. Figure 3 shows an abstract drawing, a concept map.

We admit that free-form diagrams are often complex and rich with implicit knowledge. An aspect of complexity of free-form diagrams is that they can contain different parts that are visual, spatial and abstract in the same diagram, and humans are able to rely on vast commonsense knowledge to interpret it. Ideally, we would like to provide the modeler the freedom of drawing on the whiteboard, but given the complexity of automatically understanding them, we restrict ourselves to the third type above, abstract diagrams. One possibility is to take the sKEA approach [Forbus and Usher, 2002], and allow the modeler to explicitly label every element of the drawing using an ontology like the Cyc[3] knowledge base.

We believe that concept maps are attractive for the abstract diagrams for their simplicity and flexibility. Concept maps [Novak and Cañas, 2006] are graphical tools for organizing and representing knowledge. The power of concept maps comes from the simplicity of the ontology: box-and-lines. Boxes denote concepts and have linguistic labels that identify what they represent, and lines specify a relationship (causal, spatial, etc.) between two concepts. Propositions contain two or more concepts connected using linking words or phrases to form a meaningful statement. Recently, Bredeweg *et al.*, [2006] included support for concept maps in the Garp3 system in the form of a *sketch mode*. However, in their software, the elements of the sketch are not connected to the elements of the detailed model. Here, we emphasize the value of connecting these representations.

The ease of concept maps comes at a price. First, one cannot simulate concept maps or use them to make strong predictions about system behavior. Second, one may explain away phenomena by leaving out important, nontrivial details. For example, a concept map that claims "carbon sequestration *reduces* global warming" might be too simplistic and explain away the complex mechanisms of the process. Put simply, it is possible to make models that state the very fact that the model ought to explain or predict, without providing any richer explanation. And third, one may assume shared understanding of linguistic labels, which can hide the one's preconceptions behind the ambiguity of meaning. The use of a formal, shared vocabulary, such as Cyc[1] for naming the concepts and relationships can safeguard against this problem to a large extent, but at the cost of representational freedom.

The modeling process consists of fleshing-out the concept map to a more detailed model. We call the shift to a more formal representation (i.e., one that can be reasoned over) *encoding.* This step involves moving to a well-defined ontology, such as Forrester diagrams [Forrester, 1961], qualitative process theory [Forbus, 1984], or mathematical equations, and assumes a firm understanding of the concepts. Beginning at the formal stage can be somewhat challenging, but the concept map constrains what one will encode and facilitates the formalization procedure. In the next section, we describe this relationship in more detail.

3 Concept Maps Facilitate Modeling

As described above, the concept map identifies the entities and relationships that need to be further encoded and elaborated in the detailed model. Introductory texts on modeling in various domains, e.g., biological systems [Haefner, 2005], ecological modeling [Jorgensen, 2001] advise modelers to begin with such a drawing of the system. This points to the first benefit of concept maps: ease of knowledge elicitation. Knowledge elicitation is facilitated as the concept map allows the expert to express their mental model in a vocabulary that is close to their models by allowing linguistic labels for entities and relationships. Furthermore, the concept map makes is easier to try out ideas and cast them aside if they fail to satisfy the modeling goals and constraints. In his landmark book, Productive Thinking, based on a case study of Albert Einstein, Wertheimer (1945) argues that a bottleneck to scientific breakthrough is overcoming the structure of existing theories. By providing a freer ontology, concept maps might make it easier for this to happen. In the

[3] http://www.cyc.com/

NatureNet Redime[4] effort to build qualitative models of ecological systems, a first step has been building a textual description and a concept map of the system of concern. This claim of ease of knowledge elicitation has indirect support from practical modeling efforts and conventional modeling wisdom.

Second, the concept map is an important aspect of documentation of the modeling process itself. It captures the conceptual evolution of the modelers' thought process. It also presents a higher level description of the detailed model, in the sense of requirements in software [Jackson, 1995] and design rationale [Moral and Carroll, 1996]. The concept map has communicative value, as it might be easier to get started with the concept map before looking at the simulatable model. For example, in the CMEX[5] project, which was NASA's outreach effort to explain the Mars exploration enterprise to lay people; a collection of about one hundred concept maps detailing various aspects of Mars exploration were released.

Third, for large models that don't fit on a screen, the concept map can be used as a navigational interface for browsing the detailed model by pointing to parts of it that one is interested in exploring in more detail. Furthermore, concept maps contain enough structural information that they can be used to retrieve analogous models from a library of previous models and making analogical suggestions during modeling [e.g., Leake *et al.*, 2003].

4 Usage Scenarios

Designers of model development environments can take advantage of the relationship between concept maps and models both to create a simplified user interface and to scaffold the encoding of formal models. To address the first point, the conceptual model serves as an index to the components of the detailed version, letting one navigate quickly to the relevant sections of the model and access associated interface elements with ease. For the second point, the conceptual model can highlight incompletely specified regions of the system and help the user avoid errors in consistency. In the remainder of this section, we discuss how the conceptual and detailed modeling activities fit in the modeling environment.

There are two possible scenarios of how the modeling environment might support both conceptual and detailed modeling:

1) *Sequential encoding:* One starts with a concept map that is progressively encoded into a simulatable model. In this scheme, the concept map eventually "disappears."
2) *Parallel encoding:* Both the concept and the model are maintained at all times as the modeler goes back and forth elaborating and drawing connections between them.

We believe that the parallel encoding is a more natural model of the modeling task. The Foss *et al.* [1998] study provides direct support of this intertwined nature of modeling activity where one is going back and forth between conceptual and detailed representations. Furthermore, this view suggests that a concept map is more than a stepping stone to a model. It is a continuously developing high-level representation of the model that one wants to keep around, even after having developed a detailed model for explanatory, communicative purposes.

The sequential encoding scenario constrains the ontological freedom of the concept map. It is easier to imagine gradually elaborating from concept map to the model if it were true that the concept map ontology was a strict abstraction of the model ontology. However it is not necessary. That is, the mapping of interactions expressed at the concept map to those in the model may be one-to-one, many-to-many, one-to-many, or many-to-one. A concept map is not just a sparser representation of a model. Sometimes the concept map might contain additional information about the system that never goes into the final model, as the concept map ontology allows one to represent more than what one might be able to say in the detailed modeling ontology. The argument against sequential encoding is that of ontological incompatibility.

The *sketch mode* in the current version of Garp supports the sequential encoding scenario. It is plausible that the modeler might go back and forth between the sketch mode and the qualitative modeling mode; however, the environment does not provide direct support for connecting the sketch and the qualitative model.

In the parallel encoding scenario, the software must provide facilities for keeping concept map and model in sync as they evolve. To implement such tight coupling between the concept map and the model, we need an analysis of relationships between them, which amounts to answering the questions: 1) What are the ontological relationships between elements (nodes and edges) in the concept map and the model? and 2) What kind of activities relate the elements in the concept map and the model? The answers to these questions provide the software with the knowledge required to connect the models. As a first start, the modeler can manually annotate such connections. It is an empirical question for future research to see what aspects of these can be automated and benefit the modeler by automatically pointing out incompleteness and mismatches.

5 Relating Concept Maps to Models

Concept maps draw their power from their lack of representational constraints. This freedom lets one create inconsistent diagrams and mix together causal, structural and other types of information with minimal formal syntax. In addition, one can include components that communicate the scope of a model even though those details will exist only implicitly in the formalized version. In this section,

[4] http://hcs.science.uva.nl/projects/NNR/
[5] http://cmex.ihmc.us/

we examine the relationships expressed in a concept map, how these relationships translate into an encoded model, and the utility of maintaining explicit links between the two representations.

Concept maps can take many forms and encode several types of knowledge: UML diagrams, organizational charts, flowcharts, and so on. To focus the discussion, we emphasize concept maps built as outlines for a causal model (qualitative or quantitative). We ground our discussion in the concept maps built in the CMEX project and those built by Bredeweg's group in the NatureNet Redime project. Although these maps cover a broad scope of topics, ranging from autonomous spacecraft control to river Mesta's ecosystem, we posit that they contain six distinct classes of knowledge: *causal, spatial, mereological, taxonomic, control* and *parametric*. Each type of knowledge manifests either as nodes or as edges in a concept map. We also discuss where the knowledge ends in a qualitative model built using the QPT ontology in the discussion below:

1. *Causal:* Causal knowledge is a key part of explanations, and manifests in relationships such as "causes", "effects", "increases", and "is related to". These relationships map onto qualitative proportionalities and influences, but one can also specify more complex causal relationships like "consumes", "produces", and "regulates," that map onto processes. Relationships such as "enables" and "prevents" capture causal knowledge that becomes preconditions and quantity conditions in a qualitative model. In addition to those specified, we also include temporal relationships like "before", "after", and "during" in this causal category as they often related to a vague causal knowledge.
2. *Spatial:* This type of knowledge captures the spatial layout of entities in the modeled system. Explicit spatial relationships include "above", "below", "inside", "aligned", and so on. While encoding a qualitative process model, one may translate these relationships into preconditions for model fragments as they place limits on which entities can interact with one another.
3. *Mereological:* This type of knowledge describes the part-whole relationships between entities in the system and is expressed by relationships such as "consists of", "contains", and "includes".
4. *Taxonomic:* Taxonomies describe the type information for objects, which manifests as a subtype hierarchy in Garp. Defining specific objects as instance of general types enables the reuse of model fragments. One may describe these relationships with terms like "is a", "type of", "member of", "example of" (for class–instance hierarchies), and so on.
5. *Control:* These relationships introduce control flow into the concept map. For example, one can include a node that determines which of two outcomes will happen. Often control knowledge gives an explicit statement of preconditions and quantity conditions.
6. *Parametric:* Parametric nodes and edges let one introduce modeling abstractions like parameters of interest at the concept map level itself. Ideally, these objects appear directly in the encoded model. In concept maps, such relationships may exist as nodes that represent numeric quantities or edges that represent measurement operations.

As mentioned at the beginning of this section, we are restricting our goal to knowledge contained in causal models. For instance in domains like design, teleology, economics and aesthetics might be some of the other types of knowledge that are relevant to model building. To highlight these relationships, we appeal to the specific examples shown in Figure 3. This concept map describes the river Mesta's ecosystem [Uzunov et al. 2006] and contains seventeen distinct edge labels. We place these labels into the above categories as follows.

1. *Causal:* produces, provides, stimulates, regulates, consumes, feeds on, regulates, influences
2. *Spatial:* inhabit, provides habitat for, lives on
3. *Mereological:* consists of, has, contains
4. *Control:* determines the type of
5. *Parametric:* is measured by

The relation "is profited by" fails to fit in any of the delineated categories. However, consider the statement "particulate organic matter is profited by bacteria." This claim is somewhat misleading as the bacteria consume the particulate organic matter, which defines a process relationship similar to "feeds on" between these two entities. We used the concept map from the river Mesta study to show that many relationships specified in a concept map fall within a limited set of categories. The taxonomic relations do not show up in the concept map as they are modeled separately in Garp. In the next section, we examine the encoding operations associated with these types of knowledge.

6 Operations between Concept Maps and Models

In this section, we describe the operations that a modeling environment needs to have to support the parallel encoding model. We have not built this environment yet. After creating an initial concept map, one can begin the iterative process of model and concept map revision. At this point, the concept map itself becomes a key part of the user interface. Selecting a node will reveal an entity-specific dialog with which one can define either a type or an entity. In the former case, one specifies the properties of the type, which assumes the name of the node. In the latter, one either selects a type for the entity, or, both defines a new entity type and labels the node as an instance of the type. If a taxonomic edge connects two nodes, one can infer the type and properties of the child. In addition, if the concept map lacks an edge between a distinct entity type and its instantiation, the modeling environment can add it automatically. This action synchronizes the concept map and the encoded model and is an important tool for revealing relationships that were initially implicit but that

became explicit during the formalization process. More plainly, this activity helps one see their previously implicit knowledge, which may lead to a better understanding of the system and better modeling habits in the future.

One can also select the edges of the concept map and assign them to any of the six specified categories. Structural and spatial edges establish contexts and may imply a need for a transport process. For instance, a detailed cell model may have a pool of RNA within the nucleus and a pool of RNA in the cytoplasm. A model that incorporated mechanisms for transcription and translation would require a process that shuttles the RNA through the nuclear membrane. If the concept map contains mereological and boundary information, then the modeling environment could readily infer the necessity of such a process and remind the modeler to include it. If one labels an edge as taxonomic then the environment can relay properties from the source node to the source (an entity type) to the target (either more specific type or an instantiated entity). Finally, labeling an edge as causal will call up the process editing dialog. First, the environment ensures that the source and target of the link are both entities. If not, the interface will prompt the modeler to add an instantiation of the appropriate type. Afterwards, both entities will appear in the process definition, which allows the inclusion of other entities as participants. In the simple case, the modeler defines the process between two entities and moves to another relationship. If, however, other entities are introduced, the program must bring the concept map up to date by adding process links between the newly related nodes.

Next we present a catalog of operations between concept maps and models. The software must be aware of these, and can help the modeler annotate and connect the concept map and the model. This is a coarse representation of the types of activity links that exist between concept maps and models.

1. *Typing:* The modeler takes a node or an edge in the concept map and provides the type information for it from the ontology(e.g., identify something as a process, quantity, or an influence). At this point, the software can use templates associated with the types to point out the information that is needed to fully describe it in the modeling ontology. Further, local constraint satisfaction could propagate this information and anticipate the types of other nodes and edges connected to the object.
2. *Elaboration:* The modeler takes a node or edge in the concept map and decides to explode it and model it in further detail. The software makes sure that the internal and external connectivity to this object is maintained. Other than this, one can freely elaborate the object in any way allowed by the modeling ontology. This procedure is similar to the *model containers* idea in ModKit [Bogush, Lohmann and Marquardt, 2001].
3. *Filtering:* This operation has the modeler specify the elements in the concept map that will not be described in the simulation model. This could be because the detailed modeling ontology cannot encode those elements, or they might not be relevant to the task at hand.
4. *Annotation:* We allow this as a catchall relationship between the concept map and the model, where the

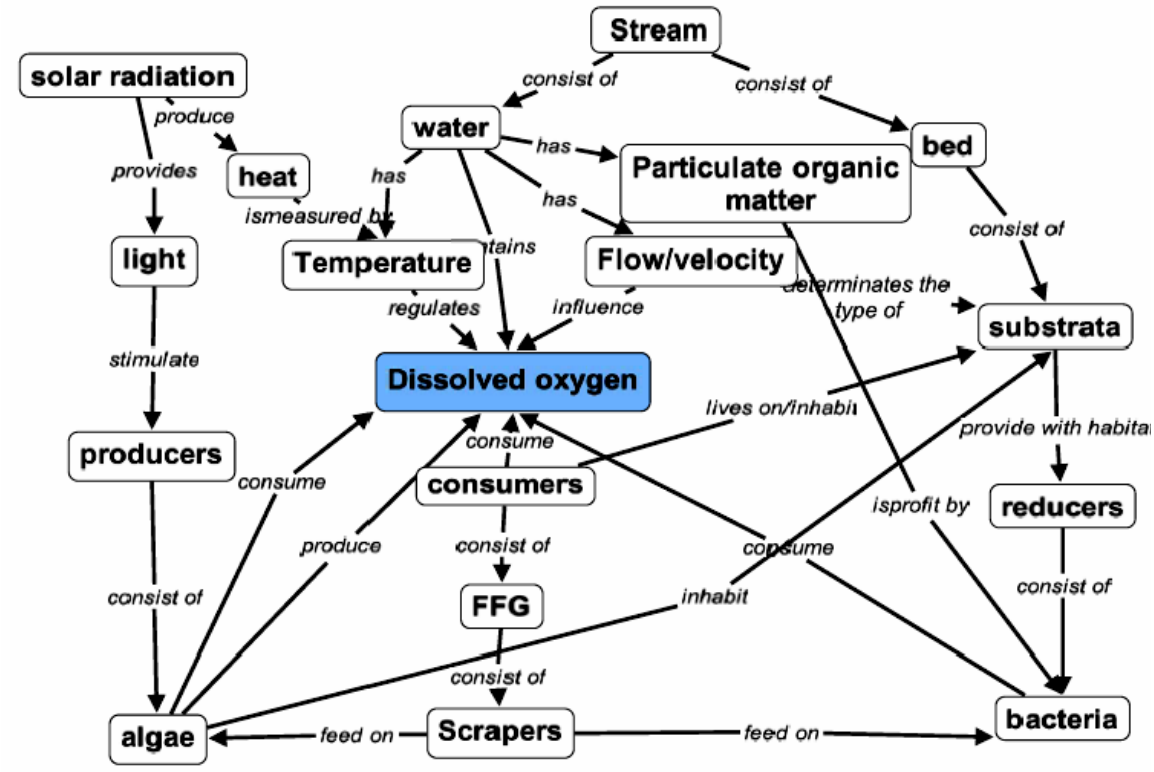


Figure 3. Concept map of the River Mesta ecosystem [from Uznov et al, 2006]

modeler can select a subset of the concept map and connect it to the model without specifying the detailed relationship between the elements.

The above list makes it possible for the modeler to explicitly connect the concept map to the model. Six types of knowledge in concept maps and four types of operations going from concept maps to models, gives a set of twenty-four connection types. Further modeling constraints might make it possible for the environment to automatically detect mismatches and/or incompleteness in the concept map or the model. Reasoning from the model fragments and assumptions [Falkenhainer and Forbus, 1991; Nayak, 1992] might play a key part in operationalizing these constraints.

7 Conclusions

Building models is hard. We argue that a tight integration of the conceptual and detailed modeling processes in the modeling environment can facilitate modeling. We claim that there are six classes of knowledge that are described in concept maps: causal, spatial, mereological, taxonomic, control and parametric. We describe four types of operations that connect concept maps to models: typing, elaboration, filtering and annotation. We believe that this raises interesting research questions about how to provide automatic support for these operations in the modeling environment. Implementing these ideas in a modeling environment like Garp or Stella will provide insights about their usefulness, and we hope that this paper sparks a conversation about building better modeling environments.

Acknowledgments

This work is supported by Artificial Intelligence Program of the Computer Science Division of the Office of Naval Research and by Grant No. IIS-0326059 from the National Science Foundation. Praveen Paritosh would like to thank Ken Forbus, Tom Hinrichs, Matt Klenk, Bert Bredweg, and Andrew Lovett for insightful discussions. Will Bridwell would like to thank Pat Langley, Dorrit Billman, Stuart Borrett, Bert Bredeweg, Anders Bower, and Desiree Tullos for conversations about conceptual modeling.

References

Bogusch, R., Lohmann, B., Marquardt, W. (2001). Computer-aided process modeling with ModKit, *Computers and Chemical Engineering*. Volume 25, Number 7, pp. 963-995.

Bredeweg, B., Salles, P., Bouwer, A., Liem, J., Nuttle T., Cioaca, E., Nakova, E., Noble, R., Caldas, A., Uzunov, Y., Varadinova, Y., Zitek, A. in press, Towards a Structured Approach to Building Qualitative Reasoning Models and Simulations, *Ecological Informatics*.

Bridewell, W., Sanchez, J., Langley, P., Billman, D. (2006). An interactive environment for the modeling and discovery of scientific knowledge. *International Journal of Human–Computer Studies*, 64, 1099–1114.

Buzan, T. (1991). The Mind Map Book . New York: Penguin.

Falkenhainer, B. and Forbus, K. (1991). Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence*, 51, 95-143.

Farabee, M., (2001). On-line biology textbook, retrieved from http://www.emc.maricopa.edu/faculty/farabee/

Forbus, K., Carney, K., Sherin, B., Ureel L., (2004). VModel: A Visual Qualitative Modeling Environment for Middle-school Students, In *Proceedings of the 16th Conference on Innovative Applications of Artificial Intelligence.*

Forbus, K., Lockwood, K., Klenk, M., Tomai, E., and Usher, J. (2004). Open-domain sketch understanding: The nuSketch approach. To appear in *AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural*, October, Washington, DC.

Forbus, K. and Usher, J. (2002). Sketching for knowledge capture: A progress report. In *Proceedings of IUI 2002*, San Francisco, California.

Forrester, J. W. (1961). Industrial dynamics. Waltham, MA

Foss, B. A., Lohmann, B., Marquardt, W. (1998). A Field Study of the Industrial Modeling. *J. of Process Control*, 8(6):325--338.

Haefner, J.W. (2005). Modeling Biological Systems: Principles and Applications, Springer.

Jackson, M. (1995). Software requirements & specifications: a lexicon of practice, principles and prejudices, ACM Press/Addison-Wesley Publishing Co. New York.

Jorgensen, S. (2001). Fundamentals of Ecological Modelling, 3rd edition, Elsevier.

Leake, D., Maguitman, A., Reicherzer, T., Cañas, A., Carvalho, M., Arguedas, M., Brenes, S., Eskridge, T., (2003). Aiding Knowledge Capture by Searching for Extensions of Knowledge Models, *Proceedings of K-CAP ´03*, October 2003, Sanibel Island, Florida.

Leelawong, K. (2005). Using the Learning-by-Teaching Paradigm to Design Intelligent Learning Environments, Doctoral Dissertation, Vanderbilt University.

Levins, R. (1966). The Strategy of Model Building in Population Biology, *American Scientist* 54: 421–431.

Lohner, S., van Joolingen, W., Savelsbergh, E. (2003). The effect of external representation on constructing computer models of complex phenomena, *Instructional Science* 31: 395–418.

Moran T.P., Carroll J.M., (1996). Design Rationale: concepts, techniques, and use.

Novak, J. D. & A. J. Cañas. (2006). The Theory Underlying Concept Maps and How to Construct Them, Technical Report IHMC CmapTools 2006-01, Florida Institute for Human and Machine Cognition.

Nayak, P. P., (1992). Automated Modeling of Physical Systems, Doctoral Dissertation, Stanford University.

Rittel, H., and M. Webber; "Dilemmas in a General Theory of Planning" pp 155-169, *Policy Sciences*, Vol. 4,

Elsevier Scientific Publishing Company, Inc., Amsterdam, 1973.

Sins, P., Savelsbergh, E., van Joolingen, W. (2005). The Difficult Process of Scientific Modelling: An analysis of novices' reasoning during computer-based modeling. *International Journal of Science Education*, Volume 27, Number 14, Number 14/18 November 2005, pp. 1695-1721(27)

Tomai, E., Forbus, K., and Usher, J. (2004). Qualitative spatial reasoning for geometric analogies. *Proceedings of the 18th International Qualitative Reasoning Workshop.*

Uzunov, Y., Elena Nakova, E., and Varadinova, E. 2006. Textual description of river Mesta case study, Naturnet-Redime, STREP project co-funded by the European Commission within the Sixth Framework Programme (2002-2006), Project no. 004074, Project Deliverable Report D6.3.1.

Choosing Abstractions for Hierarchical Diagnosis

Fabien Perrot and **Louise Travé-Massuyès**
{perrot,louise}@laas.fr
0033561336952 0033561336302
LAAS-CNRS
Université de Toulouse
7, avenue du Colonel Roche,
31077 Toulouse Cedex 4
France

Abstract

This paper deals with the choice of abstractions for stating hierarchical diagnosis problems. Generally, hierarchical models are built manually and choosing the appropriate abstractions is quite an empirical science. To tackle this issue, we frame a diagnosis problem as an optimal constraint satisfaction problem (OCSP) and we define abstraction related to two OCSP's, in the structure and in the search space. This allow us to analyse the influence of the abstraction on the temporal computational complexity reduction offered by hierarchical reasoning. Optimal abstractions are shown to be built on the well-known diagnosis concept of potential conflict.

Introduction

In the artificial intelligence community, diagnosis problems are often formulated along the theory of consistency-based diagnosis (Hamscher, Console, & de Kleer 1992), one of the most widely used approaches to model-based diagnosis. However, such a logical formulation is being replaced more and more by a constraint optimisation formulation (Williams & Ragno 2003) which allows one to compare solutions. More precisely, a diagnosis problem is framed as an optimal constraint satisfaction problem (OCSP).

Abstraction has been advocated as one of the main remedies for the computational complexity of model-based diagnosis. It can be viewed as the process of generalisation by reducing the information content of a concept or an observable phenomenon, typically in order to retain only information which is relevant for a particular purpose. Consequently, from a detailed diagnosis problem, one can construct by iterated abstractions more abstract diagnosis problems. These diagnosis problems can be solved together in hierarchical fashion (generally from the most abstract one to the least). Discovered solutions and non-solutions of the i^{th} diagnosis problem can be used to prune the search space of the $i - 1^{th}$ diagnosis problem. Theoretically, temporal computational complexity reduction is exponential. In practice, this reduction notably depends on the choice of the abstract diagnosis problems, and thus depends on the choice of the abstractions. This reduction of complexity is observed in several research fields like symbolic model-checking, planning, constraint solving,.... It can be explained by the fact that abstraction is a way of reasoning about a set of objects (or states) satisfying specific properties rather than directly reasoning on the enumeration of objects. Such sets of states can be proved to be non-solution or solution, thus (if abstraction is well-chosen) it is a proof of solution or non-solution for all the belonging to the sets states. For example, the concept of conflict is by definition a direct proof that a set of states only contains non-solutions. It is more efficient to identify conflicts rather than testing and proving all the states one by one.

The aim of our work is to find, among all possible abstract diagnosis models, the one which gives the greatest complexity reduction. It would allow one to design and implement algorithms that build guaranteed "good" hierarchical diagnosis problems of a system, without the need for expertise. We are especially interested in abstraction techniques for CSP's in order to apply them to diagnosis. Abstraction of CSP's was recently surveyed in (Lecoutre *et al.* 2006) who proposed a new framework to describe general kinds of abstractions. Hierarchical diagnosis based on structural abstraction (which is a special case of abstraction) has been studied in detail in (Chittaro & Ranon 2004). Structural abstraction, from the diagnosis point of view, consists of clustering the components of the system. However, as far as we know, only one publication (Torta & Torasso 2006) in the diagnosis field, has been devoted to the choice of abstractions. The research in (Torta & Torasso 2006) deals with behavioural abstraction [1]. Our approach deals with more general kinds of abstractions.

The rest of this paper is organised as follows. In the second section, we succinctly present the OCSP framework and we recall how consistency-based diagnosis problems can be viewed in that framework. In the third section, we present the hierarchical diagnosis approach and the abstraction concepts. In the fourth and main section, we analyse the influence of the choice of abstractions on temporal computational reduction of hierarchical reasoning. Finally, we conclude and discuss future work.

OCSP's and diagnosis problems

The CSP framework

A constraint satisfaction problem (CSP) is defined as a set of variables and a set of constraints on these variables. Notably,

[1] The authors aggregate component modes but not combinations of modes, a special case of structural abstraction.

it is recognised by the following definition :

Definition 1 *A* CSP *is a pair* (V, C) *where :*

- $V = \{V_1, V_2, \ldots, V_n\}$ *is a set of* variables. *Each variable* V_i *has a non empty* domain $D(V_i)$ *of possible values.*
- $C = \{C_1, C_2, \ldots, C_m\}$ *is a set of* constraints. *Each constraint* C_j *involves some subset* $vars(C_j)$ *of* V *and specifies the allowable combinations of values for that subset.*

A state *of the problem is an assignment to all the variables in* V *:* $(V_1 = v^1, V_2 = v^2, \ldots, V_n = v^n)$. *A* partial state *of the problem is an assignment to only some variables (but not all) in* V. *A* consistent *assignment is one that does not violate any constraints. A* solution *to a CSP is a consistent state. A state is sometimes called a complete assignment and a solution is sometimes called a consistent and complete assignement.*

The structure of a CSP is given by a constraint hypergraph *where the nodes of the hypergraph correspond to variables of the problem and the hyperarcs correspond to constraints.*

The OCSP framework

An OCSP is a CSP for which one is allowed to compare solutions thanks to a cost function; and it is adapted to diagnosis in the sense that one is only interested in values of a subset of variables called decision variables. More formally :

Definition 2 *An* optimal constraint satisfaction problem (OCSP) *consists of a* $CSP = (V, C)$*, a set of* decision variables $W \subset V$*, and a* cost function $g : W \rightarrow \mathbb{R}^2$. *The remaining variables* $V - W$ *are called* non-decision variables. *An assignment of decision variables is called a decision state. A* solution *to an OCSP is a minimum cost decision state that is consistent with the CSP.*

For more details about OCSP's, please refer to (Williams & Ragno 2003).

The diagnosis problem

In the consistency-based diagnosis approach (Hamscher, Console, & de Kleer 1992), a model of the system to be diagnosed is used. This model is component-centered. Each component has a set of possible modes. Behaviour of the system in each mode is described with formulae. Observables are also provided with formulae. Given this model, the task of solving the diagnosis problem consists of finding those modes of the components which are consistent with observations. More formally, the definition of a diagnosis model in the DX community (Hamscher, Console, & de Kleer 1992) is the following :

Definition 3 *A diagnosis model is a triple* $(SD, COMPS, OBS)$ *where :*

- SD *is the system description. It consists of a set of first-order logic formulae which describe the behaviour of the system.*
- $COMPS$ *is a set of constants which represent the components of the system.*
- OBS *is a set of first-order logic formulae which represent the observations given by the sensors.*

The diagnosis problem in the OCSP framework

Like some authors (Williams & Ragno 2003), we argue that a diagnosis problem can be framed as an OCSP (the model) where one must find the n best assignments of mode variables consistent with the constraints describing the system (the task). Decision variables are the mode variables. Solutions are the diagnoses. Non-decision variables are all other variables (including observable and non-observable variables). This has a significant impact because non observable variables may have infinite domains. Considering mode variables as decision variables provides a kind of abstraction that turns an infinite domain CSP into a finite domain CSP. Each mode variable m_{C_i} is attached to one component C_i.

Example The formulation as an OCSP of the diagnosis problem of the classic polybox toy example in the DX community is given in the following.

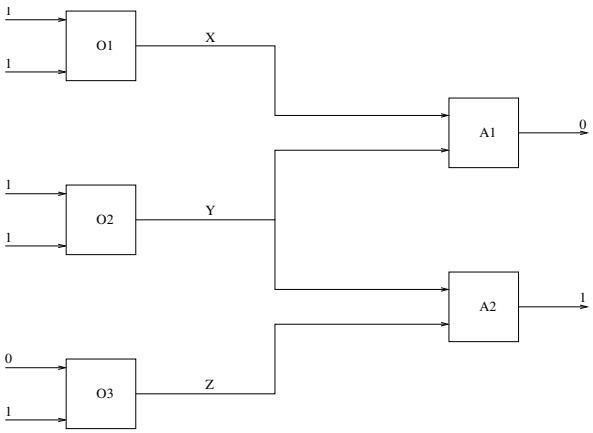


Figure 1: Topology of the boolean polybox

The topology of the polybox is represented in figure 1. O_1, O_2, O_3, A_1, A_2 are the components of the system. O_1, O_2, O_3 are OR gates and A_1, A_2 are AND gates. X, Y, Z are non-observable variables; each of those can take values in $\{0, 1\}$. $m_{O_1}, m_{O_2}, m_{O_3}, m_{A_1}, m_{A_2}$ are mode variables associated to the components; each of those can take values in $\{G, B\}$ (G for Good and B for Bad). A set of constraints link mode variables, observable and non-observable variables. When the observable variable values are provided, one can always instantiate observable variables rather than keeping them in the model. This is why there are no observable variables in the polybox constraint model below.

For sake of clarity we use a constraint language which explicitly describes assignments of values to variables (we could have written the constraints in pure propositional logic). The constraints for the polybox are :

- $(m_{O_1} = G) \implies (X = (1 \vee 1))$
- $(m_{O_2} = G) \implies (Y = (1 \vee 1))$
- $(m_{O_3} = G) \implies (Z = (1 \vee 0))$
- $(m_{A_1} = G) \implies (0 = (X \wedge Y))$
- $(m_{A_2} = G) \implies (1 = (Y \wedge Z))$

These constraints can also be represented in extension because, in this example, the domains of all the involved variables are finite. For instance, the first constraint can be rewritten : $\{(O_1, X), \{(G, 1), (B, 0), (B, 1)\}\}$.

Finally, the diagnosis problem of the polybox in the OCSP framework is :

- decision variables : $\{m_{O_1}, m_{O_2}, m_{O_3}, m_{A_1}, m_{A_2}\}$. Each associated domain is $\{G, B\}$,
- non-decision variables : $\{X, Y, Z\}$. Each associated domain is $\{0, 1\}$,
- constraints are those described above.
- cost function is $g(m_{C_i} = v_i) = \prod_j P_j(m_{C_i})$. Cost represents the candidate probability. The component mode probabilities $P_j(m_{C_i})$ are combined with multiplication because faults on different components are assumed to be independent.

The associated constraint hypergraph is represented in figure 2.

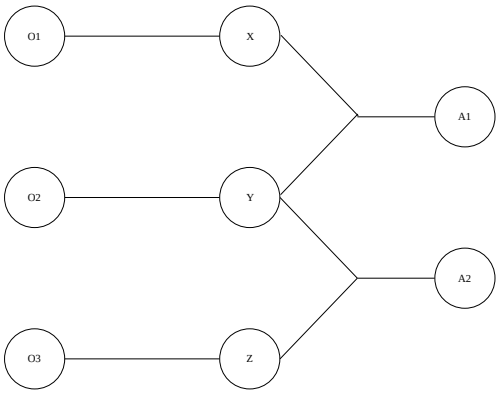


Figure 2: Constraint hypergraph of the boolean polybox

Hierarchical diagnosis and abstraction

Hierarchical diagnosis reasoning

Hierarchical diagnosis reasoning consists of solving a diagnosis problem using an ordered list of diagnosis problems. Generally, this is initiated by a diagnosis problem P_0 and builds up to a more abstract diagnosis problem P_1. Iteratively, one can build an ordered list of n diagnosis problems. Solving these problems is commonly achieved in a top-down fashion. First, the most abstract problem P_{n-1} is solved. Then (or at the same time), knowledge of solutions and non-solutions of P_{n-1} is used (through abstraction) to prune the search space of the problem P_{n-2}. By iterating, one can solve the original diagnosis problem P_0. Results from solving problem P_k can be useful for solving problem P_{k-1} in different ways. It depends on the kind of abstraction which has been used to build problem P_k.

Abstraction concepts

In this section, different views of abstractions are described, each being more or less appropriate to exhibit properties interesting for diagnosis.

Topological view of abstractions (mapping of components) The topological view of abstractions relies on a component-centered representation. Graphically, as shown in figure 3, components are represented by nodes and labeled by their name. The links between components are represented by arcs (direction is chosen according to causality). Each arc is labeled by a value (observable variable) or the name of a non obervable variable. Some abstractions interpret naturally according to the topological view but others cannot be represented in this way. When such a representation exists, topological views of the concrete model and of the abstract model are those giving the best intuition of the abstraction.

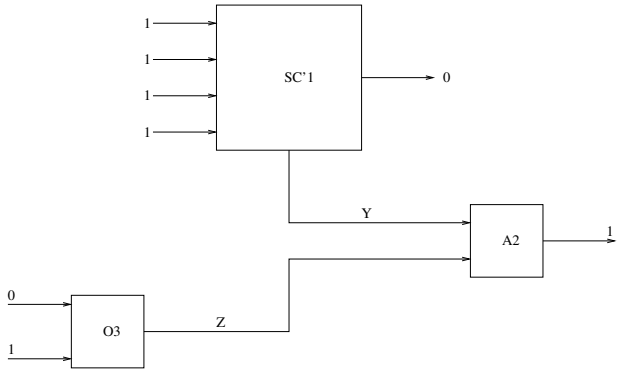


Figure 3: Topology of the abstract boolean polybox.

Example In the polybox example, let us consider a simple structural abstraction (a special case of abstractions) which consists of aggregating for instance the components O_1, O_2 and A_1 in one supercomponent SC'_1. The topology of the abstract polybox is represented in figure 3. But to completely define this abstraction, the mapping between possible values of $(m_{O_1}, m_{O_2}, m_{A_1})$ and $m_{SC'_1}$ must be specified. As an example, a natural mode value mapping is :

- when $(m_{O_1} = G, m_{O_2} = G, m_{A_1} = G)$, then $m_{SC'_1} = G$,
- $m_{SC'_1} = B$ in other cases.

The topological view cannot capture all the abstraction information because the mode value mapping does not explicitely appear.

This mode value mapping is represented in figure 4 and corresponds to the projection on $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{A_1}) \rightarrow D(m_{SC'1})$ of an interpretation mapping as defined in (Nayak & Levy 1995). We can build any value mapping among the 2^8 possible mappings defined on $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{A_1}) \rightarrow D(m_{SC'1})$.

It may be as well the case that mode variables can take more than two values. Some mappings can be more efficient than others, that is discussed in the fourth section. More general kinds of relations than mappings have been deeply described in (Lecoutre *et al.* 2006) but here, just mappings are considered.

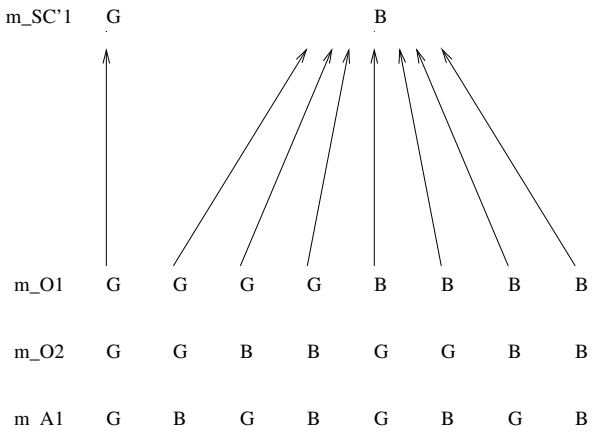


Figure 4: A simple value mapping defined on $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{A_1}) \rightarrow D(m_{SC'1})$.

Limits of the topological view In the last example, one can see that structural abstraction can naturally be defined by two choices :

- aggregation of variables (topological view),
- aggregation of mode value tuples.

However, let us also notice that if we first choose an aggregation of mode value tuples, it implies an aggregation of variables; but the converse does not hold. This remark shows us that criteria for choosing good abstractions, even if it is in the structural abstraction case, can be more easily found by looking directly at the aggregation of mode value tuples (and not at the topological view). This leads us towards a new view of abstractions which corresponds to the search space view.

Search space view (mapping of states) The search space of a diagnosis problem is the set of possible complete assignments to mode variables.

An extended mode value mapping (Lecoutre *et al.* 2006) (Nayak & Levy 1995) associates concrete states to abstract states. This extended mode value mapping can be found extending a mode value mapping to all mode variables. The extended mapping, because its domain spawns on the whole search space, permits to evaluate the number of steps needed by a hierarchical diagnosis algorithm to solve a diagnosis problem. Given that the mode value mapping is generally a surjective mapping (as in the polybox example), the extended mode value mapping is also surjective. Consequently, for each state of the abstract search space, the set of concrete states that correspond to it can be found thanks to the preimage of the extended mapping. The number of elements in this set is called the branching factor of an abstract state. Let us remind to the reader that the preimage of a subset B of the codomain Y under a function f is the subset of the domain X defined by $f^{-1}(B) = \{x \in X | f(x) \in B\}$. The preimage exists even if the mapping is not bijective.

The search space view is difficult to represent by a scheme when the number of components is high because the number of concrete states is exponential in the number of components. In this paper, it is exactly 2^n states for n components. In our polybox example, the 2^5 concrete states are abstracted into 2^3 abstract states by the extended mapping. The latter that we note EM can be found given the mode value mapping M represented in figure 4 in the following manner :

- the domain of EM is $D(m_{O_1}) \times D(m_{O_2}) \times D(m_{O_3}) \times D(m_{A_1}) \times D(m_{A_2}) \rightarrow D(m_{SC'1}) \times D(m_{O_3}) \times D(m_{A_2})$,
- EM maps a concrete state $S = (m_{O_1} = v_1, m_{O_2} = v_2, m_{O_3} = v_3, m_{A_1} = v_4, m_{A_2} = v_5)$ to an abstract state $ES = (m_{SC'1} = M(m_{O_1} = v_1, m_{O_2} = v_2, m_{A_1} = v_4), m_{O_3} = v_3, m_{A_2} = v_5)$.

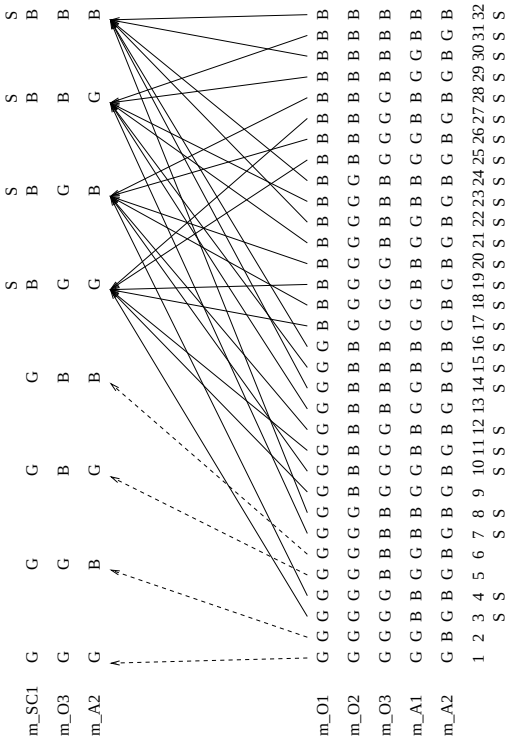


Figure 5: The extended mapping for a structural abstraction of the polybox.

The extended mode value mapping EM is represented in figure 5. Each arrow represents a mapping from a concrete state to an abstract state. The symbol S denotes a state which is solution. One can see that there are 4 solutions to the abstract problem and 26 solutions to the concrete problem.

Constraint view (mapping of constraints) The constraint view of an abstraction is established by the constraint hypergraphs of the concrete and abstract OCSP's.

The constraints of the concrete problem are abstracted into other constraints. In the example of the polybox, constraints 1, 2 and 4 are merged into one constraint : $(m_{SC'_1} = G) \implies ((X = (1 \vee 1)) \wedge (Y = (1 \vee 1)) \wedge (0 = (X \wedge Y)))$. Constraints 3 and 5 are preserved.

The search space view and the constraint view correspond to semantic and syntactic abstractions as defined in (Nayak & Levy 1995), respectively.

Influence of abstractions on computational complexity

The word "best" in "best abstraction" is used for minimal computational temporal complexity, in other words for the minimum number of steps to solve the diagnosis problem. For this analysis, we consider that abstract diagnosis problem(s) are precomputed, so we do not take into account the temporal computational complexity of constructing a hierarchical model of the system to be diagnosed (which is a reasonable hypothesis in the case of on-line state-tracking applications).

The analysis proceeds as follows : firstly, the set of solutions (hence the set of non solutions) is supposed to be known. This permits us to give criteria of "good" abstractions in the space search view; secondly, these criteria are interpreted in the topological and constraint views to determine how they can be used to precompute "good" abstractions that are garantied to be efficient.

Analysis in the search space view

Among all possible abstractions, two general kinds have been identified in the litterature (Nayak & Levy 1995) (Chittaro & Ranon 2004) (Lecoutre *et al.* 2006). :

- Concrete Solution Increasing (CSI),
- Concrete Solution Decreasing (CSD).

Definition 4 (CSI abstraction) *An abstraction is CSI iff for all concrete states which are solutions of the concrete problem, their corresponding abstract state (w.r.t extended mode value mapping) is solution of the abstract problem.*

From this definition, one can trivially deduce the following proposition :

Theorem 1 *Consider a CSI abstraction, then if an abstract state is shown not to be solution, then all its correponding concrete states are not solutions.*

For example, the structural abstraction of the polybox mentioned above is CSI, one can verify it in figure 5.

Definition 5 (CSD abstraction) *An abstraction is CSD iff for all concrete states which are not solutions of the concrete problem, their corresponding abstract state (w.r.t extended mode value mapping) is not solution of the abstract problem.*

From this definition, one can trivially deduce the following proposition :

Theorem 2 *Consider a CSD abstraction, then if an abstract state is shown to be solution, then all its correponding concrete states are solutions.*

A general and simple hierarchical diagnosis algorithm begins from the most abstract problem, generates and tests all the states of a level n in the abstraction hierarchy, then goes down to the level $n-1$, and iteratively, the algorithm finishes to test all the concrete states of the level 0 to give the solutions of the concrete problem.

In this paper, we consider two levels in the abstraction hierarchy but the results can be extended to more than two levels.

Let us recall to the reader that without abstraction, with a simple-minded diagnosis algorithm which just generates and tests states, the temporal computational complexity is $O(n)$ where n is the number of states of the diagnosis problem, i.e. exponential in the number of components.

Consequently, the temporal computational complexity of the hierachical diagnosis algorithm described above depends on two factors :

- the number of abstract states,
- the number of concrete states.

The number of concrete states cannot be changed. An ideal abstraction would consist of two abstract states a and b. a maps all the states which are solutions and b maps all the states which are not. Let us notice that the ideal abstraction we propose is CSI and CSD. So an abstract solution is sufficient to represent all the solutions of the concrete problem; and a non solution abstract state rules out all the non solution concrete states. Consequently, from the CSI and CSD propositions, one can deduce the following proposition :

Theorem 3 *An abstraction that is CSI and CSD reduces the complexity of the diagnosis problem to $O(n')$ where n' is the number of abstract states. The ideal CSI and CSD abstraction has two abstract states.*

But this ideal abstraction cannot be easily built for two reason :

- which states are solutions and which are not is not known in advance,
- constraints associated to the abstract states may not exist.

Refering to the first issue, solutions are supposed to be known (for the analysis) to target which abstractions can easily capture solutions and non solutions. We tackle the second issue using the constraint space view to exhibit abstract constraints which can be easily built.

It appears that the ideal abstraction does not exist in the general case. However one can capture all the solutions and non solutions of a problem using several abstractions of the same problem.

Example

For the polybox example, there are 32 states and, among them, 26 solutions. The diagnosis community is used to represent the 26 solutions by 3 minimal diagnoses : $\{O_1\}, \{A_1\}, \{O_2, A_2\}$. These minimal diagnoses represent 16, 13 and 8 solution states, respectively. Some states are represented by several minimal diagnoses. Non solution states can be captured by minimal conflicts. Minimal conflicts $\{O_1, O_2, A_1\}$ and $\{O_1, A_1, A_2\}$ each captures 4 non solutions.

The ideal abstraction of the polybox example maps the 26 solutions to one abstract state a and the 6 non solutions to another abstract state b. But with such a partition of the search space, building abstract constraints means solving diagnosis problem. However, given that we want to build hierarchical model for all possible set of inputs, this approach is not feasible. One may however notice that using abstract states corresponding to minimal potential conflicts (Cordier

et al. 2004), we can build a relatively efficient hierarchical model for all sets of inputs. All non solutions are captured and ruled out in an efficient way. This idea is developped in the next subection.

Using constraints to find "best" abstractions

We can relax the ideal abstraction requirement using several "two abstract states-based" abstractions to capture the whole search space. One needs, for each abstraction, one abstract state which can represent the highest number of non solutions and another abstract state to represent the highest number of solutions.

To choose these states, the topological and constraint views of abstractions are used. Choosing abstractions in the search space view may lead to non existant abstract CSPs, i.e. for which one cannot find the corresponding variables and constraints.

To represent a maximal number of states, one has two main choices :

- aggregating variables and mode values as described in the topological section,
- removing variables and/or values.

In the constraints view, aggregating variables corresponds to merging constraints and removing variables corresponds to removing constraints. The first option exactly corresponds to structural abstraction and one can see in the polybox example that with natural mappings, abstraction is CSI but does not permit to rule out a lot of non solution states in one test. For the second option, removing n variables permits one to represent 2^n states. So, one has to remove the highest number of variables while keeping the constraints decidable. Removing variables means removing their associated constraints. Consequently, one must find the smallest set of constraints which remain decidable. These sets are hence just overdeterminated, i.e. they involve n equations for $n-1$ unknowns, and they are well-known in the diagnosis field as corresponding to minimal potential conflicts (Pulido & Alonso 2002) (Cordier *et al.* 2004).
When the test on a given OBS of one such just overdeterminated constraint set does not pass, the conjunction of assignments to G of the involved mode variables is a minimal conflict. The case in which at least one variable is assigned to B does not help for finding solutions since the constraints are always satisfied. It is easy to show that when using a minimal potential conflict to construct an abstraction, this abstraction is CSI but not CSD. It is also known that all minimal potential conflicts capture all concrete non solutions. So checking all the abstract states built this way garanties to rule out all the non solution states. However, one abstraction per minimal potential conflict is necessary since one concrete state may correspond to more than one minimal potential conflict. Indeed, we have restricted our analysis to mappings between concrete states and abstract states. With general relations, we conjecture that one abstraction can contain all minimal potential conflicts but in this case, one concrete state may have more than one image.

Conclusion and future work

Hierarchical diagnosis efficiency strongly depends on the choice of the abstractions used to build a hierarchical model of the system. For analysing the influence of this choice, three complementary views of abstractions have been defined : the topological view, the search space view and the constraint view. In the case in which abstractions are built removing variables or contraints, we argue that most efficient abstractions are obtained from minimal potential conflicts because they rule out all non solution states.
But finding efficient abstractions has a cost. For instance, determining minimal potential conflicts is known to be of exponential complexity. So our future work will focus on defining an efficiency measure for an abstraction that takes in account not only the computational gain on the hierarchical model but also the cost of finding the model. In particular, we will pay much attention to structural abstraction because it is generally cheap; and we will consider building a bridge between structural abstractions and other types of abstractions (like conflict-based) that all together may boost the diagnosis resolution process efficiency.

References

Chittaro, L., and Ranon, R. 2004. Hierarchical model-based diagnosis based on structural abstraction. *Artif. Intell.* 155(1-2):147–182.

Cordier, M. O.; Dague, P.; Levy, F.; Montmain, J.; Staroswiecki, M.; and Travé-Massuyès, L. 2004. Conflicts versus analytical redundancy relations : a comparative analysis of the model-based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man and Cybernetics.* 34(5):2163–2177.

Hamscher, W.; Console, L.; and de Kleer, J. 1992. *Readings in model-based diagnosis*. Morgan Kaufmann.

Lecoutre, C.; Merchez, S.; Boussemart, F.; and Grégoire, E. 2006. A csp abstraction framework. *Revue d'intelligence artificielle* 20(1):31–62.

Nayak, P., and Levy, A. 1995. A semantic theory of abstractions. In Mellish, C., ed., *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 196–203. San Francisco: Morgan Kaufmann.

Pulido, B., and Alonso, C. 2002. Possible conflicts, arrs, and conflicts. *Proceedings of the Thirteenth International Workshop on Principles of Diagnosis (DX 02)*.

Torta, G., and Torasso, P. 2006. Qualitative domain abstractions for time-varying systems: an approach based on reusable abstraction fragments. In *Proc 17th Workshop on Principle of Diagnosis (DX 06)*.

Williams, B. C., and Ragno, R. J. 2003. Conflict-directed a* and its role in model-based embedded systems. *Journal of Discrete Applied Math*.

Arguing About Radioisotope Dating

Laura Rassbach*, Elizabeth Bradley*, Kenneth Anderson*, Christopher Zweck^, and Marek Zreda^

* University of Colorado
Department of Computer Science
Boulder, CO 80309-0430
laura.rassbach@colorado.edu, lizb@cs.colorado.edu,
kena@cs.colorado.edu

^ University of Arizona
Department of Hydrology and Water Resources
Tucson, AZ 85721
czweck@hwr.arizona.edu, marek@hwr.arizona.edu

Abstract

We present a prototype of AICronus, an argumentation system that automates a challenging reasoning process used by experts in cosmogenic isotope dating. The architecture of the system is described and preliminary results are discussed.

1. Introduction

Scientific reasoning is a complex process, alternately requiring flashes of insight and tedious analysis. This dichotomy is evident in the process of determining an age for a landform using cosmogenic isotope dating. Experts in the field of cosmogenic geochronology frequently spend months on repetitive mathematical tasks, until they have gathered enough information to suddenly understand the data. The AICronus project is aimed at understanding and automating this process.

Automating this reasoning process is challenging because the science of cosmogenic isotope dating is quite new. There is only a relatively small number of completed, detailed analyses to draw knowledge from. Therefore, it is necessary to build a knowledge base through interaction with experts. Unfortunately, it is often difficult for experts to clearly articulate how and why they come to specific conclusions. We have been working with experts in this field for more than two years, but our knowledge base is still very incomplete. Complicating the difficulty of acquiring the knowledge used by experts, the reactions that produce cosmogenic isotopes are not fully understood, leading to incompletely understood theories about them. As a result, most expert analysis relies on heuristics that are sometimes imprecise and frequently contradictory. Any system that automates the geological timeline construction process, then, must be able to handle both contradiction and uncertain heuristics gracefully.

AICronus addresses these issues through the use of a nonmonotonic logic called argumentation. Argumentation uses symbolic logic, so that rules acquired from experts can be directly input into the system. In addition, the system's reasoning can be presented to the user in a legible format, facilitating engagement and speeding further knowledge engineering. In the argumentation architecture used by AICronus, conclusions can receive partial support (modeling uncertain heuristics) and support for a conclusion can be defeated by contrasting evidence or rules (handling contradiction gracefully).

This paper presents the prototype version of the AICronus system along with some preliminary results, which show significant initial success in accurately modeling the reasoning process of isotope dating experts. Section 2 details the process of constructing a chronology for a landform using cosmogenic isotope dating. Section 3 discusses the particular challenges that arise in attempting to automate parts of this process. Section 4 demonstrates how argumentation addresses these challenges. Section 5 discusses the AICronus architecture in more detail. Section 6 walks through a concrete example of the working system. Section 7 discusses future work for the AICronus system, and section 8 covers related work.

2. Cosmogenic Isotope Dating

Cosmogenic isotope dating is a method for computing the age of a landform using cosmogenic isotope measurements of samples taken from that landform. Other methods for landform dating rely heavily on heuristic examination of features such as lichen growth (Bradwell 2001) or on dating materials associated with the lanform such as fossils (Noller, Sowers, and Lettis 2000). Cosmogenic isotope dating is more consistent and less subject to influence from the preconceptions of individual geologists.

This dating procedure is based on the knowledge that cosmic ray particles hit the earth at a fairly constant rate. When these particles come into contact with certain target elements, they can change them into cosmogenic isotopes (e.g. Chlorine-36 or Aluminum-26). The creation of these isotopes happens at a calculable rate. Most types of cosmic rays penetrate only a few meters, so the radioactive isotopes are generated almost exclusively at the surface. This knowledge enables a geologist to determine how long a particular sample has been at the surface based on the number of radioactive atoms present, the sample's chemistry, and other factors. The mathematics involved are

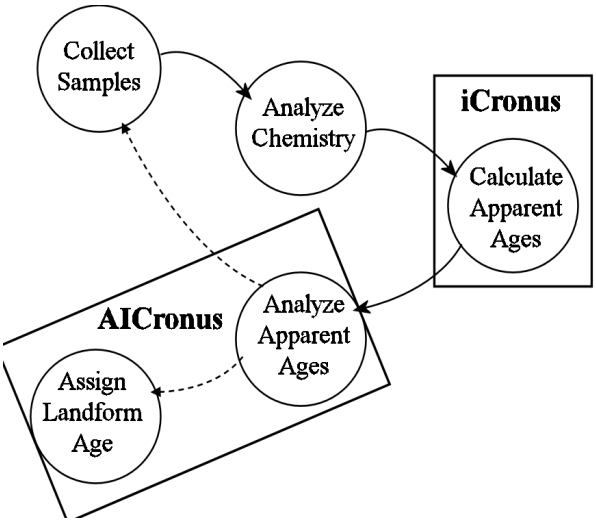


Figure 1: The process of assigning a landform's age.

quite complicated, and are handled by a different system being developed by the iCronus project at CU Boulder (Anderson and Bradley 2006).

For many landforms, the length of time a sample has been at the surface is actually a measure of the age of the landform (e.g. moraines, which are formed by glaciers carving boulders from deep underground and eventually depositing them, along with soil, as the glacier retreats). Other landforms are formed over a longer period of time, or from rock that was at the surface prior to the landform's formation. In these cases the length of time that samples have been at the surface can provide other information (e.g. how long the landform took to form) but will not give the actual age of the landform. Currently, cosmogenic isotope dating is used primarily to estimate ages of suddenly-created landforms.

The process of cosmogenic isotope dating begins with an expert taking samples of surface rock—generally thin chips from several boulders—from a single landform. Significant expertise is needed to choose good samples: sample boulders should not be excessively weathered, should show no signs of having been rolled or turned, should usually be of similar composition to the surrounding surface, etc. In many cases it will only be possible to take a small number of samples that meet these requirements. Experts also record as much data as possible about the location and status of the samples, including the sizes of the boulders they are collected from, the amount of visible sky, and the sample's exact location.

After collection, the expert sends the samples to an accelerator mass spectrometry (AMS) lab that measures the chemical properties of the samples. This includes the chemical composition and the percentage of certain isotopes (e.g. Chlorine-36 compared to overall Chlorine). The lab's services are extremely costly, further limiting the number of samples for which data are available. Based on these measurements, information about where the sample was taken, and a large amount of background knowledge, the expert calculates preliminary (or "apparent") ages for the samples. The background knowledge involved includes data about changes in cosmic ray intensity, changes in sea level (which affects cosmic ray intensity at particular altitudes), and information about the production rates of the isotope in question from other isotopes (Gosse and Phillips 2001). For many of these background data, multiple measurements are available and will yield different results. Handling these calculations and organizing the background data is the task of the iCronus project (Anderson and Bradley 2006).

Next, the expert compares the preliminary individual ages. If all the apparent sample ages for a single landform are the same, within the margins of error introduced by the AMS analysis, that age is assigned to the landform and the process is complete (the expected distribution is a Gaussian, the typical expectation for events affected by random errors). However, this happy situation rarely occurs. It is more usual for preliminary age measurements for different samples to differ by as much as 10,000 years (Shanahan and Zreda 2000). In this case, the expert attempts to explain the divergence so that s/he can assign a single age to the landform. The AICronus system is designed to assist with this explanation.

Sometimes there is no good explanation, or there are several explanations that cover the data equally well. Often the result of this first round of analysis is the conclusion that more samples are needed. In this situation, experts often reason from the existing data to guide further sampling–for instance, when it is not possible to determine whether erosion or inheritance caused a spread in ages, experts may take a soil sample or sample a landform at depth to obtain more information. This reasoning is not captured by the prototype version of AICronus described here; ideas for adding it to the deployed version are discussed in Section 7. Figure 1 illustrates this cyclic process.

3. Automation Challenges

Most explanations for spread in apparent ages come from a short list of about fifteen geologic processes that affect the preliminary exposure times of samples from a single landform. For example, erosion gradually exposes new surfaces, causing some samples to have apparent ages much younger than the age of the landform. A "process" called inheritance affects samples that were exposed before the landform in question was formed, giving them apparent ages older than the age of the landform. Other processes include cover by snow or vegetation, gradual formation such as from soil deposits, or earthquakes, which may suddenly expose large amounts of rock at the surface. Multiple processes may act on a single landform. Finally, possibilities like lab error and mis-sampling must also be taken into account.

Although we need only consider explanations from among a relatively small number of processes, the complexities of how these processes affect the data—especially when multiple processes are involved—make this task far from simple. Data are noisy and frequently

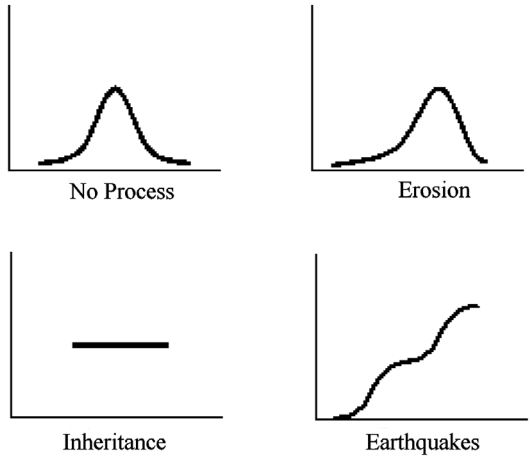


Figure 2: Apparent-age distributions (number of samples vs. apparent age) produced by various processes.

cannot be trusted (experts may have mis-identified the type of landform they are considering, for example), and the manifestation of one process may be quite similar to the manifestations of other processes.

Many processes that affect apparent ages of samples give the apparent age distribution a characteristic shape. When no geologic process has affected the samples, random errors in measurement are expected to produce a Gaussian distribution, with the true age of the landform equal to the peak of the curve. Matrix erosion of a moraine, which gradually exposes new boulders as the top soil of the moraine erodes, looks something like a skewed bell curve with the peak towards the older end of the scale. On the other hand, inheritance usually involves a simple uniform distribution over an age range. Figure 2 shows some examples of these distributions. Unfortunately, it is difficult to diagnose the process that affected a particular landform from the distribution of apparent ages because we rarely have enough samples to see the distribution shape. Instead, experts usually perform this diagnosis using heuristics about how various types of landforms are formed and how each process affects different landforms. Other heuristics are applied to the apparent distribution of the small number of samples: experts label three samples that are approximately evenly spaced erosion—not inheritance, as we would expect based on the *a priori* knowledge of sample distributions. The experts we are working with have not yet been able to explain this apparent contradiction.

Choosing which geologic process is responsible for the data is complicated by the fact that there is generally some evidence both for and against several processes, a major challenge for automatic analysis. From our observation of experts in this field, it appears that a standard approach to this problem is to select one process and look for evidence both for and against that process. If it is possible to gather enough evidence in favor of a process and not possible to gather a similar or greater amount of evidence against the process, then it is considered a good candidate for explaining the data.

4. Solution: Argumentation

The specific task of AICronus is to assist with the analysis of apparent sample ages and help determine what processes are good candidates to explain the spread in apparent ages. As just mentioned, many of the heuristics that experts use in this process are vague and sometimes contradictory. For example, matrix erosion is expected to produce a skewed bell curve of initial sample ages. In practice, however, experts usually assign matrix erosion to cases that have a uniform distribution (Shanahan and Zreda 2000). To further complicate matters, inheritance is the process expected to produce a uniform distribution! Contradictions may also arise when input observations are incorrect in some way, for example when samples are entered as members of one landform but have actually come from two different landforms (Desilets and Zreda 2006). Heuristics like "this is a moraine, so inheritance is more likely" are also common and are clearly not absolute; we do not always conclude inheritance when the landform is a moraine. Therefore AICronus must gracefully handle both contradiction and partial support.

In addition to these technical issues, experts are unlikely to agree with any conclusions made by AICronus unless they understand the reasoning behind those conclusions. Thus it is also critical to the usefulness of AICronus that it be capable of convincingly presenting the reasons for its conclusions. This capability provides the additional benefit that students of geology can examine the reasoning and heuristics that are used in selecting a process.

Argumentation systems are a good solution here. They provide the functionality needed for AICronus to be useful to both experts and students in cosmogenic isotope dating. They are capable of handling contradictory rules and input data, partial support for conclusions, and can report their reasoning in a clear and understandable way (Krause, Ambler, Elvang-Gøransson, and Fox 1995) (Doyle 1983). In fact, the reasoning used in argumentation appears to closely match the flexibility and methodology that experts in the field actually use in their analyses.

The argumentation framework used by AICronus is based on the Logic of Argumentation introduced by Krause et. al. (Krause, Ambler, Elvang-Gøransson, and Fox 1995). Unlike in traditional first-order logic systems, rules, input data, and "proofs" in argumentation systems may all be considered defeasible. Proofs in classical logic correspond to arguments in these systems—an argument is a reason for believing some conclusion, but contradictory arguments may also be formed.

Krause et. al. implement an argumentation system as an extension to a limited form of first order logic, where rules and data are labeled with a confidence level used to determine which of two arguments is stronger. As arguments are built, the confidences propagate to their conclusions using some system of combination (defined as

appropriate to individual problems). Arguments are labled with the rules and data used to form them. The logic is limited because the 'not' operator is not implemented and disjunction is handled only implicitly by defining multiple rules that can apply to the same conclusion.

The confidence values in AICronus are in the range [-1, 1], and are currently combined using several different functions, selected by which rule is being used. Negative confidence in some literal is interpreted as confidence in the negation of that literal (zero confidence implies the system knows nothing about a term). Conclusions generated by the system are labeled with the arguments for and against them, so that as new information is discovered, the arguments about a conclusion can be examined and possibly defeated. An argument can be defeated in two ways: a stronger argument can be found against the conclusion of the argument (rebuttal), or arguments can be found against the evidence used in the defeated argument (undercutting).

AICronus treats the arguments for and against a particular conclusion like grains of sand on a scale. Stronger arguments, formed using rules and data with higher confidence levels, add more weight to their side of the balance. However, a large enough number of poor arguments on one side of the scale can overpower a single good argument on the other side. Unlike a balance loaded with sand, additional "weight" is added in a system of decreasing returns: two poor arguments of the same quality have less weight than one argument of twice their quality. That is, if a single argument has a confidence of 0.8, it will defeat two combined, rebutting arguments, each with a confidence of 0.4 (but it will be defeated by three such arguments). Undercutting is handled by reducing the degree of confidence in the undercut argument.

5. Constructing Arguments

AICronus takes as input all available data about a set of samples, along with information about the site where they were collected. This includes both qualitative data (e.g. the type of landform and the color of the boulder the sample was taken from) and quantitative data (e.g. the calculated apparent sample ages and the elevation of the landform).

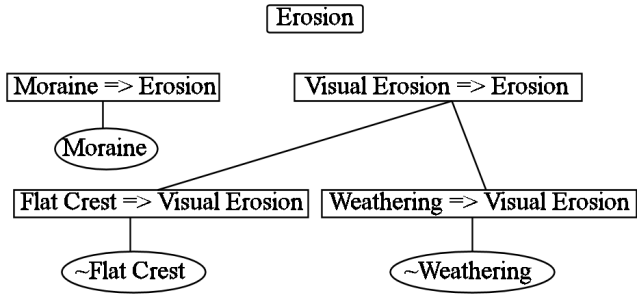


Figure 3: An example AICronus argument. At the top is the conclusion being argued about. Beneath is a collection of trees arguing about this conclusion. Rules are shown in boxes and entered observations in ovals.

Information about nearby landforms may also be included in the input, since the ages of these landforms may imply strict upper or lower bounds on the age of the landform in question (e.g. moraines must decrease in age as the elevation in a single valley increases because of the way they are formed).

AICronus generates a list of processes that may have affected the landform, with more-common processes (as specified by experts) higher on the list. Arguments for and against each process on the list are generated via backward chaining, building an argument tree. Once a process has been found for which the "pro" evidence significantly outweighs the "con" evidence, the system stops and reports its results to the user. These results include all of the processes so far considered and the complete arguments for and against each process. Processes with the most convincing arguments are listed first. The user can choose to generate arguments for more processes if s/he finds the presented results insufficiently convincing.

5.1. Arguments

Rules in AICronus have a standard first-order logic structure, where a rule is written in the form $A => C$. A may be a single literal or the conjunction of several literals. An argument is a collection of trees, with rules from the system's database forming the nodes of the trees. Rules in child nodes have the same variable in their conclusions as one of the literals on the antecedent side of the parent node's implication. At the root of each tree is a single rule that allows AICronus to argue about whether some particular process is responsible for the observed data. The leaves of the trees are drawn from the observations entered by the user.

Figure 3 shows an example collection of argument trees in AICronus which might read: "Erosion is a likely explanation because moraines are likely to erode and this landform is a moraine. However, there is no visual evidence of erosion such as a flat crest or weathering, making erosion a less convincing conclusion." The total confidence in the argument is determined by the total confidence in the trees that argue for the root process versus the total confidence in the trees that argue against the root process.

5.2. Rules

AICrounus rules have a number of parts. Rules are implicitly built around the classic implication structure from first-order logic. In addition, they contain guards to indicate when a rule is applicable, and instructions for how to obtain a confidence for the conclusion of the rule from the confidences in its premises. Figure 4 shows an example of an AICronus rule in with its parts labeled. Each of these parts is discussed in more detail below.

Conclusions and Antecedents These two fields define the implication that is the main part of the rule. In a classical first-order logic system, when the antecedents of an implication are true, we can conclude the conclusion

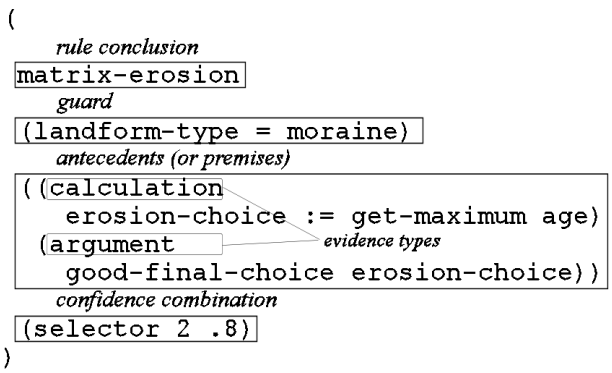


Figure 4: An example AICronus rule with individual parts labeled.

with absolute confidence. In AICronus, when there are arguments supporting the premises in a rule, they can be used, along with the rule, to form an argument for the rule's conclusion. Unlike in a classical system, this argument may eventually be overturned, possibly causing us to conclude the negation of the conclusion.

When combining antecedents with a rule to form an argument, the backwards-chaining engine makes no distinction between evidence for the premises and evidence against them. Antecedents with a negative confidence rating are treated identically to those with a positive confidence, although frequently (as in the example) this case will generate an argument against the rule's conclusion.

The rule in Figure 4 encodes the fact that when matrix erosion is the selected explanation for a spread in sample ages, we will choose the oldest sample age as the correct age for the entire landform. It is important to make sure that this age will be compatible with any restrictions on this landform's age due to known ages of surrounding landforms. In addition, our experts use a heuristic that tends to reject the choice of a single sample's age for the age of the landform when it is distant from the other samples. Arguing for whether the chosen age is a "good-final-choice" addresses both of these issues. Each of the antecedants is labeled with a type to tell the engine how to handle it. These types are discussed in more detail in section 5.3.

Guards The guards on an AICronus rule prevent the system from building arguments using rules that are not applicable to the current case. For example, AICronus has a rule that snow cover is more likely if samples appear younger at higher elevations. However, elevations are recorded for all samples, even when they were collected at essentially the same elevation. Obviously the rule only makes sense when we are dealing with elevation ranges large enough to have different levels of snow cover. Therefore the guard on the rule states that it is only applicable when the elevations of the samples have a large enough range.

In Figure 4, the guard states that the rule is only applicable when the landform under consideration is a moraine. This is because matrix erosion is a process specific to moraines, and it would not make sense for the system to consider it as a possible conclusion on a landform that is not a moraine.

Confidence Combinations We intend to produce a standardized methodology to combine argument confidences into consistently meaningful values. Unfortunately, we currently have an insufficient number of cases to generalize confidence. To allow for rapid feedback and prototyping, confidence combinations are handled somewhat individually until we can determine the correct unified method. We add new combination methods to the system as we add rules for which none of the current methods seem adequate, so the list of combinations types is expanding. Our current methods for confidence combination include:

- Scalar combinations: this method uses a linear combination to combine the confidences in the antecedents into a confidence in the rule's implicant. These combinations are used in rules where all the antecedents are directly related to the conclusion. For example, moraines are formed with a pointed crest which flattens as they erode: A flattened moraine crest is evidence for matrix erosion, and an unflattened crest is evidence against it. Selectors (such as in Figure 4) are a special case of scalar combinations where all weights but one are 0.
- Asymmetric scalars: this combination is like simple scalars, except the linear combination coefficients change based on whether the confidence in the antecedent is positive or negative. These are used in cases where the antecedent is more useful in drawing conclusions one way than another. For instance, we may be interested in whether one sample came from a different landform than the rest. If the samples were collected from the bedrock of the area, we can be very certain that they came from the same landform. However, we cannot be confident of a different origin simply because the samples were not taken from bedrock.
- And-like combinations: if the confidence of every antecedent is positive, a constant confidence is assigned to the implicant of this rule. If any antecedent's confidence is negative, then the confidence is the negative of the constant. For example, if all samples entered have similar ages, we can conclude no process is needed to explain the data. Otherwise, we need to look for some process to explain our observations.
- Combination combinations: some rules use a composition of the other combinations (e.g. a scalar combination, instead of a constant, as the confidence value for an and-like combination). For instance, we can guess that a location is not cold enough for very much snow, at least in recent geological time, if it is both near the equator and at a relatively low elevation. Our confidence in the likelihood that the area does not get cold enough

for significant snowfall goes up as we move closer to the equator and to even lower elevations.

It is apparent from this list of combination mechanisms that most combinations include some scaling factors. Selecting the correct weights for these scaling factors is problematic from both a theoretical and practical standpoint (Doyle 1983). We have addressed this problem initially by using only a small set of actual values for weights corresponding to English phrases used by experts to describe how convincing they find the evidence in each rule (such as "very convincing," "somewhat convincing," "minor evidence," etc.). We have assigned each of these phrases a weight value. Numeric weights are used in the AICronus prototype solely for convenience purposes, and may be changed to symbolic confidence values when we begin to use a more-uniform system of confidence combination. We believe one of the strengths of argumentation is that it allows the system—and its users—to focus on the reasoning behind a conclusion rather than the specific value assigned by the system, so that the somewhat *ad hoc* choice of values is not a major weakness.

5.3. Evidence

The data AICronus uses to draw its conclusions are referred to as evidence. The antecedents in a rule's implication are patterns for evidence—they indicate what evidence will be needed to satisfy the rule. The actual data that causes AICronus to conclude something about the antecedent is the evidence. The system has four different kinds of evidence: observations, simple calculations, complex calculations, and arguments. The distinction between these types of evidence is inspired by the PRET (Stolle and Bradley 1996) system. The separation allows less computationally intensive rules to be considered first.

Observations Observations are direct uses of the data entered by the user. Usually an observation is some binary involving the data, for example checking that all samples have apparent ages less than a certain value. Because the user's observations are generally assumed to be noisy, a piece of observational evidence has more confidence if the relation is stronger. For an antecedent like "elevation < 10000 ft.", we will be more confident that the condition has been met with an elevation value of 7000 ft. than a value of 9999 ft. Observations may also take the form of a quantifier such as for-all or there-exists. These are handled by selecting the highest (for there-exists) or lowest (for for-all) individual confidence value among the quantified entities.

Simple Calculations Simple calculations are generally calculations of simple statistical properties of entered data. They are used for the purpose of generating the calculation's results and all simple calculations have a confidence value of 1. A simple calculation might find the mean of all apparent sample ages so that another part of the rule can check that all apparent sample ages fall within a certain distance of this mean.

Simulations More-complex calculations are called "simulations" because they usually are. Simulations have varying confidence values based on their results. They are implemented as procedures called by the engine examining the rules, allowing them to be as complex as necessary. An example simulation tries different levels of erosion, looking for the rate that best explains the distribution of the apparent sample ages. The simulation returns this erosion rate (which can then be checked to confirm, e.g., that it is reasonable for the climate of the sampling area) and a confidence value indicating how well the returned rate reduces the spread in the calculated ages.

Arguments Sometimes the antecedents of a rule cannot be directly gleaned from the input data. In this case it may be necessary to build a sub-argument for an antecedent and to use the sub-argument as evidence. For example, we know that snow cover is much less likely in areas that are not cold. The system can build a sub-argument for whether the sampling area is cold as part of an overall argument about snow cover.

6. AICronus in Action

Although still in a prototype stage, AICronus is able to produce answers and arguments similar to those produced by experts. Here is an example set of input data created for the purpose of communicating with experts about their reasoning about specific examples.

Landform Type: Moraine
Flat Crest

Sample	Age (yrs)	Error (yrs)	Chemistry
1	95000	5000	A
2	100000	6000	A
3	105000	4000	A
4	110000	4500	A
5	115000	5500	A

Two experts, shown this set of input data, concluded that the process affecting the data was almost certainly matrix erosion, primarily because of the distribution of apparent ages but also because the landform is a relatively old moraine with a flat crest. AICronus considered inheritance, but rejected it because the errors were too small and because all the sample appeared to have the same origin, making different inheritance levels for different samples unlikely.

Here is AICronus's actual output, given this input set:

```
argument for conclusion matrix erosion:
total confidence: 0.87
  evidence for erosion:
    age is approximately linear
    landform is relatively old (>100000 yrs)
    visual erosion observed
     argument for conclusion visual-erosion:
```

```
      total confidence: 0.6
        evidence for visual-erosion:
          flat crest
        evidence against visual-erosion:
          (none)
      consistent with other landforms in area
       argument for conclusion consistent-age:
       total confidence: 1
        evidence for consistent-age:
          no other landforms known
        evidence against consistent-age:
          (none)
      landform is a moraine
    evidence against erosion:
       (none)

  argument for conclusion inheritance:
  total confidence: 0.37
    evidence for inheritance:
      consistent with other landforms in area
       argument for conclusion consistent-age:
       total confidence: 1
       evidence for consistent-age:
         no other landforms known
       evidence against consistent-age:
         (none)
      removal of older samples allows "no
process"
      landform is a moraine
      does not violate max theoretical inher-
itance
    evidence against inheritance:
      small error values (<10000 yrs)
      samples have same origin
       argument for conclusion same-origin
       total confidence: 0.5
       evidence for same-origin:
         all samples have same chemistry
         not taken from bedrock
       evidence against same-origin:
         landform is a moraine
```

Because AICronus rules are already built around the idea of arguments, producing an English argument from a set of rules is relatively simple. The system simply needs to print a representation of all the argument trees for all the conclusions it considered. Since the trees already contain both the rules and the evidence that allowed the system to use each rule, simply translating each rule and its evidence into English produces a complete and coherent argument.

Although both inheritance and erosion have positive confidence values, the system's confidence in erosion is much higher. This exactly matches the judgement of the experts who were shown these data. Moreover, AICronus's arguments about the possible processes closely match the arguments given by the experts in each case. Despite the difficulties inherent in the field of cosmogenic isotope dating, AICronus already shows significant promise in understanding and automating the reasoning used by experts.

7. Future Work

AICronus is a work in progress. We plan a number of improvements over the next several years. The most critical of these improvements is expanding the system's knowledge base. We are in the process of using this prototype version to solicit feedback and new knowledge from experts. In addition, we are working on integrating this system with the iCronus project (Anderson and Bradley 2006) so input data need not be entered by hand and output arguments can be presented visually rather than via the current command-line interface. Our current interface requires the user to enter a complete set of samples as Scheme code—an obvious drawback for many users. We expect the system to go into regular use by geologists once these improvements are complete.

We are considering other improvements to make the system more user-friendly. These include removing the numeric confidence values in the output and presenting arguments in a more natural prose form. Currently the system does not provide any assistance for going back to collect more samples to distinguish between processes that appear to have equally good arguments. We are considering an approach similar to (McIlraith and Reiter 1992) for implementing this functionality.

Other future projects include allowing the user to engage in an argument with the system to update the knowledge base over time based on user input. We also hope to apply this framework to problems in other fields. As discussed above, the major strength of the argumentation framework is that it copes well with uncertain and contradictory information. These features often appear in new or rapidly changing scientific fields. Although we expect that the rule and argument structures of AICronus will translate well between fields, areas with more conclusions might require a more performance-aware engine. Other fields might also require us to consider new types of evidence.

8. Related Work

Many diagnostic systems solve problems similar to the one solved by AICronus, in which there is some normal, expected behavior (in isotope dating, all samples of the same apparent age) and the causes of divergences from this behavior (e.g. a geologic process) must be diagnosed. However, the predominant paradigm in medical diagnosis is to build a complete model of a system and to use that model to make predictions about malfunctions (Lucas 1997), (Struss 2004). This methodology is not suited to our particular domain because complete models of most geologic processes simply do not exist. In addition, model-based systems are not as suited to handling contradiction.

Diagnosis systems that handle contradiction do exist, for example (Doyle 1983), (Santos 1991), (Cem Say 1999) and

(Gaines 1996). However, all of these systems use "absolute" rules. It is not possible to express the idea that some data may only partially support a conclusion. Instead, the conditions under which the rule does not provide support are explicitly encoded in the system. AICronus needs to include rules for partial support of conclusions in order to accurately reflect the reasoning process used by experts. For instance, experts are more likely to accept (and require more evidence to reject) an "inheritance" conclusion for a moraine than for other landform types. Trying to model this behavior without an ability to express partial support would be extremely difficult. On the other hand, all of these systems are capable of presenting their reasoning to the user to help convince experts of initially rejected conclusions, an important feature of AICronus. (Santos 1991) is also capable of presenting alternative conclusions to the user so that if the user does not agree with a particular conclusion the tool is likely to still be useful (another AICronus feature).

Several authors have discussed the virtues of presenting the reasoning behind a system's conclusions in the form of trees or arguments, including (Boy and Gruber 1990) (Bouwer and Bredeweg 2002) and (Gaines 1996). (Puyol-Gruart, Godo, and Sierra 1992) points out that even when a particular conclusion cannot be reached by a reasoning system, it is likely that presenting what the system *has* managed to determine will be useful to the user. AICronus handles this situation by reporting its complete arguments, even in cases where the absolute values of the confidences are quite small.

Case-based reasoning (Kolodner 1993), (Cunningham, Bonzano, and Smyth 1995), and (Clark 1989) presents a way to sidestep the issues of partial support and contradiction by presenting intact the reasoning of experts on previous cases that are similar to the current problem instance. Unfortunately, case-based reasoning is unsuitable to AICronus because the field of cosmogenic isotope dating is still very new and relatively small. As a result, there are too few already-analyzed cases to cover all of the possible variables in selecting a responsible process. (Surma and Vanhoof 1995) offers a solution to this objection by using rules for "normal" cases and case-based reasoning for cases that are exceptional in some way. Unfortunately, the problem being solved by AICronus has so many variables to address that it is difficult to classify any case as "normal."

(Turner 1992) uses schemas—abstracted cases—to perform diagnosis by considering particular symptoms. When a symptom is unique to a particular type of disease, the system considers diagnosing that disease. If the symptoms expected for that disease are observed, then it is considered a correct diagnosis. This architecture is not suitable for isotope dating because it fails to handle contradiction well. In addition, schemas are difficult to extract because it is difficult to determine what is typical for any process.

Several kinds of defeasible reasoning besides argumentation have been put forth by various authors. These include circumscription (McCarthy 1980), (McCarthy 1986), default reasoning (Reiter 1980), (Doyle 1983), and other forms of nonmonotonic reasoning (Pereira, Alferes, and Apar'icio 1991), (Gaines 1996). Circumscription allows the definition of normal situations and the cases that can circumscribe them. It requires the definition of specific aspects that are abnormal only in abnormal situations, so that it is necessary to create a large number of "aspect" variables to express all of the possible abnormal situations. Default reasoning uses rules with default conclusions and then defines specific exceptions where they do not apply. This is similar to the "guards" on AICronus rules which prevent them being used to build arguments in some situations. The nonmonotonic logic defined in (Pereira, Alferes, and Apar'icio 1991) assigns a likelihood to various rules so that they can normally, sometimes, or exceptionally apply. Rules have conditions stating specifically when they do apply. (Gaines 1996) uses a tree structure for rules with default conclusions at the root and repeated refinements or rejections of the initial conclusion(s) as the tree is descended.

While all of these logics are excellent choices for solving many different problems, they all require some explicit definition of when particular rules are defeated. The heuristics used by our experts are insufficiently complete for these explicit definitions. Also, all of these nonmonotonic logics use defeat of specific rules rather than attacking conclusions. AICronus rules are not bound in a strict fashion to conclusions; a rule may be in support of a conclusion (but turn out to be unimportant in light of other rules or conclusions) or against one (but be negated by the presence of higher-confidence results elsewhere). (Etherington, Kraus, and Perlis 1991) describes other problems with various nonmonotonic logics.

There is a large body of work on different kinds of argumentation systems. Most of this work grapples with the question of when it is appropriate to declare a particular argument defeated, with different authors reaching various conclusions. Most authors (Dung 1995), (Pollock 1994), (Vreeswijk 1991), (Farley 1997), and (Prakken 1996) consider only absolute defeat of arguments. Little work on partial support and defeat has been done, although the Logic of Argumentation introduced by Krause et. al. (Krause, Ambler, Elvang-Gøransson, and Fox 1995), on which the AICronus framework is based, does partially address these issues.

Few results exist for applying argumentation to specific problems. Most practical systems are aimed at communication-based applications, especially communication between agents (Parsons, Sierra, and Jennings 1998). The idea of argumentation as a form of communication has also been explored by (Prakken 1996), (Farley 1997) and (Vreeswijk 1993), who cast the construction of arguments as a form of dialectics. In these systems, two agents repeatedly try to form arguments for a given conclusion, and then defeat those arguments. (Prakken 1996) allows defeat to take the form of defeating particular rules, rather than only the more-traditional undercutting and rebuttal.

This defeat is analogous to the attachment of confidence values to specific AICronus rules; rules with greater confidence can defeat rules with smaller absolute confidences. (Farley 1997) allows the user to globally alter the relative strength of arguments. Three modes are allowed, where a conclusion is made if some argument for it exists, a conclusion is accepted if there are more arguments for it than against it, and a strict mode where a conclusion is believed only if there is an argument for it and all arguments against it are defeated. The second mode in particular is similar to the mechanism used by AICronus, except that the strengths of the arguments in (Farley 1997) are not determined by which rules are used to form them—all defeasible rules have the same believability.

9. Conclusion

Although still in its prototype stage, AICronus is a promising model for the process of cosmogenic isotope dating. Using a logic of argumentation, we have generated preliminary results that closely parallel the reasoning and explanations of experts in the field. We expect that once the knowledge base for the system is complete AICronus will be able to reach insightful conclusions more quickly and consistently than experts under certain circumstances. In particular we expect this benefit in cases where superficially contradictory evidence disguises an extremely typical manifestation of some process.

We expect that AICronus will be a significant advancement for the field of cosmogenic isotope dating. Among other things, creating AICronus forces experts to make explicit many implicit rules and assumptions, allowing the easier identification of faulty or missing assumptions. These assumptions can then be used consistently between different experts. New knowledge about cosmogenic isotope dating can be written in an unambiguous form, allowing easier communication of new knowledge. Finally, we hope that AICronus will increase the speed of discovery of new knowledge about the ancient Earth.

10. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. ATM-0325812.

References

Anderson, K. and Bradley, E. 2006. iCronus information is available at: http://serl.cs.colorado.edu/projects/icronus/

Bouwer, A. and Bredeweg, B. 2002. Aggregation of Qualitative Simulations for Explanation. In Proceedings of the Sixteenth International Workshop on Qualitative Reasoning. Barcelona.

Boy G. and Gruber, T.R. 1990. Intelligent Assistant Systems: Support for Integrated Human-Machine Systems. In Proceedings of 1990 AAAI Spring Symposium on Knowledge Based Human-Computer Communication. Stanford University.

Bradwell, T. 2001. A New Lichenometric Dating Curve For Southeast Iceland. *Geografiska Annaler: Series A, Physical Geography* 83(3): 91-101.

Cem Say, A.C. 1999. Using Inter-Behavior Contradictions for Modeling and Diagnosis. In Proceedings of the Thirteenth International Workshop on Qualitative Reasoning. Loch Awe, Scotland.

Clark, P.E. 1989. Exemplar-based reasoning in geological prospect appraisal, Technical Report, 89-034, Turing Institute, Univ. of Stratchclyde.

Cunningham, P., Bonzano, A., and Smyth, B. 1995. An incremental case retrieval mechanism for diagnosis. Technical Report TCD-CS-95-01, Department of Computer Science, Trinity College.

Desilets, D. and Zreda, M. 2006. Elevation dependence of cosmogenic 36Cl production in Hawaiian lava flows. *Earth and Planetary Science Letters* 246: 277-287.

Doyle, J. 1983. Methodological Simplicity in Expert System Construction: The Case of Judgments and Reasoned Assumptions. *AI Magazine* 4(2): 39-43.

Dung, P.M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence* 77(2): 321-357.

Etherington, D., Kraus, S., and Perlis, D. 1991. Nonmonotonicity and the Scope of Reasoning. *Artificial Intelligence* 52(3): 221-261.

Farley, A.M. 1997. Qualitative Argumentation. In Proceedings of the Eleventh International Workshop on Qualitative Reasoning. Cortona, Italy.

Gaines, B.R. 1996. Transforming Rules and Trees into Comprehensible Knowledge Structures. In *Advances in Knowledge Discovery and Data Mining*, 205-226. Cambridge, MA: MIT Press.

Gosse, J. C. and Phillips, F. M. 2001. Terrestrial cosmogenic nuclides: theory and application. *Quaternary Science Reviews* 20(14): 1475-1560.

Kolodner, J. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Krause, P., Ambler, S., Elvang-Gøransson, M., and Fox, J. 1995. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence* 11: 113-131.

Lal, D. 1958. Investigation of Nuclear Interactions Produced by Cosmic Rays. PhD diss., Univ. Bombay.

Lucas, P.J.F. 1997. Symbolic diagnosis and its formalisation. *The Knowledge Engineering Review* 12(2): 109-146.

McCarthy, J. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence* 13(1,2): 27-39, 171-172.

McCarthy, J. 1986. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence* 26(3): 89-116.

McIlraith, S. and Reiter, R. 1992. On tests for hypothetical reasoning. In *Readings in Model-Based Diagnosis*, 89-96. San Mateo, CA: Morgan Kaufman.

Noller, J. S., Sowers, J. M. and Lettis, W.R., Eds. 2000. Quaternary Geochronology: Methods and Applications. AGU Reference Shelf. Washington, American Geophysical Union.

Parsons, S., Sierra, C., and Jennings, N. 1998. Multi-context Argumentative Agents. In Proceedings of the Fourth International Symposium on Logical Formalizations of Commonsense Reasoning. London.

Pereira, L.M., Alferes, J.J., and Apar'icio, J.N. 1991. Non-monotonic reasoning with well founded semantics. In Proceedings of the Eighth International Logic Programming Conference, 475-489. Paris: MIT Press.

Pollock, J.L., 1994. Justification and defeat. *Artificial Intelligence* 67(2): 377-407.

Prakken, H. 1996. Dialectical proof theory for defeasible argumentation with defeasible priorities. In Proceedings of the Biannual International Conference on Formal and Applied Practical Reasoning Workshop, 202-215. Bonn.

Puyol-Gruart, J., Godo, L., and Sierra, C. 1992. A Specialization Calculus to Improve Expert Systems Communication. In Proceedings of the European Conference on Artificial Intelligence, 144-148.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13: 81-132.

Santos, E. 1991. On the Generation of Alternative Explanations with Implications for Belief Revision. In Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence, 339-347. San Mateo, CA: Morgan Kaufmann.

Shanahan, T.M. and Zreda, M. 2000. Chronology of Quaternary glaciations on Mount Kenya and Kilimanjaro. *Earth and Planetary Science Letters* 177: 23-42.

Stolle, R. and Bradley, E. 1996. A Customized Logic Paradigm for Reasoning about Models. In Proceedings of the Tenth International Workshop on Qualitative Reasoning. Stanford Sierra Camp, CA.

Struss, P. 2004. Deviation Models Revisited. In Proceedings of the Eighteenth International Workshop on Qualitative Reasoning. Evanston, IL.

Surma, J., and Vanhoof, K. 1995. Integrating rules and cases for the classification task. In Proceedings of the First International Conference on Case-Based Reasoning, 325-334. Berlin: Springer Verlag.

Turner, R.M. 1992. A view of diagnostic reasoning as a memory-directed task. In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society. Bloomington, IN: Cognitive Science Society.

Vreeswijk, G. 1991. The feasibility of defeat in defeasible reasoning. In Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, 478-483. Morgan Kaufmann.

Vreeswijk, G. 1993. Defeasible dialectics: a controversy-oriented approach towards defeasible argumentation. *Journal of Logic and Computation* 3: 317-334.

Learning Financial Rating Tendencies with Qualitative Trees

Llorenç Roselló[1], Mónica Sánchez[1], Núria Agell[2], Francesc Prats[1]

[1]GREC Group and Ma2-Universitat Politècnica de Catalunya
[2]GREC Group and ESADE-Universitat Ramon Llull

Abstract

Learning financial rating tendencies requires knowledge of the ratios and values that indicate a firm's situation as well as a deep understanding of the relationships between them and the main factors that can modify these values. In this work, the Qualitative Trees provided by the algorithm QUIN are used to model financial rating and to learn its tendencies. Some examples are given to show the system's predictive capabilities. The rating tendencies and the variables that most influence those tendencies are analyzed.

1. Introduction

In this paper, a learning process to induce a qualitative model providing a causal interpretation between the variation of some input variables and the tendency of the output variable is described. To obtain the model, the algorithm QUIN (QUalitative INduction) is used [Šuc and Bratko, 2001, 2003, 2004]. QUIN addresses the problem of the automatic construction of qualitative models across an inductive learning of numerical examples by means of Qualitative Trees. These trees have qualitative functional restrictions inspired in the and-predicates introduced by Forbus [Forbus, 1984] in their writings. A qualitative tree defines a partition in the attributes space in zones with a common behavior of the chosen variable. The algorithm was designed and implemented by Dorian Šuc and Ivan Bratko. This qualitative model is especially suitable for analysing financial rating tendencies because it allows one to analyze how the variables describing the state of a firm at a given moment can modify its valuation rating.

Big data sets containing patterns or examples with many attributes are unmanageable with QUIN because of its algorithmic complexity. Considering that this is a characteristic of the case of study, it has been necessary to reduce the number of variables and to group the sets of data in order to simplify the available data set. Data have been provided by Thomson Financial and Standard & Poor's, and correspond to 1177 firms represented by 46 input variables together with their financial rating given by Standard & Poor's for 2003. The input variables are ratios that try to capture aspects of liquidity, profitability, financial structure, size and turnover or level of activity of the company.

The QUIN algorithm was applied using data for firms operating in Canada, Japan, a group containing European firms, and random samples of firms in the USA.

The structure of the paper is as follows: in Section 2 outlines the preprocessing of data through factorial analysis. Section 3 gives general descriptions of qualitative trees and the QUIN algorithm. Section 4 explains the general approach to financial ratings, the experiments undertaken, and the results obtained. The concluding section sets out findings and suggests new ways for solving the problems presented.

2. Data preprocessing

In the words of QUIN's authors "QUIN cannot efficiently handle large learning sets, neither in terms of examples nor the attributes" [Zabkar et al, 2005] due to its complexity. When either the number of patterns or the number of attributes is too large for the algorithm, data pre-processing is needed.

In this work, the set of patterns has been partitioned following the categories of a nominal qualitative variable. The country where the firm has its headquarters has been used to partition the set of 1177 worldwide firms. In addition, the number of input variables considered in the learning process has been reduced by using factorial analysis. SPSS software has been used to extract principal components for the whole set of 46 variables, turning out 5 principal components to explain 60% of the total variability. QUIN has been applied using data of these 5 principal components corresponding to companies of Canada, Japan, a group containing European firms, and some random samples of firms operating in USA.

3. Qualitative Trees and QUIN

In this section the concept of qualitative tree is outlined as a previous concept to introduce the QUIN algorithm used for the case of study.

Given a set of *N* patterns, each pattern described by *n+1* variables where $X_1,\ldots,X_n$, are the *attributes* and X_{n+1} is the *class*, the goal is learning zones of the space that should present a common behavior of the class variable. These zones are described by means *of qualitative trees*. A qualitative tree is a binary tree with internal nodes called *splits*; its leaves are *qualitatively constrained functions*. From now on these functions will be denoted by QCF. The internal nodes define partitions of the space of attributes. In each node there is an attribute and a value of this attribute. The QCF define qualitative constraints of the class variable in the following way: if $F: R^n \rightarrow R$ is a map that associates to each *n* attributes a value of the class variable, a QCF associated to the function *F* and to a *m*-tuple $(x_1, \ldots, x_m) \in R^m$, with $m \leq n$, is denoted by $F^{s1,\ldots,sm}(x_1,\ldots,x_m)$, where $s_i \in \{+,-\}$, and it means that:

$$\left.\frac{\Delta F}{\Delta x_i}\right|_{\Delta x_j = 0,\, j \neq i} = s_i$$

In other words, $s_i = +$ means that *F* is a strictly increasing function respect to the variable x_i, and it is strictly decreasing when $s_i = -$. Figures 1 and 2 give a simple example of qualitative tree; figure 1 shows the graphical representation of the function $F(x,y) = \frac{x}{y^2+1}$

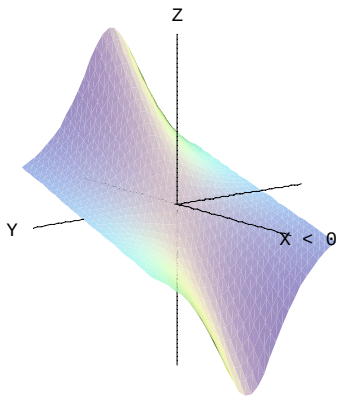


Figure 1. The plot of $F(x,y) = \frac{x}{y^2+1}$

And figure 2 shows the induced qualitative tree of this function.

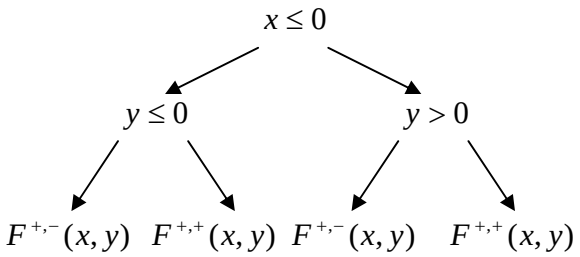


Figure 2. The qualitative tree of $F(x,y) = \frac{x}{y^2+1}$

One should note that in general the explicit expression of the function *F* is unknown.
In order to decide which QCF is better adjusted to a given set of patterns, the qualitative changes q_i of variables x_i are used, where $q_i \in \{pos, neg, zero\}$, in such a way that if $\Delta x_i > 0$ then $q_i = pos$ and if $\Delta x_i = 0$ then $q_i = zero$.
Let us define de QCF-prediction $P(s_i, q_i)$, $(s_i \in \{+,-\})$ as

$$P(s_i,q_i) = \begin{cases} \text{pos}, & \text{if } (s_i = + \wedge q_i = \text{pos}) \vee (s_i = - \wedge q_i = \text{neg}) \\ \text{neg}, & \text{if } (s_i = + \wedge q_i = \text{neg}) \vee (s_i = - \wedge q_i = \text{pos}) \\ \text{zero}, & \text{otherwise} \end{cases}$$

Then, for any pair of patterns (e,f) a *qualitative change vector* is formed, being each component of this vector $q_{(e,f),i}$ defined by:

$$q_{(e,f),i} = \begin{cases} \text{pos}, & \text{if } x_{f,i} > x_{e.i} + \varepsilon \\ \text{neg}, & \text{if } x_{f,i} > x_{e.i} - \varepsilon \\ \text{zero}, & \text{otherwise} \end{cases}$$

Where $x_{f,i}$ is the *i*-th component of *f*. The parameter ε is introduced to solve the cases with tiny variations: 1% of the difference between maximal and minimal value of the *i*-th attribute. Once these concepts have been introduced, the method to choose the QCF that better describes data will be explained. What constitutes `better' in this context will be discussed later.

A QCF, $F^{s_1,\ldots,s_n}$, that describes the behavior of the class X_{n+1} is *consistent* with a vector of qualitative changes if all QCF-predictions $P(s_i, q_{n+1})$ are non negative with at least one positive. In other words, it is consistent when the vector of qualitative changes does not contradict the QCF. If there are simultaneously positive and negative QCF-predictions or when all the predictions are zero, then *there*

is an ambiguity in the prediction of the QCF with the vector of qualitative changes. Finally, a QCF is *inconsistent* with a vector of qualitative changes if it is neither consistent nor ambiguous.

For each QCF an error-cost is defined from the number of consistent vectors of qualitative changes and the number of ambiguous vectors of qualitative changes (this has to be verified for all possible qualitative change vectors for the problem under consideration). This error-cost gives a measurement of the suitability of the QCF function to describe data.

The QUIN algorithm constructs the qualitative tree with a greedy algorithm that goes from top to bottom similar to the ID3 [Quinlan, 1986]. Given a set of patterns, QUIN computes the error-cost for each one of the QCF found for each partition, and chooses the partition that minimizes the error-cost of the tree. The error-cost of a leaf is the error-cost of the QCF that there is in this leaf. The error-cost of a node is the error-cost of each one of the sub-trees plus the cost of the division.

4. Case of Study: Financial Rating

The case below falls within the development frame of the AURA research project, which sets out to adapt soft-computing techniques to the study of the financial rating tendencies by using qualitative reasoning.

The main goal of the project is to use these techniques to extract knowledge and allow prognosis. In particular, in this paper, a qualitative system based on QUIN is considered to represent the factors that are relevant in computing credit risk. Using factorial analysis, five principal components have been extracted and used to study tendencies of the level of risk. QUIN has been applied to several sets of firms (characterized for these five components and their Standard & Poor's rating) corresponding to different countries.

4.1. Financial rating

The rating is an attempt to measure the financial risk of a given company's bond issues. The specialized rating agencies, such as Standard & Poor's, classify firms according to their level of risk, using both quantitative and qualitative information to assign ratings to issues. Learning the tendency of the rating of a firm therefore requires the knowledge of the ratios and values that indicate the firms' situation and, also, a deep understanding of the relationships between them and the main factors that can modify these values.

The processes employed by these agencies are highly complex and are not based on purely numeric models. Experts use the information given by the financial data, as well as some qualitative variables, such as the industry and the country or countries where the firm operates, and, at the same time, they forecast the possibilities of the firm's growth, and its competitive position. Finally, they use an abstract global evaluation based on their own expertise to determine the rating. Standard & Poor's ratings are labeled AAA, AA, A, BBB, BB, B, CCC, CC, C and D. From left to right these rankings go from high to low credit quality, i.e., the high to low capacity of the firm to return debt.

4.2. Learning Financial Rating Tendencies

The problem of classifying firms by using their descriptive variables has already been tackled by several authors [Ammer, J.M. and Clinton, N., 2004]. The goal of this paper is to analyze the variables that influence variations in ratings and how this influence is expressed. Data are the financial results presented by 1177 companies worldwide and the rating that Standard & Poor's granted in reference to year 2003. Each firm is considered as a pattern, described by 46 input variables, and the variable class is the rating. The QUIN algorithm is used to learn which the qualitative tree associated to this problem is.

4.3. Experimental Results

The experiment began with preprocessing of the data. The algorithmic complexity of the QUIN, especially when the number of patterns, as well as the number of attributes, is considerable, making it advisable to start with the following two steps:

1. To limit the number of patterns, by grouping the companies, in particular: Canada (83 patterns), Japan (26 patterns), a group containing all European firms (129 patterns), and some random samples of firms operating in USA (between 60 and 80 patterns).
2. To reduce the number of variables treated, by using factorial analysis extracting principal components.

Several tests have been carried out for firms in the above selected groups. The 46 input variables are grouped into five groups, each group describing a certain financial characteristic. Using SPSS software for the whole set of 46 variables, turned up 5 sufficient principal components to explain 60% of the whole set of patterns variability (and thus learn the financial rating tendency) - the corresponding results are commented upon below. It has to be pointed out that, in addition, experiments with more principal components were carried out; specifically, with 7 principal components, explaining 63% of variability, with 9, explaining 75% of variability, and with 13 principal components (98%). It has been seen that if the number of components increases, then the qualitative trees corresponding to Europe and to the American groups become very complex, making the observation of behavioral patterns difficult. By contrast, the qualitative tree corresponding to Japan with $n \geq 7$ principal components becomes more simplified than in the case $n < 7$, and invariant from 7 on.

The obtained results show certain common trends in the rating tendencies in the European and American groups of firms, whereas Japanese firms exhibit different behavior.

When applying QUIN using data in the case of 5 principal components corresponding to the sets of data of the mentioned groups, the obtained qualitative trees show the rating tendencies and reveal the most relevant variables. The five principal components are named F_1, …, F_5. Three of them, in particular F_1, F_4 and F_5 are related to liquidity, F_2 is related to financial structure and F_3 is related to profitability. The considered order for the rating has been the order given by Standard & Poor's, but from less to more risk, i.e., $AAA \prec AA \prec \ldots \prec D$. The results are represented in the following figures.

$$R^{-,+,-,-,+}(F_3, F_5, F_4, F_1, F_2)$$

Figure 3: Canadian firms' induced qualitative tree

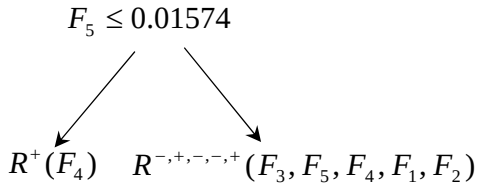


Figure 4: European firms' induced qualitative tree

$$R^{-,+,-,-}(F_3, F_5, F_4, F_1)$$

Figure 5: Random samples 1, 3 and 4 of USA firms' induced qualitative tree

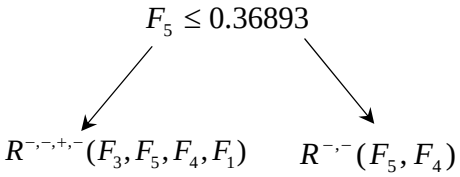


Figure 6: Random sample 2 of USA firms' induced qualitative tree

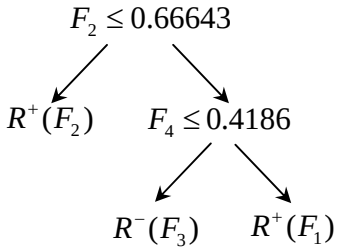


Figure 8: Japanese firms' induced qualitative tree

The induced qualitative tree s obtained for the different countries, though not identical, show certain common characteristics that provide useful information about the problem. With these five principal components, the rating dependency is the same for Canada, Europe and the random samples of USA. Mostly, trees with only one leave are obtained. Within the set of variables corresponding to liquidity, rating always increases with respect to F_1 and F_4, and decreases with respect to F_5. Rating always increases with respect to the profitability-related component. The component related to financial structure appears in few trees, and, at this level, it can be concluded that this component does not give much qualitative information.

One should note that in general trees obtained from different data sets, even though deduced from the *same function*, are not the same. For instance, in the case considered in figures 1 and 2, the qualitative tree corresponding to $(x,y) \in [-1,1] \times [-1,1]$ is totally different to the qualitative tree corresponding to the domain $(x,y) \in [3,7] \times [3,7]$. In addition, when the explicit expression of the function F is unknown, the complexity of the problem increases.

In the presented case of study, one possible explanation of the difference between trees is that each tree has been constructed over a different domain. The examination of the numerical data shows that, for example, the range of the third factor in Japan is approximately [-0.066, 0.095], whereas in Canada, as well as in the first, third and fourth random sets of USA firms, the range is approximately [-0.3, 0.3].

Therefore, it is perfectly natural that firms from very different countries present different features, whereas firms operating in countries under similar economical conditions (as it can be the case of Canada and USA) show similar features. On the other hand, the more factors are considered the more differences are able to appear. To sum up, in this case of study, induced qualitative tree s of different areas are being compared, and, in these different areas the rating behavior must not be neither the same nor very similar.

Conclusion and Future Work

This paper presents on-going work, which provides new strategies for credit risk prediction. The choice and definition of the variables involved, as well as study of the influence of each variable on the final result, have been analyzed.

The induced qualitative tree s provided by QUIN lead to a useful model for learning rating tendencies and studying to what extent ratings depend on several variables representing different financial features. When using the five principal components, the qualitative trees provided by QUIN algorithm for different sets of European and American firms show internal common trends.

In the case studied, the QUIN algorithm was used for an output qualitative variable described on an ordinal scale. In

general, due to the non deterministic intrinsic nature of the problem, the expected results are a probability function for the rating corresponding to different values of the input variables. However, use of the QUIN algorithm provides qualitative information about the monotonic behavior of rating with respect to financial features.
Future work will cover the speed of rating tendencies (i.e., how "fast" or "slow" ratings change) by using orders of magnitude descriptions.
The particular evolution of the rating of a given firm and its prediction from the previous rating and the values of its present financial ratios is currently being studied.

Acknowledgement

This work has been partially financed by MEC (Ministerio de Educación y Ciencia): AURA project (TIN2005-08873-C02-01 and TIN2005-08873-C02-02)

References

Ammer, J.M. and Clinton, N: *Good news is no news? The impact of credit rating changes on the pricing of asset-backed securities.* International Finance Discussion Paper, no 809, Federal Reserve Board, (2004) July

Forbus, K: *Qualitative Process Theory*, Artificial Intelligence (1984) 24:85-168

Quinlan, J.R. *Induction of decision trees.* Machine Learning, 1, pp. 81-106, 1986.

Šuc, D., Bratko, I. *Induction of Qualitative Trees.* In L. De Raedt and P. Flach editors, Proc. 12th European Conference on Machine Learning, pages 442-453. Springer, 2001. Freiburg, Germany.

Šuc, D., Vladusic, D., Bratko, I.: *Qualitative Faithful Quantitative Prediction,* Proc. ŠŠŠ18th In. Joint Conf. on Artificial Intelligence, IJCAI.2003, pp. 1052-1057

Šuc, D., Vladusic, D., Bratko, I.: *Qualitative Faithful Quantitative Prediction.* Artificial Intelligence 158 (2004) 189–214

Zabkar, J., Vladusic, D., Zabkar, R., Cemas, D., Šuc, D., Bratko, I.: *Using Qualitative Constraints in Ozone Prediction,* Proc. 19th In. Workshop on Qualitative Reasoning QR05, pp. 149-156.

Modelling sustainability in the Riacho Fundo water basin (Brasília, Brazil)

Paulo Salles[1], Bert Bredeweg[2], Ana Luiza Rios Caldas[1], Tim Nuttle[3]

[1]Institute of Biological Sciences, University of Brasília, Brasília, Brazil;
psalles@unb.br; analoo@gmail.com

[2]Human Computer Studies Laboratory, University of Amsterdam, Amsterdam, The Netherlands;
bredeweg@science.uva.nl

[3]Institute of Ecology, Friedrich Schiller University, Jena, Germany;
and Department of Biological Sciences, University of Pittsburgh, USA;
nuttle@pitt.edu

Abstract

This paper explores the use of assumptions to build multiple models about sustainable development in a compositional way. It presents a model aiming at supporting stakeholders to improve their understanding about a water basin system under the pressure of changes in land use. Domain knowledge is approached from three perspectives: urban, semi-urban and rural. Simulations explore sustainability issues related to (a) the effects of urban drainage systems; (b) the dynamics of erosion and water infiltration in the soil; and (c) the effects of vegetation cover on soil and water conditions and agricultural production. The paper discusses the use of modelling primitives to define and implement perspectives based only in conceptual knowledge to approach ill-defined domains as sustainability.

1. Introduction

This paper describes the use of assumptions (Falkenhainer and Forbus, 1991) to define perspectives for organizing knowledge in a qualitative model about sustainability of the Riacho Fundo basin (Brasília, Brazil). The model was designed to support stakeholders and decision makers to improve their understanding about the complex problems they have to deal with, and is being developed in one of NaturNet–Redime project (www.naturnet.org) case studies. The work described here aims at answering the following research questions: dealing with a large body of domain knowledge, (a) how to organize knowledge encoded in a library of model fragments in order to create sets of simulation models each addressing a class of sustainability issues in the Riacho Fundo basin? and (b) how to optimize the use of Garp3's representational apparatus and algorithm (Bredeweg *et al*. 2006) to create multiple models as mentioned in (a)?

The use of modelling assumptions to define perspectives for reasoning with multiple models about physical systems is a long standing problem in Qualitative Reasoning. The goal of this paper is to discuss problems and solutions of perspective-taking using only conceptual knowledge about sustainability – which perspectives to take, how to represent them and how to explore modelling primitives to implement assumptions. The paper is organized as follows: in section 2, we briefly discuss relevant aspects of sustainability in the Riacho Fundo basin. Next, in section 3, fundamental aspects of QR theory and some details of Garp3 are presented. The implementation of three perspectives taken to sustainability is described in section 4, and in section 5 selected simulations are used to illustrate the model results. The implementation of assumptions is discussed in section 6. Finally, the paper ends with references to ongoing work and conclusions.

2. Riacho Fundo: transition from a natural to an urban environment

The Riacho Fundo is a small basin (225,48 km^2) in Brasília, central Brazil. Since the new capital, Brasília, was built in the late 1950's, it has been the most impacted area of the Paranoa Lake water basin. Most of the impacts are related to changes in land use, that transformed natural areas into rural and urban areas, currently with ca. 200.000 inhabitants. Due to the urbanization process, springs, streams and natural vegetation are disappearing, and biodiversity is being reduced. Changes in habitat put high pressure on many species, including the Riacho Fundo's largest mammal, the capybara (*Hydrochoerus hycrochaeris*).

Most of the economic activities in the area are related to services, including business offices, commerce and automotive service garages. Garages are often responsible for soil and water contamination with petroleum-based products. Despite their small scale in the Riacho Fundo, industrial activities and agriculture have significant impacts. Most of industries is related to food and clothes production and contributes effluents rich in organic matter and chemical pollutants into rivers and streams. Agriculture is based on corn and vegetables production, resulting in soil and water contamination with pesticides. Cattle, pork, and chicken are the most important livestock in the basin. Runoff of animal waste may cause eutrophication of water bodies. Details about the the Riacho Fundo basin can be found in Salles and Caldas (2006).

According to the stakeholders the most relevant problems in the basin are: (1) uncontrolled land occupation; (2) deforestation and destruction of natural habitats; (3) problems with basic sanitation (including garbage and sewage deposition in open land and water bodies) and lack of adequate rain-water drainage system; (4) unsustainable practices by farmers and by the industrial sector; and (5) deficit in community participation, in part due to lack of knowledge about local degradation processes and environmental concern (Salles, 2001).

3. Assumptions and perspectives in qualititative models

Given the large amount of knowledge involved in the discussion about sustainability it is necessary to organize such knowledge in order to create meaningful sets of simulation models to support stakeholders. The solution proposed here is to build up *perspectives* using modelling assumptions.

3.1 Perspectives

Given a large qualitative model about sustainability, a perspective defines a subset of simulation models that can be created to achieve a particular goal, that is, to answer questions of a particular type. Creating a perspective requires the selection of a sub-system within the larger system of interest, which includes a sub-set of entities and potentially a sub-set of the entities' features (quantities).

Perspectives serve an organizational function that, once the properties of a perspective are defined, guides the modeller in selecting appropriate assumptions, structural relations and scenarios. Perspectives are thus useful not only in defining and constraining a simulation, but also to automate the search of model fragments in a library, taking into consideration certain aspects of the encoded knowledge while ignoring the rest. Depending on which perspective is adopted, different entities, quantities, values, and causal relationships are included in the simulation. For the Riacho Fundo model, three perspectives were defined – *Rural*, *Semi-urban*, and *Urban* – each focusing on particular combinations of environmental, economic and social phenomena.

Perspectives can be implemented using explicitly represented assumptions and other modeling primitives, including hierarchies of entities and model fragments, attributes, alternative quantity spaces for key quantities and alternative representations of key concepts. The consequences of adopting a given perspective in a simulation are determined automatically by the reasoning engine based on the encapsulated knowledge relevant to the perspective (Bredeweg *et al*. 2006).

3.2 Assumptions

Conceptually, modelling assumptions fall into two categories: simplifying and operating assumptions (Falkenhainer and Forbus, 1991). *Simplifying assumptions* are used to make explicit how knowledge details such as the underlying perspective, approximations, and level of granularity are represented in the model

fragments. Simplifying assumptions are classified as (a) *ontological* assumptions, to provide the vocabulary used in the model, explicating what kinds of things exist and what sort of relationships between them can be held; (b) *grain* assumptions, to define the level of details represented in the model, perhaps aggregating some features and ignoring others; (c) *approximation* assumptions, to make models that are easy to use, sometimes at the cost of accuracy; and, often intertwined with approximation assumptions, (d) *abstraction* assumptions, used to reduce the complexity of the modelling language, usually reducing information available and increasing ambiguity.

Operating assumptions are used to manage complexity. In a way, they give focus to the simulation, by implementing constraints so that the model describes the behaviour relevant for answering specific questions. Three types of operating assumptions are considered here: (a) *local restrictions*: restrictions on quantity values implemented by means of inequalities between quantities and constants (e.g. *number_of* >0); (b) *operation mode*: a 'general assumption' that controls a collection of local restrictions; and (c) *steady-state* assumptions: determine that all derivatives for some class of parameters have value zero. Ultimately, operating assumptions increase the efficiency of the simulation by ruling out entire classes of behaviour (e.g. immigration and emigration in closed population dynamics), and by indicating the range of parameter values for which certain approximations are valid (e.g. birth rate can only exist when *number_of* >0).

3.3 Garp3

The model was implemented in Garp3 (www.garp3.org), a qualitative reasoning workbench that provides a graphical interface for building models and inspecting simulations (Bredeweg *et al*., 2006). Garp3 models are created around entities, modelling primitives used to represent relevant objects of interest. Their continuous properties are represented as quantities. Possible qualitative values are represented in quantity spaces (QS), typically an ordered set of points and intervals. It may happen that specific values of two quantities always co-occur, as for example, the number of individuals and the biomass of the population. This notion is captured by means of correspondences, that can involve specific values or the whole quantity space, and can be either direct (e.g. *large* corresponds to *large*) or inverse (e.g. *large* corresponds to *small*).

Following Forbus (1984), it is assumed that changes in the system are initiated by processes, which are modelled as direct influences ($I\pm$). Qualitative proportionalities propagate the effects of processes to other quantities ($P\pm$). Knowledge is represented in model fragments automatically selected by Garp3 to create representations of qualitative states of the system during a simulation. A particular type of model fragment, Agent, is used to model external factors that cause changes in the system. Both entities and model fragments are organized in a hierarchical way, so that features described at higher levels are inherited by the lower levels. A scenario describes the system structure and initial values of some quantities to be considered in the simulation. Garp3 allows for representations of exogenous variables in the scenario, assigning them complex behaviour (e.g. random or sinusoidal) that is not motivated within the system being modelled (Bredeweg *et al*., 2007).

Garp3 provides two useful modelling primitives to enforce the selection of certain model fragments: assumptions and attributes. An assumption identifies specific model fragments that implement particular features or conditions for causal relations (influences and proportionalities) to become active. Attributes are special labels that can be attached to a particular entity for defining features that can take fixed values, so that different instances of the same entity can be created. Both assumptions and attributtes should be included in the scenario in order to activate model fragments with the same assumptions and attributes during the simulation.

4. Describing the Riacho Fundo model

Four perspectives are defined to organize the library of model fragments about sustainable development in Naturnet-Redime case studies: *Natural*, *Rural*, *Urban* and *Social*. The *Natural* perspective relates to natural phenomena, including ecosystem services (Daily *et al*., 1997; Alcamo *et al*., 2005). The *Rural* perspective focus on human activities aiming at exploring natural resources for economic purposes (Castells, 1996; Garrity, 2004). The *Urban*

perspective addresses the city and its physical and communications infrastructure, its dependence on resources coming from outside and its own metabolism, as discussed by Egger (2006). Finally, the *Social* perspective is related to economy, governance, culture and human well being (Colby, 1991; Castells, 1996; Dodds, 1997; Egger, 2006). It is possible to have elements from all of the four perspectives combined in a single simulation model. Note that the Riacho Fundo model includes two of these four perspectives (*Rural* and *Urban*) and the *Semi-urban* perspective. Elements of *Social* and *Natural* are combined to the other perspectives.

Table 1 summarizes the main concepts addressed in each perspective of the Riacho Fundo model, selected in accordance to expert and stakeholders' opinions as discussed in section 2.

	Perspectives		
Land use	*Urban*	*Semi-urban*	*Rural*
Main problems	Drainage system; flooded areas; transported garbage and damage caused by floods	Urbanization; water infiltration; and soil erosion	Erosion; loss of water resources and biodiversity
Economic features	Services: garages	Industry: textile and food industries	Agriculture: cattle; crops
Soil	Impermeable soil	Soil particle aggregation	Soil fertility
Water resources	Effects of uncontrolled flow of water run off and of the drainage system	Effects of erosion and underground water on springs and rivers	Effects of erosion and underground water on springs and streams
Biological entities	Mosquitos, Pathogens	Vegetation	Vegetation; Vertebrates; Capybara
Human	Economic activities; Human well-being: garbage and water related diseases	Economic activities	Economic activities
Agents	Rainfall	Urbanization	--
Sustainability	Control of diseases; Control of residues	Water quality; Control of residues	Soil fertilization; Reuse of residues

Table 1. Overview of the main concepts addressed by the Riacho Fundo model.

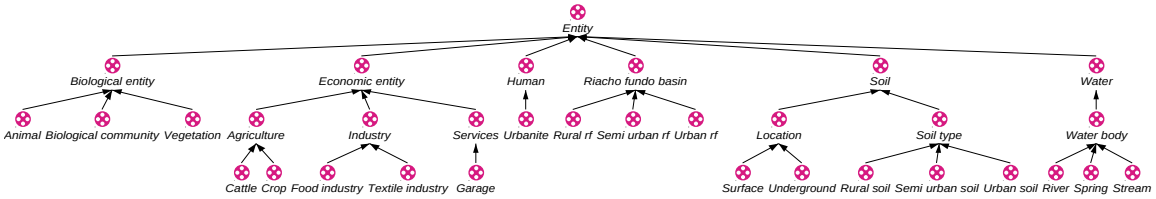


Figure 1. Entities used in the Riacho Fundo model.

4.1 Entities

The system structure is representated with entities and configurations in Garp3 models. Figure 1 shows the entity hierarchy used in the Riacho Fundo model.

Three entities define the implementation of perspectives as discussed in section 3: 'Rural rf', 'Semi-urban rf' and 'Urban rf'. These entities represent types of land use, and are associated to three types of soil: 'Rural soil', 'Semi-urban soil' and 'Urban soil'. Economic activities are represented by the entities 'Agriculture', 'Industry' and 'Services', respectively associated to each type of land use. Other entities represent relevant types of biological resources and water bodies, and a particular type of human being, the urbanites. Figure 2, a screenshot of a simulation in Garp3, shows the complete Riacho Fundo system structure. Simulation models created in the three perspectives explore part of this system structure, as discussed in the following sections.

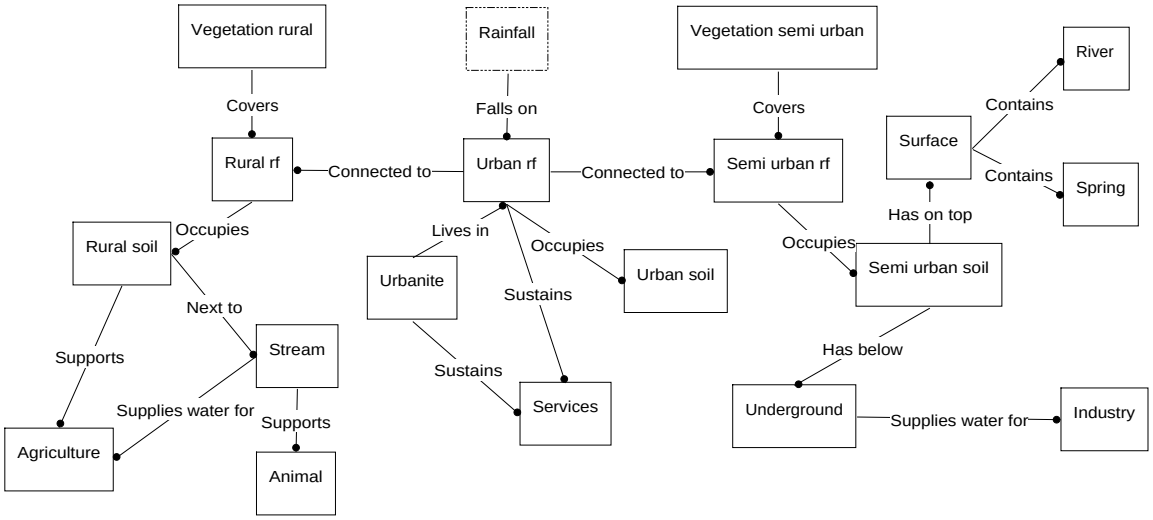


Figure 2. System structure showing all the entities and configurations included in the Riacho Fundo model.

4.2 The *Rural* perspective

Changes in vegetation cover drive the dynamics of soil fertility, water, biodiversity and the agricultural production in the *Rural* perspective. Initially, the balance between regeneration and degradation of the vegetation determines the vegetation growth process and set the value of *Growth rate*, a direct influence on *Vegetation cover*. This quantity has a negative effect on the area degraded by erosion by means of influencing soil particles aggregation, modelled with two proportionalities: P+(*Level of aggregation*, *Vegetation cover*) and P–(*Eroded land*, *Level of aggregation*). When *Eroded land* increases, it causes the amount of nutrients in the soil to decrease, which in turn causes the quantity *Fertility* to decrease.

Agriculture is represented in the model as cattle and crop production. Resource inflow for cattle production comes from soil fertility; residues produced by cattle can either become organic pollution in water bodies or be used as manure to add nutrients to soil fertility and as such become part of the resource inflow for agriculture. Crop production depends on irrigation, being the water abstracted from a water body; residues are associated to pesticides and may also pollute water bodies.

Erosion triggers another causal chain, leading to sediment deposition in water bodies. A simplified version of erosion associates soil aggregation to the quantity *Removed soil* via a positive proportionality: P+(*Removed soil*, *Level of aggregation*). Next, *Removed soil* is connected to the quantity *Sediment* of the entities 'Spring' and 'Stream' in separate model fragments, allowing for simulations that explore the consequences of erosion for both types of water bodies, either together or separate.

Depth and amount of water in streams in the Riacho Fundo basin have been associated to the survival of animals, in particular of capybaras. Such relation is captured in two ways. A simplified version is implemented by means of proportionalities in the causal chain *Amount of water* → *Vertebrate survival* → *Animal biodiversity*. A detailed version describes the animal population growth process (reusing a generic model fragment that applies to all biological entities, e.g. vegetation).

4.3 The *Semi-urban* perspective

Large areas of the Riacho Fundo basin are changing due to urbanization. Models in the *Semi-urban* perspective capture this pressure to provide a different view on features already addressed by the *Rural* perspective, such as soil aggregation and erosion and their consequences to water bodies and to economic activities in the basin.

Soil aggregation is represented as a process, in which the rate is influenced by an agent (Urbanization), and a negative feedback is used

to assure that the process stops when the level of aggregation reaches its maximum value. This detailed description is important to set the effects of urbanization on two other processes: water infiltration and erosion. The quantity *Level of aggregation* influences the infiltration and erosion processes via their rates: when aggregation increases, both infiltration rate and erosion rate decrease. The basic mechanism of water infiltration in the soil is represented in Figure 3.

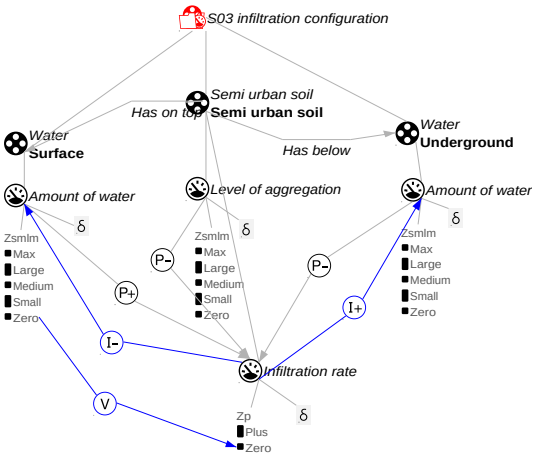


Figure 3. Model fragment representing the Infiltration process.

Two other model fragments are used to make explicit the conditions for this process to become active: in 'Infiltration active', if the inequalities *Amount of water* > *zero* (at the surface) and *Level of aggregation* < *maximum* hold, the rate gets the value *plus*; in 'Infiltration inactive', if *Amount of water* = *maximum* (at the underground), then *Infiltration rate* = *zero*. This mechanism implements the notion of saturation, useful function for modelling biological systems (Haefner, 2005).

Water supply for industrial activities comes from different sources: springs, river and underground water. Two types of industries are included in the Riacho Fundo *Semi-urban* perspective: textile and food processing. Pollutants produced by these industries include chemical and organic substances.

4.4 The *Urban* perspective

The main aspect explored by the *Urban* perspective are the effects of an engineered drainage system. Pairs of scenarios show the outcomes of a particular situation in which the uncontrolled flow of rain water affects different aspects of urban areas, both with and without the drainage system.

Two direct consequences of uncontrolled water are represented in the model: garbage transportation and floods. Garbage is seem as residues from economic activities. Economic damages caused by floods include the destruction of public and private assets. Quantity *Flooded areas* stimulates the increase of mosquito populations, and some of them can be associated to diseases such as dengue fever, a real problem in the Riacho Fundo basin. However, the current version of the model represents only the increase of generic pathogens, that may cause a number of water related diseases.

Finally, the quality of life of the urbanites is represented as a balance between generic positive and negative factors, used to calculate the rate of the improvement of life quality process, and this quantity is a positive direct influence on *Well-being*. In the current implementation of the model, garbage and pathogens are associated to the negative factors, and specific assumptions are used to control the interaction between these quantities.

4.5 Economic activities

Economic activities are modelled in generic terms so that a unique set of model fragments can be reused to represent different activities. Input of resources is represented by the quantity *Resource inflow*, and the use of resources, by *Resource consumption*. A qualitative subtraction combines these two quantities to calculate the value of *Production rate*. This rate may cause the quantities *Product* and *Residue* to increase, decrease or remain stable. This model fragment is shown in Figure 4:

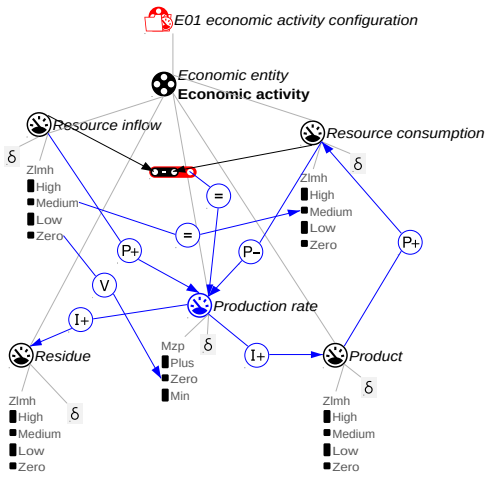


Figure 4. Model fragment representing a generic economic production process.

Additional model fragments define specific types of products and residues related to economic activities. For example, residue produced by the cattle is organic matter that can be used as fertilizer. Accordingly, a model fragment represents *Manure* as corresponding to *Residue*. Similar model fragments implement correspondences to *Pesticides*, *Chemical pollution*, *Organic pollution* and other quantities according to the type of economic activity.

5. Simulating sustainability in Riacho Fundo

The current implementation of the model supports 48 simulations, exploring the three perspectives. Within each perspective, simulations exhibit increasing levels of complexity. Initially only basic processes and mechanisms are simulated. Next, different basic processes are combined with other elements in order to compose more complex simulation models, building up the knowledge available in the library until an overview of the perspective is achieved. Due to space restrictions only one simulation is described in detail here. More details are available at www.naturnet.org.

5.1 *Rural* perspective

Simple simulations explore, for example, only vegetation dynamics (four quantities); vegetation, eroded land and fertility (eight quantities); erosion, stream and biodiversity (10 quantities); vegetation, erosion, and fertility determined by soil nutrients and manure (14 quantities). The more complex simulation in the current implementation of the model involves 20 quantities related to vegetation, erosion, biodiversity and fertility.

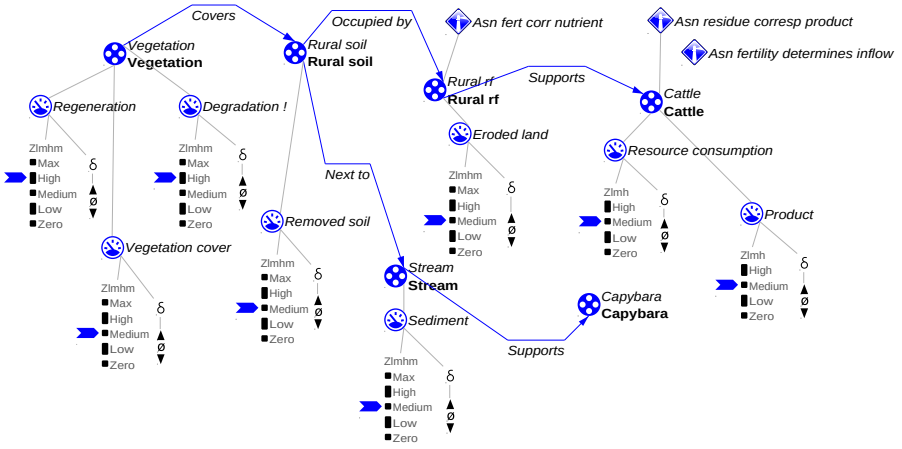


Figure 5. Initial scenario 'Vegetation, erosion, biodiversity and resource inflow determined by fertility', from the *Rural* perspective on sustainability in Riacho Fundo.

Figure 5 shows this intial scenario. The following assumptions hold in this scenario: 'Fertility corresponds to nutrient', 'Fertility determines (resource) inflow' and 'Residues correspond to product'. The exclamation mark that follows the quantity *Degradation* indicates its behaviour is assumed to be exogenously driven (Bredeweg *et al.*, 2007), in this case, constant. The simulation involves 20 quantities and produces three initial states; the full simulation, 85 states.

As in the scenario the initial values of *Regeneration* and *Degradation* are in the interval

high, the situation is ambiguous and the value of *Vegetation cover* is undefined. Accordingly, in the initial states, the system may be in equilibrium, with *Vegetation cover* and all the quantities constant and steady (state 1), or *Vegetation cover* can be either increasing (state 2) or decreasing (state 3). The causal model, as it appears in state 3, is shown in Figure 6. The causal model shows that *Growth rate* is negative and *Vegetation cover* has value <*medium*,– >.

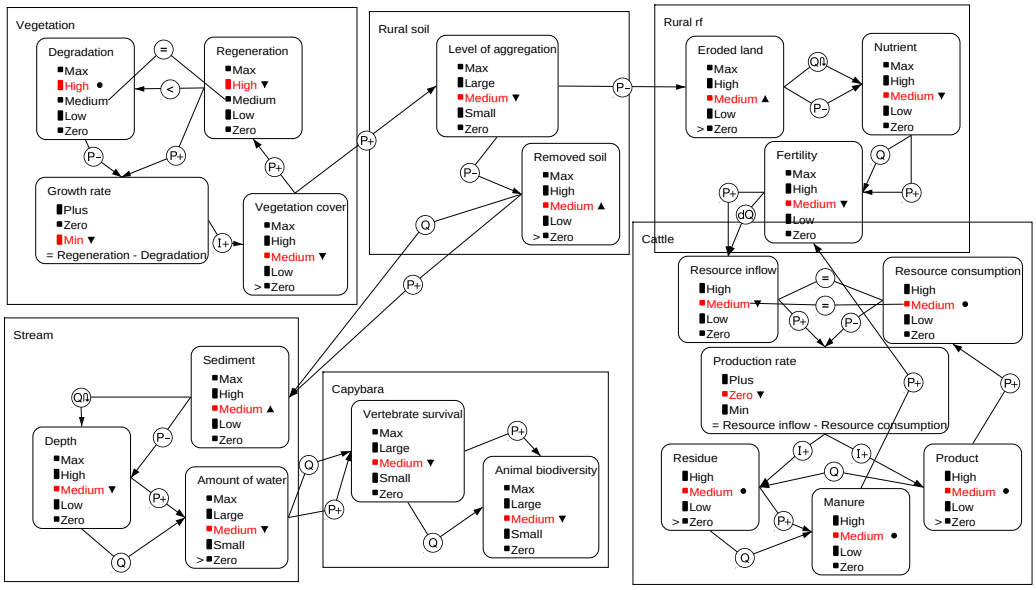


Figure 6. Causal model as it appears in state 3 of the simulation 'Vegetation, erosion, biodiversity and resource inflow determined by fertility', *Rural* perspective.

As a consequence, *Level of aggregation* also decreases, and this tendency propagates in two causal chains. On the one side, *Removed soil* and the amount of *Sediment* in the stream increase, and *Depth* decreases. *Amount of water* also decreases, a tendency that propagates to both *Vertebrate survival* and *Animal biodiversity*. On the other side of the causal chain, soil aggregation causes *Eroded land* to increase, leading *Nutrient* and *Fertility* to decrease. Note that *Fertility* could also be influenced by *Manure*, but is this particular state the latter quantity is steady, so the proportionality is inactive. Following the assumption 'Fertility determines (resource) inflow', *Resource inflow* is decreasing and, as a consequence, the equilibrium between this quantity and *Resource consumption* is broken. *Production rate*, which has the value zero, will decrease in the following state (in this state it has a negative derivative). *Production rate* = *zero* means that both *Product* and *Residue* are steady, and the proportionalities put by these quantities on *Resource consumption* and *Manure*, respectively, are inactive.

One of the possible outcomes of this simulation is the behaviour path [3 → 4 → 11 → 22 → 81 → 82]. In this case, *Vegetation cover* goes to *zero* and *Removed* soil goes to *maximum*, eventually causing the disappearance of the stream and of capybaras, representing biodiversity loss. Similarly, *Eroded land* also goes to *maximum*, *Fertility* goes to *zero* and the whole productive system collapses. The values of relevant quantities in this path are shown in Figure 7. A behaviour path starting in state 2 produces basically opposite results: *Vegetation cover* increases, and reduces the erosion process. As a consequence, either the amount of water in the stream may go to *maximum*, leading biodiversity to higher values as well, or soil fertility also goes to *maximum*, leading the cattle production to higher levels. (Figure 8).

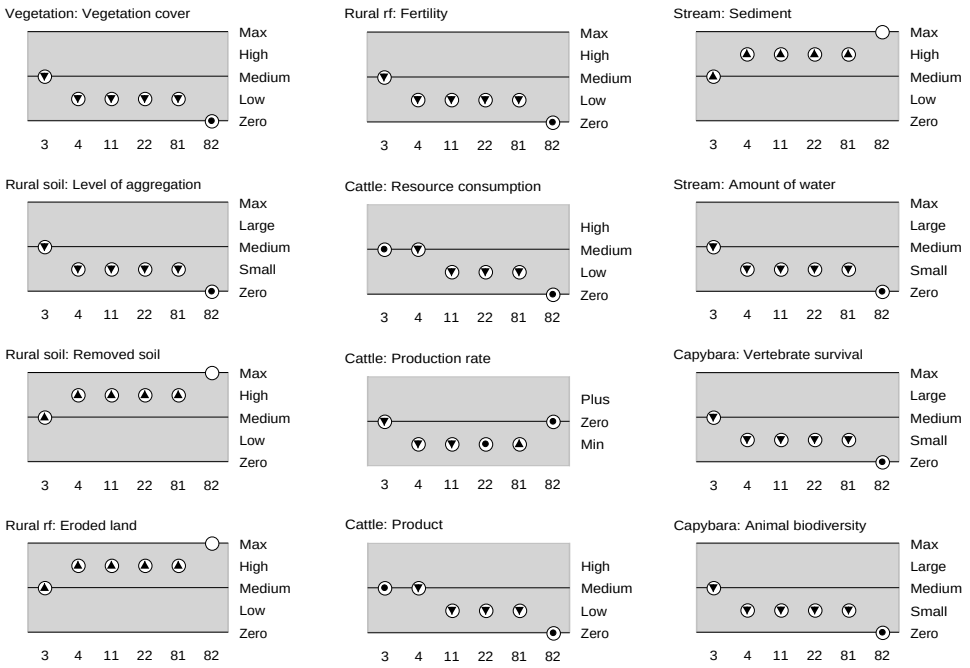


Figure 7. Value history of selected quantities in a simulation 'Vegetation, erosion, biodiversity and resource inflow determined by fertility', *Rural* perspective.

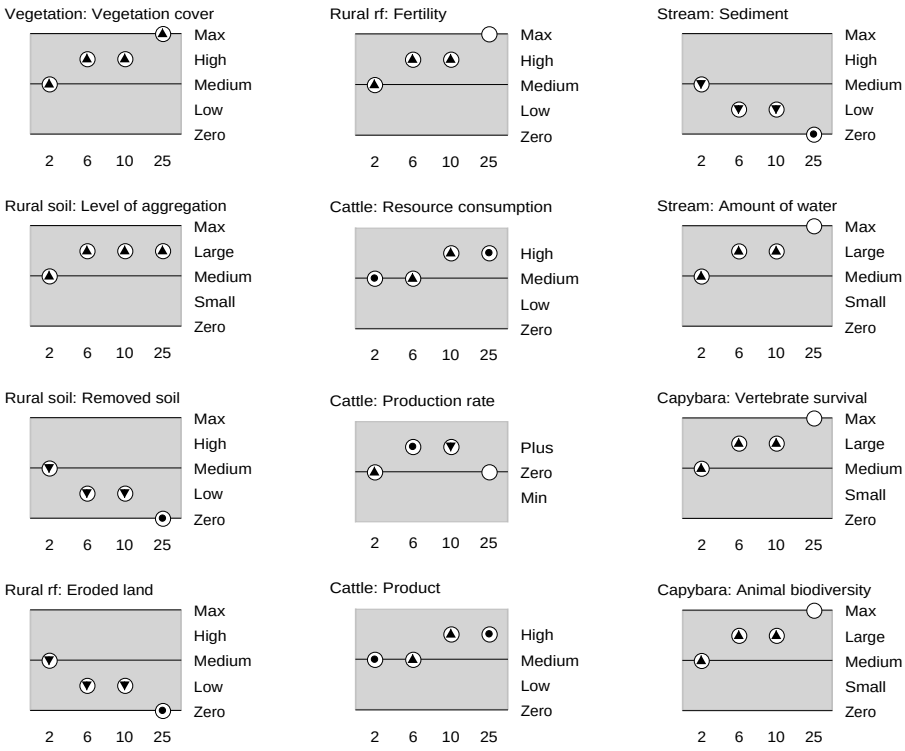


Figure 8. Value history of selected quantities in a simulation 'Vegetation, erosion, biodiversity and resource inflow determined by fertility', *Rural* perspective.

5.2 *Semi-urban* perspective

Simulations exploring only one process or basic components in the *Semi-urban* perspective show the aggregation process (two quantities), infiltration (five quantities) and erosion (five quantities). Complexity increases when erosion and the conditions of springs (eight quantities), or infiltration and springs (11 quantities) are combined, and when economic activities are included.

Figure 9 shows the causal model for a simulation that includes the effects of erosion on springs and on the water is being supplied to the food industry (14 quantities). In this simulation, for example, one of the behaviour paths show that, although the level of aggregation increases and erosion rate decreases, the amount of removed soil goes up to *maximum* and eventually causes the amount of water in the spring to become *zero*. As this is the main resource for the industry, production also goes to *zero* and the whole productive system collapses.

5.3 *Urban* perspective

Simulations in this perspective allow comparison of situations in which there is no drainage system, to those in which the flow of controlled water is increasing. The simplest simulations demonstrate the mechanism of drainage (involving seven quantities), production of garbage (seven quantities), growth of mosquito populations (two quantities or four quantities, if the details of the process are included), and the mechanism of well-being improvement (four quantities). Simulations with intermediate level of complexity explore, for instance, the importance of the drainage system for: controlling flooded areas and water related diseases (eight quantities); mosquito populations (nine quantities); eliminating garbage transportation (15 quantities) and, in doing that, to reduce negative factors on well-being (19 quantities). The most complex simulations in the current implementation (Figure 10) involve 22 quantities and include all the elements mentioned in section 4.4.

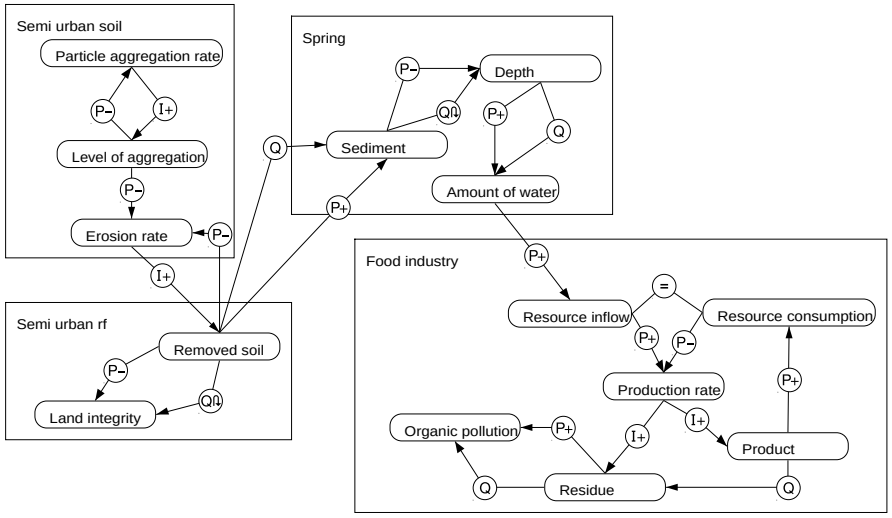


Figure 9. Causal model obtained in state 5 of the simulation 'Erosion, springs and water supply to food industry", *Semi-urban* perspective.

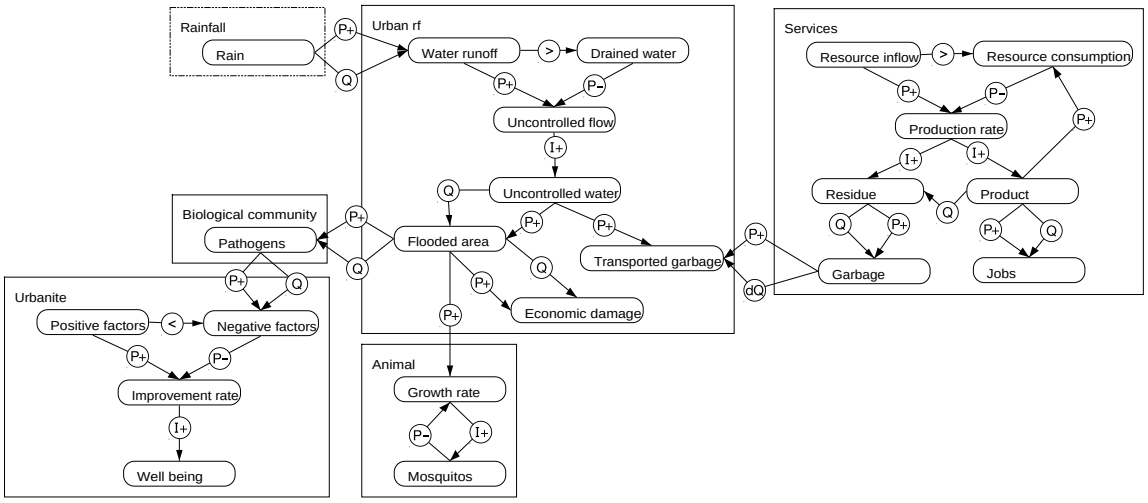


Figure 10. Causal model obtained in state 1 of the simulation 'Drainage increasing, transported garbage and well-being', *Urban* perspective.

In this simulation, it is assumed that *Rain* and *Water runoff* are constant and steady (<*medium,0*>) and *Drained water* starts in zero, but increasing. Until this quantity reaches the value *medium*, the overall situation worsens, with increasing values of quantities such as *Flooded area*, *Economic damage* and *Negative factors* on well-being. However, as soon as *Drained water* > *Water runoff*, the situation improves and, at the end of the simulation, *Well-being* has value *high*.

6. Discussion

This paper describes the use of perspectives to organize a large library of model fragments in order to create sets of simulation models each addressing a class of sustainability issues in the Riacho Fundo basin. Three perspectives (*Rural*, *Semi-urban* and *Urban*) are taken to represent different types of land use, and a wide range of assumptions were defined to implement these perspectives.

The use of assumptions for reasoning with multiple models has a long tradition in Qualitative Reasoning (Bobrow, 1984). de Kleer and Brown (1984) point out the importance of making modelling assumptions explicit and of changing them during problem solving. A number of authors have been working on developing algorithms for automatically selecting or changing models according to certain assumptions. For example, Addanki *et al*. (1991) represent domain knowledge as graph of models and change assumptions to move from one model to the other; and Liu and Farley (1991) took a different approach to automate task-driven reasoning about physical systems using multiple perspectives. Falkenhainer and Forbus (1991) developed compositional modelling, a technique to decompose domain knowledge into model fragments, and implemented an algorithm for model composition given a domain theory, a structural description of the system and a query to be answered. Rickel and Porter (1997), using the compositional modelling approach, developed an algorithm to build the simplest adequate model from building blocks (single variables and influences) for answering prediction questions within a certain time scale, and tested it in the domain of botany and plant physiology.

Differently from these previous approaches, the work described here addresses sustainability using no numerical information or mathematical functions to define perspectives or to implement assumptions, only qualitative representations of concepts. Garp3's representational *apparatus* and algorithm are explored to capture ecological knowledge and to create alternative models according to the perspective taken. The first element used to create a simulation model taking a certain perspective are the entities 'Rural rf', 'Semi-urban rf' and'Urban rf'. Increasing levels of complexity can be further obtained by means of the inclusion of new entities in the system structure.

In fact, control over entities and quantities introduced in the model is an important and quite effective use of simplifying assumptions to implement perpectives. Considering that: each entity can be associated to a number of quantities; each quantity can be modelled using different quantity spaces; and each qualitative value represents a qualitative state of the entity, the choice of entities, quantities and quantity space defines specific vocabulary for a certain perspective. For example, different types of economic activities can be associated to any perspective taken in the Riacho Fundo model (section 4.1). Besides that, the set of model fragments created to identify residues produced by different types of economic activities (section 4.5) provides adequate vocabulary for each perpective. This way, entities 'Urban rf' and 'Garage' used in *Urban* perspective introduce vocabulary to describe how garbage produced can be transported by uncontrolled rain water runoff and affect human well being.

Grain assumptions provide different levels of details to some relevant phenomena that reappear in different contexts. Erosion is a well developed example in the Riacho Fundo model. When the *Semi-urban* perspective is taken (section 5.2), the soil aggregation process defines the value of *Level of Aggregation*, which in turn influences *Erosion rate*, and this process defines the value of *Removed soil*. A less detailed representation is adopted in *Rural* perspective models (section 5.1): *Vegetation cover* indirectly influences *Level of Aggregation* and this quantity also indirectly influences *Removed soil*. Similar options are available to represent population growth of capybara (section 5.1) and mosquitos (section 5.3).

Closely related to these assumptions, approximations can produce simpler accounts for the same phenomenum that are easier to use at the cost of accuracy. For example, disappearance of springs can be addressed in simulation models when both *Rural* and *Semi-urban* perspectives are taken (sections 4.2 and 4.3). As processes soil aggregation and erosion are not explicitly described in the *Rural* perspective, a model on this topic is easier to use than a similar model in the *Semi-urban* perspective.

Operating assumptions can be used both to give focus and to reduce the complexity of the simulations. For example, in the *Semi-urban* perspective models disappearance of functional springs can be caused by erosion and/or lack of undergroung water (Figure 3). Garp3 model ingredient Attributes was used to capture these possibilities: entity 'Spring' has an attribute 'Focus', with two possible values: 'Effects of erosion' and 'Effects of infiltration'. Depending on the attribute value introduced in the scenario, two independent causal chains may become active: (a) 'Focus: Effect of erosion': *Level of aggregation* → *Removed soil* → *Sediment* → *Depth* → *Amount of water*; (b) 'Focus: Effect of infiltration': *Level of aggregation* → *Underground water* → *Amount of water*. An additional model fragment, in which 'Springs' has no attributes, allows for expressing simultaneous effects of erosion and infiltration on the springs.

Similarly, different causal chains can be constructed within the *Rural* perspective, depending on the use of focus operating assumptions. Soil fertility can be determined in three ways: (a) by assuming that *Fertility* values correspond to *Nutrient* values; (b) by considering that vegetation cover determines the amount of organic matter, and calculating *Fertility* = *Organic matter* + *Nutrient*; and (c) by considering the combination of nutrients and manure, a by-product of cattle livestock (see Figures 6-8). Two assumptions take care of options (a) and (b). If no assumption is introduced in the scenario, option (c) is selected.

Operating assumptions are used to reduce complexity in simulations either by reducing ambiguity or preventing some behaviours to happen. Local restrictions, implemented as correspondences, were widely used in the Riacho Fundo model to reduce ambiguity and, as such, to reduce the number of states in the simulation. For example, directed correspondences between quantity values express co-occurences of values *zero* (e.g. Figures 3 and 4); correspondences between quantity spaces, co-occurrence of all possible values of two quantities (see 'Q' relations in Figures 6, 9 and 10). Inverse correspondences represent co-occurrence of opposite values of two quantities (see, for example, the Q↓ relation between *Sediment* and *Depth* in Figures 6 and 9). Finally, correspondences between derivatives significantly reduce ambiguity in the simulation, as they determine the strongest influence when two or more proportionalities apply to the same quantity. For example, it was used to enforce *Transported garbage* to take the value of the

derivative of *Garbage*, and not of *Uncontrolled water* in *Urban* perspective (see the dQ relation in Figure 10).

Local restrictions may also be implemented by means of inequalities. Examples include definitions of the level of pollution produced by economic activities: a fair level is set by assuming *Residue* < *medium*, no matter the amount of products; less sustainable options are *Product* ≤ *Residue* and *Residue* corresponds to *Product* (correspondence between the quantity spaces of the two quantities). As these assumptions are implemented at the level of 'Economic entity', they are applicable to the three perspectives.

Steady state assumptions reduce complexity by giving a unique behaviour to a quantity (decreasing, steady, increasing), and can be implemented both as exogenous quantities and in model fragments. An example of the former is presented in Figure 5, a scenario in which the quantity *Degradation* which is assumed to have value *high* constant. Note that exogenous quantities may express more complex behaviours (Bredeweg *et al*., 2007) Steady state assumptions may also involve quantity magnitudes or derivatives when implemented in model fragments. In the Riacho Fundo model examples may be found in the three perspectives (e.g. *Drained water* = <*zero*, *zero*> and *Drained water* = <*?, +*> in *Urban* perspective).

The contents of the Riacho Fundo model is in accordance to stakeholders demands (section 2). From the technical point of view, perspective-taken simulation models correctly provide views to sustainability in the basin. Assumptions are conceptually clear and pedagogical. However, some problems remain. Models implementing *Natural* and *Social perspectives* are still lacking. The use of hierarchies of model fragments and entities and of other modelling primitives should be optimized. Integration of perspectives is an issue, as ambiguity surfaces when unrelated quantities are included in the same simulation model. New modelling assumtions will become necessary to take care of integrated simulations. A point that was not addressed here was the issue of shifting from one perspective to another. Identifying the requirements for such transitions will lead to better understand the nature of perspective-taking in qualitative reasoning (Liu and Farley, 1991).

7. Conclusions

The Riacho Fundo model comprises, in its current implementation, 33 entities, 9 processes and 48 quantities, organized in 112 model fragments. It has 48 scenarios that simulate different subsets of the whole system structure. The three perspectives – *Urban*, *Semi-urban* and *Rural* – proved to be efficient in creating simulations about relevant aspects of sustainability in the Riacho Fundo basin.

Simplifying assumptions facilitate vocabulary creation for each perspective, as they are used to control how entities, quantities and quantity values are introduced in the simulations. Assumptions are also effective to implement alternative views on similar phenomena, shifting from coarse to fine grained representations, according to the perspective taken. Operating assumptions provide focus and reduce complexity of simulations within each perspective.

Garp3 is an interesting tool for implementing compositional models, as it provides a rich modelling language for expressing both model components and assumptions constraining their use. Some of Garp3 modelling primitives, such as entities and configurations, attributes and agents are particularly useful for implementing perspectives. Model fragments, inequality relations, correspondences and exogenous quantities are particularly suited for implementing both simplifying and operating assumptions. This way, besides being functionally important, assumptions were also conceptually aligned to the rest of the domain knowledge represented in the library.

Lessons learned during the modelling effort described here will be useful for improving stakeholders' understanding about the problems they face. Sustainability is a complex issue, and learning about its multitudinous aspects is an intergenerational commitment for the current generation, that should properly take care of river basins still rich in natural resources and rehabilitate the damaged ones, while promoting human development for those who live in these areas.

Acknowledgements The work described in this paper is co-funded by the Commission of European Communities, project Naturnet – Redime, EU STREP, contract number 004074.

We are grateful to our colleagues of the project and to the anonymous reviewers for the valuable comments to the work described here. One of the authors (PS) would like to thank the support offered by the Human Computer Studies Laboratory at the University of Amsterdam, where part of this work was developed.

References

Addanki, S.; Cremonini, R. & Penberthy, J.S. (1991) Graphs of Models. *Artificial Intelligence*, 51: 145-177.

Alcamo, J.; van Vuuren, D.; Ringler, C.; Cramer, W.; Masui, T.; Alder, J.; and Schulze; K. (2005) Changes in nature's balance sheet: model-based estimates of future worldwide ecosystem services. *Ecology and Society* 10(2): 19. [online]

Bobrow, D.G. (ed.) (1984) *Qualitative Reasoning about Physical Systems*. Elsevier Science, Amsterdam, The Netherlands.

Bredeweg, B., Bouwer, A., Jellema, J., Bertels, D., Linnebank, F. and Liem, J. (2006) Garp3 - A new Workbench for Qualitative Reasoning and Modelling. In Bailey-Kellogg, C., and Kuipers, B. (eds) *Proceedings of the 20th International Workshop on Qualitative Reasoning* (QR-06), pages 21-28, Hanover, New Hampshire, USA, 10-12 July 2006.

Bredeweg, B.; Salles, P. and Nuttle, T. (2007) Using exogenous quantities in qualitative models about environmental sustainability. *AI Communications*, Volume 20, Number 1, pp. 49-58.

Castells, M (1996) *The rise of the network society*. vol. I. Oxford, UK, Blackwell Pubs.

Colby, M.E. (1991) Environmental management in development: the evolution of paradigms. *Ecological Economics*, 3: 193-213.

Daily, G.C.; Alexander, S.; Ehrlich, P.R.; Goulder, L.; Lubchenco, J.; Matson, P.A.; Mooney, H.A.; Postel, S.; Schneider, S.H.; Tilman, D. and Woodwell, G.M. (1997) Ecosystem Services: Benefits Supplied to Human Societies by Natural Ecosystems. *Issues* in Ecology, Number 2, Spring 1997, pp 1-16.

de Kleer, J. & Brown, J.S. (1984) A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24: 7-83.

Dodds, S. (1997) Towards a 'science of sustainability': Improving the way ecological economics understands human well-being. *Ecological Economics* 23: 95-111.

Egger, S. (2006) Determining a sustainable city model. *Environmental Modelling and Software* 21: 1235-1246.

Falkenhainer, B. & Forbus, K. (1991) Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence*, 51(1-3): 95-143.

Forbus, K. (1984) Qualitative Process Theory. *Artificial Intelligence*, 24: 85-168.

Garrity, D.P. (2004) Agroforestry and the achievement of the Millennium Development Goals. *Agroforestry Systems* 61: 5–17.

Haefner, J.W. (2005) *Modeling Biological Systems: Principles and Applications*. New York, N.Y.: Springer.

Liu, Z-Y. and Farley, A.M. (1991) Tasks, Models, Perspectives, Dimensions. In Kuipers, B. (ed.) *Proceedings of the 5th International Workshop on Qualitative Reasoning (QR-91)*, pages 1-9, Austin, TX, 19-21 May 1991.

Rickel, J.& Porter, B. (1997) Automated Modeling of Complex Systems to Answer Prediction Questions. *Artificial Intelligence*, 93: 201-260.

Salles, P. (2001) Comitê de Gestão da Bacia do Paranoá. In: Fonseca, F.O. (org.) *Olhares sobre o Lago Paranoá*. Brasília: Secretaria de Meio Ambiente e Recursos Hídricos, pp. 296-307.

Salles, P. and Caldas, A.L.R. (2006) *Textual description of Riacho Fundo case study focusing on basic biological, physical, and chemical processes related to the environment*. Deliverable D6.4.1, NaturNet-Redime, EU STREP, project number 004074. Available at www.naturnet.org (access in 25/02/2007).

M^2CIRQ: Qualitative fluid flow modelling for aerospace FMEA applications

Neal Snooke
Department of Computer Science
University of Wales Aberystwyth
Penglais, Aberystwyth
Ceredigion, SY23 3DB, U.K.

Abstract

This paper presents fluid flow system simulation using the MCIRQ qualitative simulator. MCIRQ was designed as an electrical simulator, however this work exploits the close analogy between fluid flow and electrical current at the level of qualitative behaviour. The core qualitative flow algorithm is applicable to both domains but there are differences in the systems structures and assumptions that require additional modelling. The concepts of multiple source networks, and explicit propagation of multiple substances through a network, are necessary to model important characteristics of fluid flow networks. Both of these characteristics are developed on top of the MCIRQ simulator with the aim to produce an automated FMEA for aircraft fuel systems similar to previously developed automated electrical FMEA.

Introduction

This paper describes a circuit-based approach to modelling both hydraulic and fluid flow systems. The approach is based on the MCIRQ multi-level qualitative flow simulator used for simulation of electrical circuits (Lee 1999; Price, Snooke, & Lewis 2003). This work proposes two main enhancements to the simulator ontology:

- multiple pump configurations are common in fluid systems and require multiple connected power sources within a single flow network;
- an explicit representation of the substance being propagated is necessary both to reason about the effect of faults, such as leaks that allow ingress of air or escape of fluid, as well as to represent the states of components such as tanks.

These features allow simulation of many significant behaviours and potential failures of an aircraft fuel system that is the application area of this work. Important characteristics include: emptying and filling of tanks, flow of fuel and air within a system, and gravity based flow. Important faults include blocked and stuck valves, blockages in pipes and vents, pump failure and inefficiency, and leaking components.

New capability is added on top MCIRQ algorithm by allowing multiple execution of the analysis together with additional representation of substances at the connection points of the network. This additional layer is referred to as M^2CIRQ in this paper.

The focus of the work is on modelling the fuel transfer systems of multi engine aircraft. This paper presents the modelling concepts developed and a simple demonstration example, avoiding unnecessary complexity and commercial issues associated with presenting the real system.

Qualitative fluid flow

The approach taken in this work separates global system characteristics and the local causal effects of components. The global flow behaviors are predicted by MCIRQ, leaving the modeler with only component behavior to define. This creates a more natural modeling environment than can be the case with general qualitative constraint systems such as QSIM (Kuipers 1994) or (Kitamura, Ikeda, & Mizoguchi 1996), although the latter author deals with thermal effects in addition to flow characteristics. The MCIRQ circuit analyzer described in (Lee & Ormsby 1994; Lee 2000) is a global flow based simulator and we exploit the analogy between electrical current and fluid flow as discussed in (Chittaro & Rannon 1999) however there are several significant differences:

- Electrical systems normally only have a single power source, but fluid systems often have several pumps used in different operating modes or configurations. If gravity feed is present or siphonic behaviour is required then gravity can me modelled as a weak pump for example.
- Several significant substances may be required in a fluid system in contrast to electrical systems.
- The movement of the fluid is far more significant in fluid systems than flow of charge in electrical systems.
- The storage of substance and the energy sources (pumps) are not usually the same component as is the case for a battery or PSU.
- The gross movement of charge –for example battery discharge– is often ignored in electrical analyses, however the capacity and storage of fluids and gasses in fluid systems is central to the system operation.

Overview of MCIRQ

MCIRQ provides qualitative order of magnitude analysis of resistive networks. A network is comprised of a set of *arcs* that represent resistances. Each arc is associated with exactly two *nodes*. Two arcs are connected if they share a node. MCIRQ requires a resistive value to be assigned to each arc from an order-of-magnitude sequence. The minimal set of resistance values is $\{\text{zero} < \text{load} < \text{infinite}\}$, although this work uses five levels for electrical and fluid circuits. Each network has exactly one positive (+) node and exactly one negative (-) node.

Figure 1 shows a network using the resistive values $\{\text{zero} < \text{low} < \text{medium} < \text{high} < \text{infinite}\}$. The resulting analysis is shown as the flow values from the ordered set $\{\mathrm{F}_0 < \mathrm{F}_{\text{low}} < \mathrm{F}_{\text{medium}} < \mathrm{F}_{\text{high}} < \mathrm{F}_{\text{short}}\}$. The details of the analysis algorithm are presented in detail in[Lee00a].

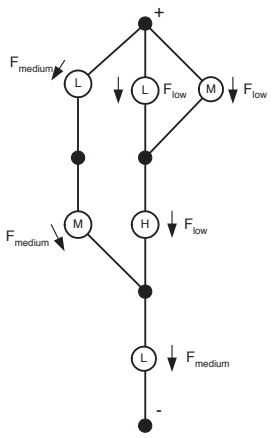


Figure 1: example MCIRQ network

Modelling circuit components

MCIRQ represents the structure of electrical components as resistances. A wire is a single resistance with value zero, a switch is represented as zero in its closed state and infinite in its open state. A lamp or other power-consuming element is represented by a non zero resistance dependent on its power consumption. In this work fluid circuit components are similarly represented; pipes provide a zero (or low) resistance, valves provide zero or infinite resistance for the open and closed states. A battery or power supply provides the positive and negative power terminals for an electrical circuit and pumps provide power terminals for fluid circuits. These terminals are mapped to the positive and negative network nodes.

The structure is changed by the state of valves and switches. Higher level component behaviour is represented by a model, such as a finite state machine, that is able to change state based on the results of the network analysis (Price *et al.* 1997; Snooke 1999). This in turn may trigger a structure change and new network simulation; the cycle is repeated until a steady state, cyclic behaviour, or ambiguity is detected.

Example system

Figure 2 shows a system using both electrical and fluid based components, and is to be used as a running example in the remainder of this paper. A CAD tool generates a netlist itemising each component and providing a connection list. The circuit represents two fuel storage tanks connected by a sequence of pipes, an electrically driven pump, and a mechanically operated bidirectional pump.

A component called atmosphere is also present in Figure 2 to model the flow of air between the tanks and completes the circuit. The atmosphere component model also allows leak faults to be simulated. Most faults are local to a component such as wire open circuit, or valve stuck open; however leak faults are analogous to electrical short to battery and short to ground faults, and require a change to global circuit structure. For specific modelling domains and tasks, abstract components such as atmosphere can be included in the translation from schematic to netlist to make the schematic drawing stage more intuitive. For clarity atmosphere is explicitly represented in this paper.

Multiple sources

Modelling multiple sources is approached using the principle of superposition. Superposition states that linear systems with multiple sources can be analysed by composing (i.e. summing) the results of separate analysis for each source. This is achieved in M^{2}CIRQ by executing MCIRQ for each pair of power nodes with all other power nodes removed and shorted with a zero resistance. Each arc in the network will have n current flow magnitude and direction values for a network with n sources. This of course may lead to qualitative ambiguity. For FMEA tasks this is not usually a problem; in fact it often provides useful information about the potential system behaviours. Indeed even a single source can generate ambiguous flow directions, for example in the case of a bridge circuit, although empirical evidence suggests that -even for electrical power systems- this occurs only when failures are being modelled; and then only rarely. Often such ambiguity has no long-term effect on the system behaviour and therefore does not cause simulation problems. In general, information about any ambiguity is passed to the analysis tool that initiated the simulation, where it is interpreted to provide information about the system level behaviour characteristics. The qualitative flow for the network is resolved for each arc as follows:

1. consider *only* the flows with the highest magnitude for each arc
2. if any flows are ambiguous the arc result is ambiguous
3. if all of the flows are in the same direction this is the result, otherwise there is a qualitative flow ambiguity for this arc

The previous paragraph applies to a *connected* network with multiple sources. A netlist such as the example system in figure 2 is easily partitioned automatically into two distinct networks: an electrical network containing the battery; and a fluid network containing two pumps. The simulator builds the system structure by creating and connecting all of

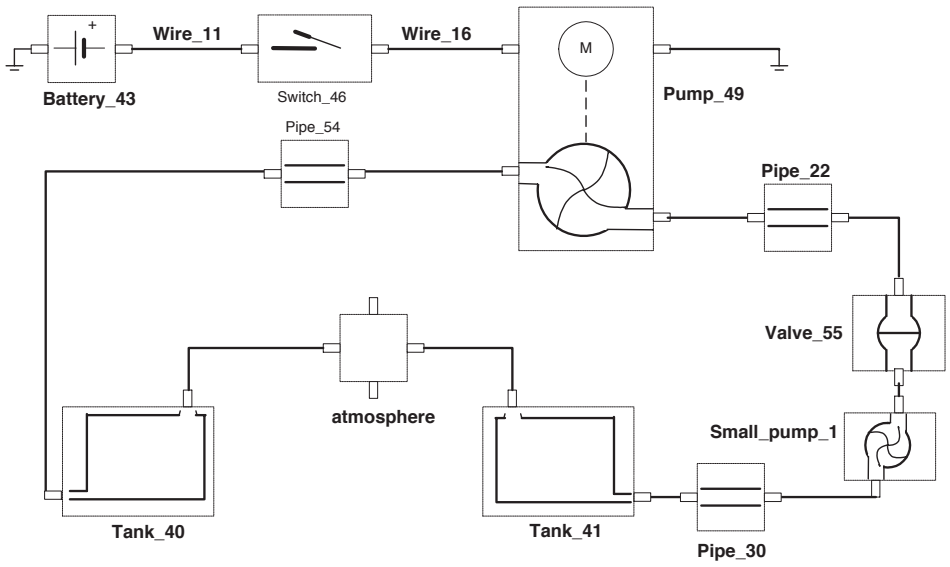


Figure 2: simple mixed circuit schematic

the component structure fragments. The network is then partitioned into disjoint subsets of nodes and arcs that represent separate networks. A component may of course belong to more than one network, for example the pump in the example. Notice that terminals may be typed to allow the drawing tool to do some checking to help prevent mistakes such as accidentally connecting elements from different domains. The simulator does not need to consider types, it simply uses the connection topology to partition networks. Indeed, several networks may be found for each domain, particularly for fluid systems that have several circuits flowing through common components. Of course if a component failure mode is being used that represents a connection between otherwise separate circuits then they will be connected, although the multiple source analysis will only produce a flow in the correct part of the circuit when the fault is not active. M^{2}CIRQ generates two networks for the example in figure 2 and produces the expanded component structure in figure 7.

Substance Propagation

It is desired to model the emptying and filling of tanks and the effects of leaks. The models therefore require that component state models can detect the substance flow present at inputs and outputs. Substance propagation between components naturally occurs at nodes, as the interface between components, and is dependent on flow direction. In the simple case of two connected components A and B, a substance is propagated into a node by the output component A, and received by component B input. In a complex circuit there may be several connections to a node and flow direction may change during simulation. For these reasons a list of substances is maintained at each node, and contains the output substances of each arc connected to it at any step of the simulation. Figure 3 depicts a node connected to three arcs.

Each node has the postfix symbol .SUBSTANCE available for assignment and evaluation by the component model (for example as part of a FSM event action). This is a consistent extension to the .RESISTANCE and .FLOW symbols available for the assignment of resistance and evaluation of flow already present for arcs.

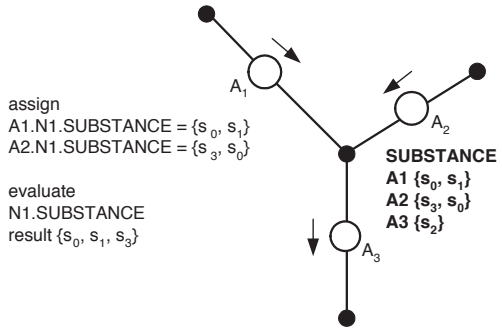


Figure 3: Node substance representation

Assignment to a node requires the arc associated with the substance and simply results in a list of substances being set for the required arc. Accessing the substance for a node within the model is straight forward as in the following example taken from the component description of a pipe.

```
IF resistance.FLOW == 'FORWARD'
  {resistance.T2.SUBSTANCE = T1.SUBSTANCE}
IF [resistance.FLOW == 'REVERSE'
  {resistance.T1.SUBSTANCE = T2.SUBSTANCE}
```

The simulator has work to do to evaluate the substance present at a node. It must consider the current flow directions for each of the connected arcs and produce a union of the substance lists associated with only the arcs that have flow directions into the node.

Modelling fluid components

The previous two sections have provided the capability to qualitatively model fluid flow systems. This section will detail the models for several components.

Using MCIRQ levels. The order of magnitude levels have been used with 5 levels to model normal flow, abnormal high flow (e.g. serious pipe fracture) and abnormal low flow (e.g. small leak). Zero flow occurs when blockage faults or valves are closed. Infinite flow or short circuit does not generally occur because even pipes are modelled as having some resistance.

Tanks

Tanks may be modelled with a number of states to represent the qualitative changes in the amount of substance contained. In the example only two states -empty and full- are provided. Qualitative order of magnitude time periods can be used in FSM based component behaviours and clearly these must be global to the entire system, since electrical and fluid system events may occur over the same time periods. The example uses an ordered set of time periods $\{\mathrm{instantaneous} < \mathrm{uS} < \mathrm{mS} < \mathrm{S} < \mathrm{Hour} < \mathrm{steadystate}\}$, although any sequence can be defined as long as they conform to the assumption that any number of events occur in one time period will take less time than an event in the next longest period. An infinite number of events in a single timeslot will generally signalling a modelling error. In practice this is unlikely to occur because most accidental infinite modelling loops will be terminated at the end of the first iteration by the simulator, which stops if a previously encountered state is reached. Mistakes with counter variables within models are the most likely culprit since the counter will prevent the system from reaching an identical state. Electrical and fluid systems do not generally require counter variables, which should be used with care when modelling software in electronic control units.

The time period used for the duration of the empty and fill events for the tank will determine its qualitative capacity, for example:

FLOW	Small tank	Medium tank	Big tank
low	Hour	Week	Month
medium	S	Hour	Week
high	mS	S	Hour

In this model a medium tank experiencing a medium flow would empty or fill in the hour order timeslot. Thus a low flow such as a small leak will cause the tank to empty in the order of week. This allows the prediction -if this tank were an aircraft fuel tank- that it would likely not imminently run out of fuel. A medium leak would cause a behavioural ambiguity indicating that the aircraft may run out of fuel and numerical information is needed. A major fracture indicates that the tank must be isolated if possible and no fuel transfers should be made into this tank (to balance the aircraft for example).

Tanks are usually designed for a specific range of substances and therefore it is reasonable to include these in the models. If an unrecognised substance is found flowing into a tank it can be made to enter an unknown behaviour state, reported to the higher-level analysis (it might be chemically attacked for example). For a fuel tank, fuel and air are the two substances the tank is expected to contain. The behaviour of the outlet, vent and contents are defined by the states of the tank. In the empty state air will flow out of the outlet, in the full state fuel will flow from the vent if flow in into the tank. Figure 4 shows this behaviour for the tank model used in the example and includes events related to flow magnitude, substance and direction.

Vented tanks have a fairly intuitive connection to the atmosphere that allows the flow of air between any number of tanks. A question arises if a tank is not vented such as for a pressure vessel. Closer inspection reveals that there remains a connection to the atmosphere, since a pressure differential will exist when as the tank is pressurised or evacuated, similar to the charging of an electrical capacitor. The connection to the atmosphere is maintained providing a logical circuit flow, although no substance is allowed to pass to the atmosphere. Once the vessel is pressurised it will become an infinite resistance to flow and may also then be modelled as a pressure source.

If the tank is in a situation where the orientation could change (for example in a aerobatic aircraft) then additional states may be required to model the movement of the fluid for example fuel entering the vent in certain orientations. If inversion is a feature of a system then all components with orientation dependent behaviour changes must be built to respond to an orientation change event, which is considered to be a system level event. All the events in the example model are component level.

Pipes

A generic pipe can be modelled with either zero resistance or a resistance value that represents the energy required to transfer the substance through the pipe. This energy may be required because the pipe bore is small or it may represent the energy required to overcome gravity if the pipe has a vertical element.

In the case where drain of substance by gravity is a feature of the system then gravity could be represented as a (small) pump(s) whose effect is overcome by normal system pumps, but causes flow when the pumps are inactive. Including several gravity pumps as part of a pipe model will possibly lead to flow ambiguity if they have opposing directions, for example in a siphon. In these cases system constraints that state relative pump sizes, may allow the ambiguity to be resolved, at least for simple topologies if not for the general case. Failing this numerical methods are required for the ambiguous region of system behaviour.

The pipe model will transfer substance from one terminal to the other dependent upon flow direction. If a pipe is considered to have small capacity this can be modled as an instantaneous event. For pipes with significant capacity or length a delay can be associated with this change of state and the pipe becomes similar to a small tank being filled with various substances. The pipe model used in the example is shown in figure 5. Event names are bold type and actions are contained in curly braces.

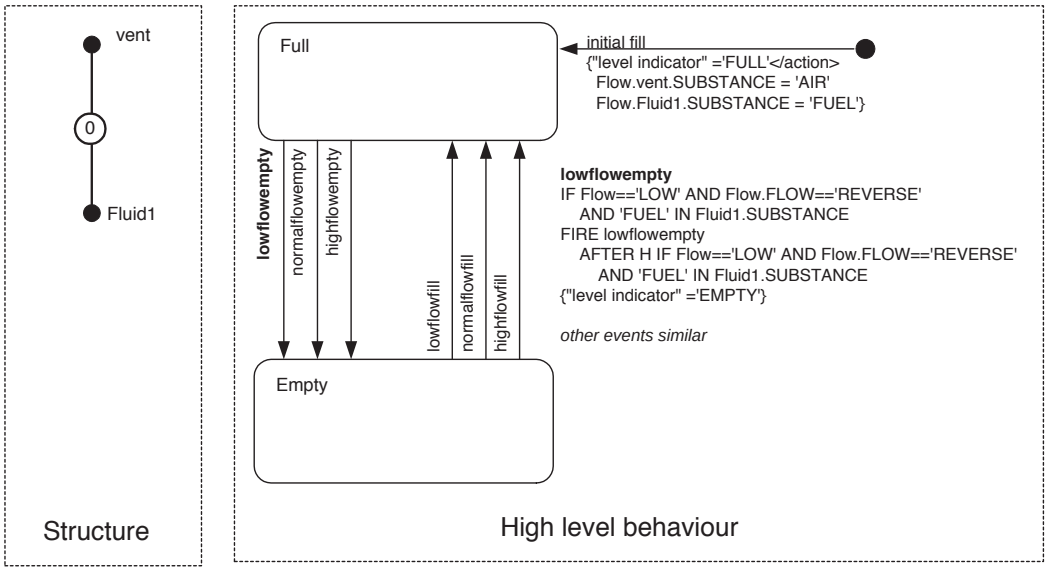


Figure 4: Tank model

Pumps

The pump acts as a transducer converting electrical or mechanical energy to a fluid pressure/flow. The pump is therefore a source for the fluid circuit and a load on the electrical one. The electrical pump becomes active when current flows through the motor causing a fluid source to be activated in the pump circuit.

For self priming pumps flow exists regardless of the substance at the terminals, non priming pumps only allow flow if the required substances are present at the input by presenting an infinite resistance in this condition.

Pumps may act as a blockage or a low resistance when not operating dependent on their design. A pair of source nodes are normally an open circuit requiring an additional resistance if the pump is to be free flowing when inactive. Pump_49 in figure 2 is an example of a free flowing inactive pump. Bidirectional pumps require the polarity of the source to be changed and can be modled as the SmallPump if figure 2, by four resistances operating in pairs based on the required pump direction. This approach saves forcing M^2CIRQ to analyse one circuit for each source polarity when clearly only one can ever be connected outside the pump component for any given simulation

The atmosphere

Leaks are the significant global fault for many fluid circuits. To simulate these situations the fault model of a component must make a connection to another component. The electrical analogy for a leak is a connection to the negative terminal of the source, but this is problematic because a leak may not only cause substance to escape, but may cause substance (e.g. air) to be drawn in on the negative pressure side of a pumped circuit. From the modelling perspective a leak should behave correctly for both situations. In addition the substance of the leak must be considered unlike the electrical case. An explicit model the atmosphere as a component provides a solution. A connection can then be made to the atmosphere for any leak faults. If the atmosphere is not otherwise used in the circuit, for example in a sealed hydraulic system, it is connected to the negative pump terminal(s) to provide the correct flow circuit.

The atmosphere model assumes an infinite capacity for inflow of substances and will always provide air if substance flows out of the atmosphere. The atmosphere component is essentially a connection point, although it is useful to provide a dedicated terminal to use to connect other components that have leak faults because air flowing out of these fault connections can be reported as air ingress into the system. The atmosphere can also be made to recognise and report substances other than air flowing into it since this may occur when vents overflow for example.

Example system

Simulation

Figure 6 demonstrates the simulator being driven manually. In this example a system level statechart[1] is being used to provide external inputs to the system. This deliberately causes the simulation to stop after S timeslot events because the system statechart has ambiguities (i.e there are a choice of switches that can be changed within a S time period). The user is asked to select from the ambiguous events. An example highlighted in figure 6 and displayed in the lower frame.

In this example the user selected to close the electrical switch, and then to open the valve. Fuel is transferred from one tank to the other in the Hour time period because the

[1] the system statechart is automatically generated from all component interface variables that are read (only) by each component model

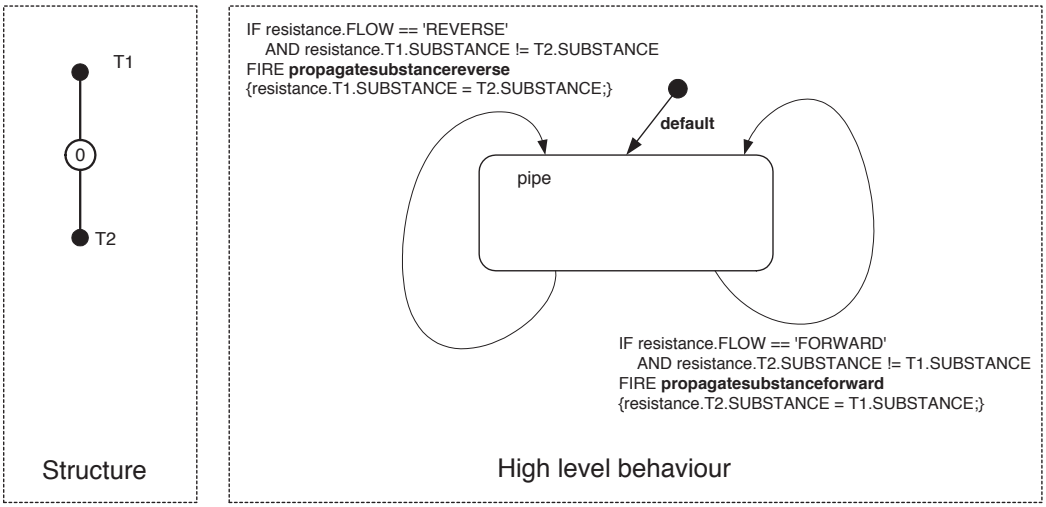


Figure 5: Pipe model

user did nothing in the highlighted S time period. The flow during this period is shown in figure 7 where the nodes are annotated with the fuel and air flow through the system.

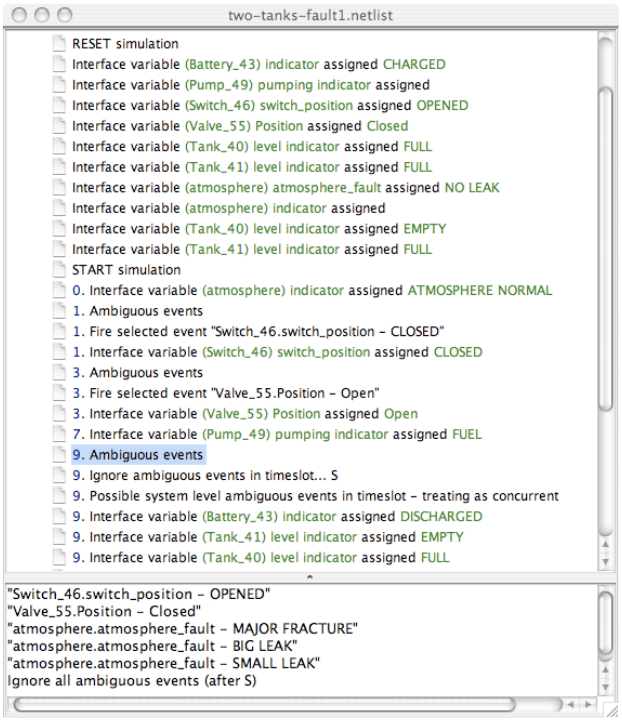


Figure 6: Simulation

Failure simulation examples (broken pipes)

Figure 8 allows the input of pipe_54 to be connected to the atmosphere to simulate a leak fault. Normally, a failure mode analysis tool inserts such faults programmatically; it is done manually here to allow the effect to be seen visually. The atmosphere can include three qualitative levels of fault and the leak in this case is caused by a Low resistance connection to the atmosphere. In addition the second pump has been activated in the same direction as the main pump, accordingly, the simulator shows twin flow arrows on the connections. The resulting flow is shown inside arc (hollow circle) symbols.

Figure 9 shows the situation where a small leak is created at the valve end of pipe_22. Air can be seen being drawn into the system by the pump, and propagated through the system. In colour reproduction the flow magnitude is indicated by the simulator, and in this case can be seen to be medium (green) for fuel, and fuel/air mix through the main circuit, with a low (cyan) flow of air between atmosphere and the small leak.

Conclusion and future work

The additions to the modelling do not affect the electrical circuit simulation or models and these are simulated as in earlier work, using the same models. These models have no substance included and only one voltage source is present and are therefore simply a subset of the new modelling and simulation capability.

Fluid and electrical circuit characteristics

The characteristics of electrical and fluid circuits are different. Fluid circuits often have multiple sources, however the topology of fluid circuits is often simpler than electrical circuits, in particular normal operating configurations of valves often reduce a specific operating model to a series circuit. This (by empirical observation) reduces the ambiguity of flow that might otherwise be expected, to a small number of failure cases. Further study of more complex systems will be carried out to verify these observations.

For electrical circuits five MCIRQ levels have been used to distinguish between signal and sensor inputs($<$1mA), and

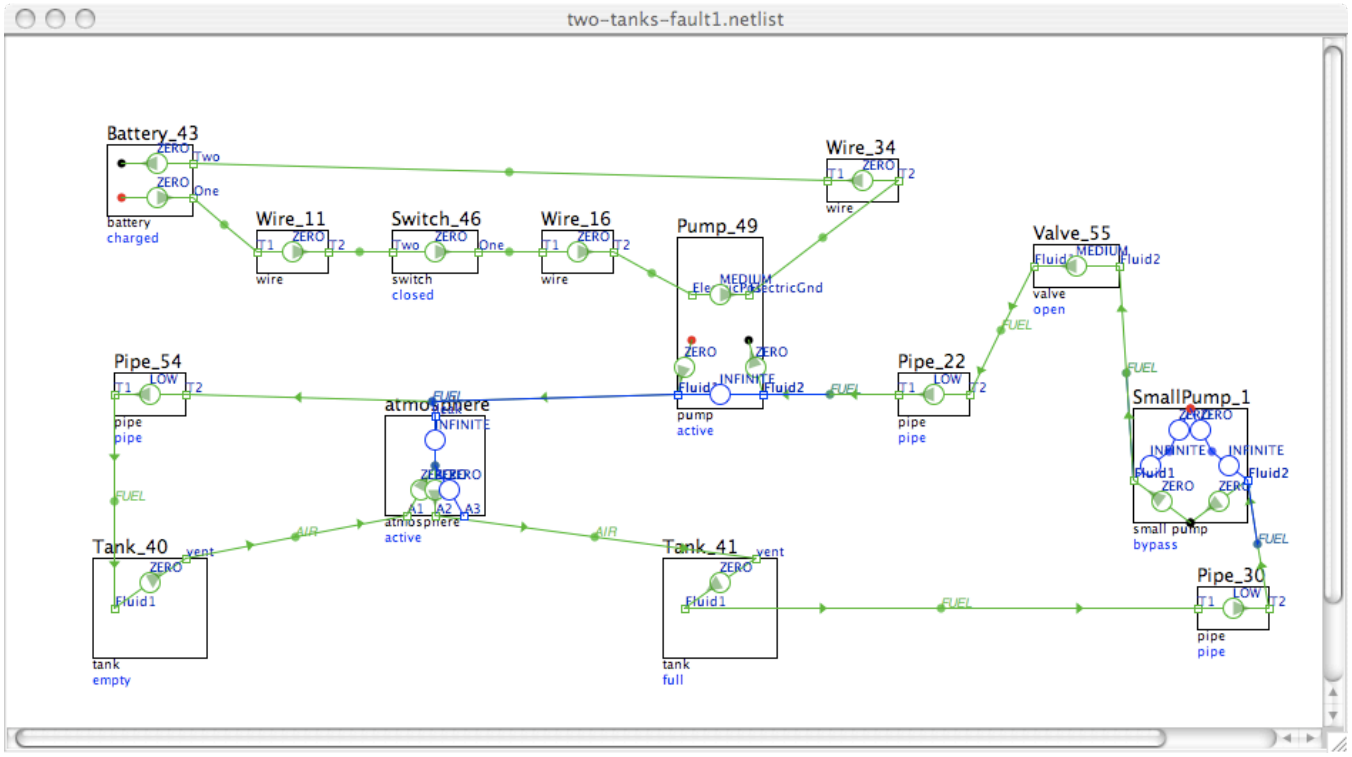


Figure 7: Simulation

low power devices such as relay coil inputs(10mA flows) and high power devices (Amps). This has allowed the analysis to ensure that a relay signal level flow would not light a car headlamp for example. In the fluid analysis it appears there is rarely a deliberate qualitative magnitude difference in flows within a system and a more useful distinction is within the failures that can exist. This is certainly the situation for aircraft fuel systems. Clearly a minor seepage leak and a major fracture have very different impacts on the behaviour of an aircraft fuel system, for example. These two distinctions again lead to a five level qualitative flow analysis. Zero, medium, and infinite resistance model normal operation. Low and high resistance are used in addition to represent faults.

Circuits with different voltage sources or batteries can easily be created and can be useful for example to model power transformers, or separate analogue and digital circuits, but faults connecting the circuits have not yet been investigated.

Substance representation within models

Some component models may be defined to operate with any substance, for example a generic pipe. Many components require behaviours that depend on the substance that is flowing, so that a tank does not fill if air is flowing into it. To some extent this limits the use of a component model to the types of system it was designed for, however this is realistic in many cases since most components will only operate as intended with the correct substance. It is useful if library models include 'behaviour out of specification' states to signal the limit of their behaviour has been reached if unknown substances are detected. It is perfectly possible to create a set of components and substances that are very generic such as liquid and gas for abstract modelling applications.

The representation of substances allows for the presence of more than one substance to flow through a connection. No modelling of mixing or separation processes are modelled with the exception that tanks always have air at the vent unless overflowing and this is adequate for the fuel transfer and hydraulic applications.

Non-linear components

The analysis of multiple sources relies on the linear resistance representation used by CIRQ. Non-return valves are the main fluid flow component that does not approximate to a linear flow component in a qualitative representation, and, unsurprisingly, have similar issues as encountered for electrical diodes. For many circuits these can be modelled as a state based component (zero/infinite resistance) with an extremely high impedance leakage resistor in parallel to detect voltage direction (voltage is not explicitly generated by CIRQ because there may be many levels required and they are not qualitatively significant across most components). This provides a requirement for a special qualitative resistance level directly below infinite, that results in a current flow that does not affect the system.

Conclusion

The network analyser enhancements have been implemented and allow variety of fuel system faults and features to be modelled. Faults include leaks and blockages, stuck and leaking valves, broken and faulty pumps, leaking tanks. System characteristics modelled include fuel flow and routing through multiple tank, multiple multi port valves. The emptying and filling of tanks and the qualitative time involved. The atmosphere is modelled allowing the egress and ingress of fuel and air from the system to be derived. An automated FMEA has not yet been generated using the simulator, but it should provide a similar level of system level results as the electrical only version. A variety of realistic fuel system models are being constructed.

Acknowledgements

This work was supported by the University of Wales Aberystwyth, the Welsh Assembly Government, BAE Systems and the DTI ASTRAEA Programme.

References

Chittaro, L., and Rannon, R. 1999. Automatic derivation of hierarchical representations for flow-based functional models. In Chantler, M., and Milne, R., eds., *Tenth International Workshop on the Principles of Diagnosis*, 45–50.

Kitamura, Y.; Ikeda, M.; and Mizoguchi, R. 1996. A qualitative reasoning based on an ontology of fluid systems and its evaluation. *Lecture Notes in Computer Science* 1114:288–.

Kuipers, B. J. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Massachusetts Institute of Technology.

Lee, M. H., and Ormsby, A. R. T. 1994. Qualitatively modelling the effects of electrical circuit faults. *Artificial Intelligence in Engineering* 8:293–300.

Lee, M. H. 1999. Qualitative circuit models in failure analysis reasoning. *Artificial Intellligence* 111:239–276.

Lee, M. H. 2000. Many-valued logic and qualitative modelling of electrical circuits. In *Proceedings 14th International Workshop on Qualitative Reasoning, (QR-2000)*.

Price, C. J.; Snooke, N.; Pugh, D.; Hunt, J. E.; and Wilson, M. S. 1997. Combining functional and structural reasoning for safety analysis of electrical designs. *The Knowledge Engineering Review* 12(3):271–287.

Price, C. J.; Snooke, N. A.; and Lewis, S. D. 2003. Adaptable modeling of electrical systems. In Salles, P., and Bredeweg, B., eds., *Proceedings of 17th International Workshop on Qualitative Reasoning (QR2003)*, 147–153.

Snooke, N. A. 1999. Simulating electrical devices with complex behaviour. *AI Communications* 12(1,2):45–58.

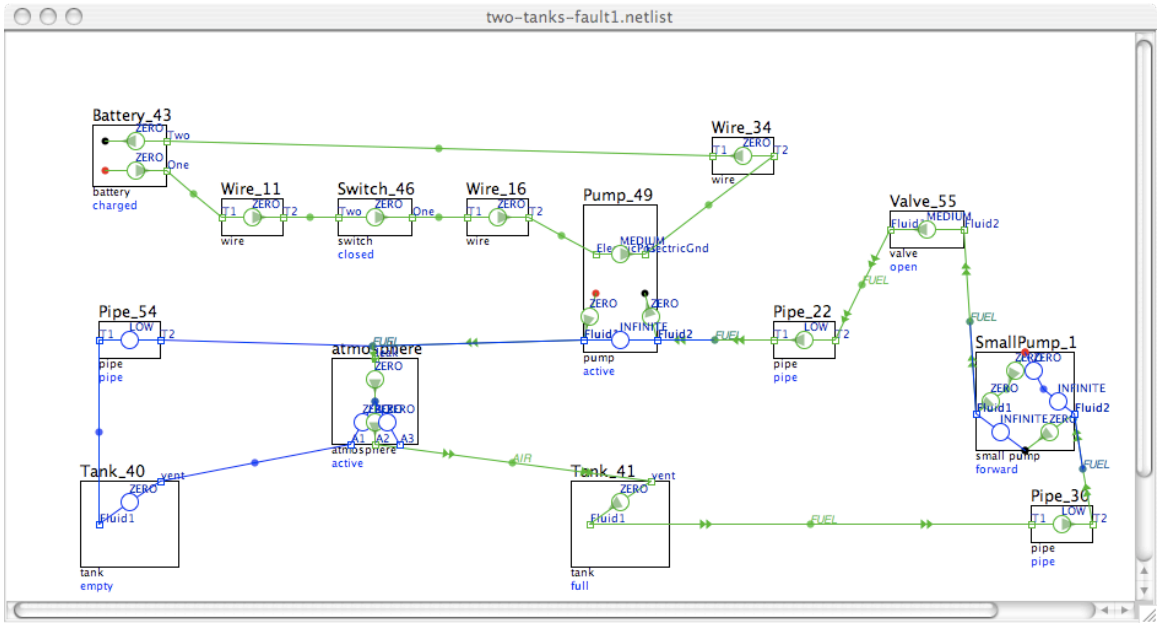


Figure 8: Major leak

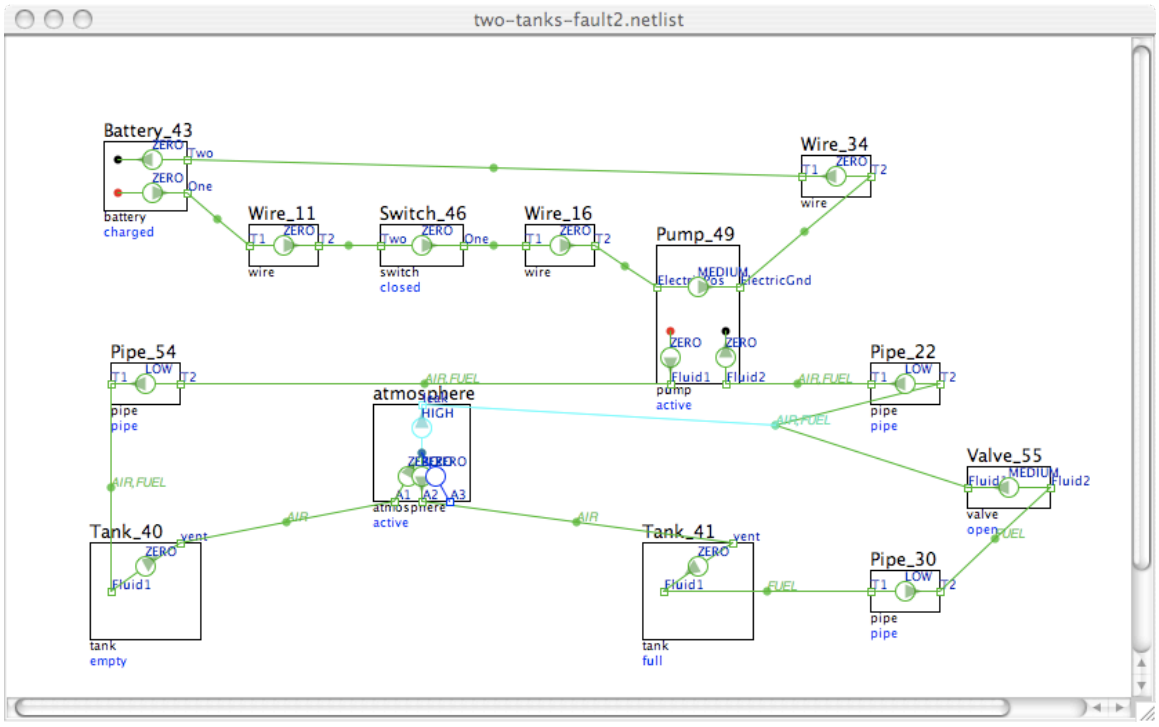


Figure 9: Small leak

PROVABLY SPURIOUS QUALITATIVE SIMULATION PREDICTIONS THAT JUST WON'T GO AWAY

Nuri Taşdemir and A. C. Cem Say
Boğaziçi University
Department of Computer Engineering
Bebek, 34342, İstanbul, Turkey
nuri.tasdemir@boun.edu.tr, say@boun.edu.tr

Abstract

It is known that sound and complete qualitative simulators do not exist; that is, there exist inputs which lead to ineradicable spurious behaviors, proving whose inconsistency is an undecidable task, and thus any sound qualitative simulator has to include them in its output. In this paper, we ask whether the next best thing, that is, a single sound qualitative simulator which detects and eliminates all *provably* inconsistent predictions, is possible, and obtain a negative answer. We prove that, for any sound qualitative simulator Q, which possesses two other reasonable properties that we define, there exists an input model which causes Q to predict a spurious prediction that can in fact be eliminated easily by many other qualitative simulators. Our result is a qualitative simulation version of Gödel's celebrated Incompleteness Theorem. We also show that, even when one restricts attention to models without self-reference, there exist infinitely many provably inconsistent inputs, which require so much time for a consistency check that such a simulator has to start printing out the spurious behaviors beginning with their initial states if it has a practical upper bound on its runtime.

1. Introduction

It is known [4] that sound and complete qualitative simulators do not exist; that is, there exist inputs which lead to *ineradicable* spurious behaviors, proving whose inconsistency is an undecidable task, and thus any sound qualitative simulator has to include them in its output. In this paper, we ask whether the next best thing, that is, a single sound qualitative simulator which detects and eliminates all *provably* inconsistent predictions, is possible, and obtain a negative answer. We prove that, for any sound qualitative simulator Q, which possesses two other reasonable properties that we define, there exists an input model which causes Q to predict a spurious prediction that can in fact be eliminated easily by many other qualitative simulators. Our result is a qualitative simulation version of Gödel's celebrated Incompleteness Theorem. We also show that, even when one restricts attention to models without self-reference, there exist infinitely many provably inconsistent inputs, which require so much time for a consistency check that such a simulator has to start printing out the spurious behaviors beginning with their initial states if it has a practical upper bound on its runtime.

2. Background

In the following, we make use of the terminology of QSIM [2], which is a state-of-the-art qualitative simulation methodology, although it should be noted that the results that we will be proving are valid for all reasoners whose input-output vocabularies are rich enough to support the representational techniques that will be used in our proofs.

This section starts by clarifying some of the additional terminology to be used in the rest of the paper. We then list a number of previously proven facts that will be utilized in our arguments.

2.1 Terminology

Qualitative simulator input: Qualitative simulators take a system model and a description of the initial system state as input. The model consists of one or more operating region descriptions and definitions of possible transitions between operating regions. Each operating region description contains variable-related definitions such as quantity spaces and legal ranges, and constraints that hold between the variables in that region. In this paper, the initial state description is always assumed to contain a complete assignment of qualitative values to all the variables of the initial operating region. When some control switches and parameters of the simulation need to be set to values other than their defaults (e.g. when the user wants QSIM to create no new landmarks for some variables during simulation,) the description of these settings is also part of the input. In the following discussion, the term *qualitative simulator input* denotes a single string encoding all the information mentioned above.

Soundness: A qualitative simulator is *sound* if it is guaranteed that, for any ODE and initial state that matches the simulator's input, there will be a behavior in its output which matches the ODE's solution. QSIM, for instance, is known to have the soundness property [2].

Completeness: A *complete* qualitative simulator would come with a guarantee that every behavior in its output corresponds to the solution of at least one ODE matching its input.

The output of a sound and complete qualitative simulator, if such a thing could exist, would thus contain a tree of qualitative states rooted at the initial state, such that all paths starting from the root and ending at a leaf (for finite branches) or containing an infinite sequence of states correspond to a solution of an ODE matching the input, and all such solutions would match such a path.

Consistent input: An input is *consistent* if and only if it could cause the prediction of at least one behavior on a hypothetical sound and complete qualitative simulator.

Note that good qualitative simulators are supposed to produce an empty tree in response to an inconsistent input.

We now define two more desirable properties for qualitative simulators, indeed, for almost any program.

Steadfastness: A *steadfast* qualitative simulator is one which does not retract any part of its output that it has already printed. In particular, once a steadfast qualitative simulator has printed the root of the behavior tree, corresponding to the initial system state, its output is guaranteed to contain at least one behavior prediction starting from that state.

The motivation behind our explicit definition of this very reasonable and easily realizable property is the interesting fact that implementations of QSIM which start printing out the behavior tree before the simulation is over, (this is inevitable for inputs that cause trees which are either infinite, or finite but so big that running the simulation to completion is not an option,) are not steadfast; since inconsistency can propagate backward from the leaves to the root, QSIM may decide to prune a branch of the behavior tree after adding arbitrarily many states to it [2]. This is a result of the rule which states that all states, except the quiescent states and the transition states (which satisfy the operating region transition or termination conditions), should have at least one consistent successor in order to be consistent. A state which has no consistent successor state is also inconsistent even if it passes all other filters. So a state which has been added to the behavior tree may be labeled much later as inconsistent, if all of its successor states have been labeled inconsistent. Therefore, there are inputs which QSIM may announce as inconsistent only after building and then destroying a large tree rooted at them. If the simulator does not keep such a tree in memory, but instead starts to print it out before the end of the simulation, the later announcement that the input was, after all, inconsistent constitutes a violation of steadfastness as defined above.

Responsiveness: A *responsive* qualitative simulator starts printing a nonempty output within a finite amount of time after it starts running.

Note that a responsive qualitative simulator should produce an output even if the input is inconsistent. In such a case, the simulator should print a statement to the effect that the simulation result is an empty tree.

A responsive and steadfast qualitative simulator announces its final verdict about the input (i.e. either reports an inconsistency or prints the initial state as the root of the behavior tree, meaning that it has deemed the input consistent) in finite time. In the discussion below, we refer to this announcement as the *response* of the qualitative simulator to its input.

2.2 Facts

2.2.1 Exact Representation of Integers in Qualitative Simulator Inputs

For any integer *z*, there exists a set of QSIM variable quantity spaces and constraints, from which *z*'s equality to a particular variable in that set can be unambiguously deduced [6]. This can be achieved easily by encoding the required value with addition and multiplication constraints. For example, if we want to express that a variable has value 5, then we can use following structure where all the variables are defined to be constant and ONE is initialized to a positive finite value:

ONE = ONE × ONE
TWO = ONE + ONE
THREE = TWO + ONE
FOUR = THREE + ONE
FIVE = FOUR + ONE

Here, it is obvious that ONE equals 1, and so FIVE is 5.

2.2.2 Computationally Universal Qualitative Simulators Exist

The unlimited register machine (URM), which is equivalent in power to the Turing machine (TM) model, is one of the many mathematical idealizations of computers [1]. A URM has finitely many registers which can store nonnegative integers. There is no upper limit for the value contained in a register. Every URM has a program which contains an ordered list of instructions (Table 1) to be performed on the registers. When an instruction (other than a *jump*) is executed, the next instruction to be executed is the one right after the current one. Table 2 contains the description of a simple URM, which gets two integers as input in registers 1 and 2, and gives the sum of these numbers as its output in register 1.

URM Instructions	
$inc(r_j)$	increments the value in register j
$zero(r_j)$	resets the value in register j to zero
$jump(r_j, r_k, i_m)$	If j is equal to k, jumps to instruction m, otherwise, the next instruction is executed
end	terminates the computation

Table 1: URM Instructions

i_1: zero(r_3)
i_2: jump(r_2,r_3,i_6)
i_3: inc(r_1)
i_4: inc(r_3)
i_5: jump(r_2,r_2,i_2)
i_6: end

Table 2: URM Program Computing $f(x, y) = x + y$

Yılmaz and Say proved [10] that any given URM/input pair can be simulated in a qualitative simulator which supports one of several quite restricted subsets of the input/output vocabulary of QSIM. To simulate a URM program with p instructions, one constructs a qualitative simulator input with $p+2$ operating regions: one for each instruction, one for the initialization, and one more for the finalization of the computation. In the qualitative simulation of the URM's computation process, each state of the behavior tree (except the root, which corresponds to the initialization,) corresponds to the execution of an instruction. This simulation can be performed in a behavior tree with a single branch. (Note that this requires some additional filters which "decode" the input to obtain and then keep track of the exact numerical values of the simulation variables to be incorporated to presently available qualitative simulators, and nobody has seriously tried to implement the construction in [10] to our knowledge. However, an

implementation is entirely possible, and in fact quite straightforward when compared with some of the mathematically much more sophisticated filters that have been developed for QSIM, e.g. [5, 7].)

In fact, such qualitative simulators can be thought of as an alternative computational model like the URM, and appropriately prepared qualitative simulator inputs play the roles of the programs to run on this computational model.

Note that qualitative simulators can be (and are) simulated by our computers; therefore they can be simulated in a TM, which is capable of doing everything which can be done by our computers [8]. It follows from the computational universality of URM's that any qualitative simulator can be simulated by a URM. As a result, a qualitative simulator which supports one of the subsets of the QSIM input vocabulary listed in [10] can simulate any other qualitative simulator implementation.

2.2.3 The Recursion Theorem

This theorem, [8] which is a well-known fact of computability theory, provides the following technique, which can be used when one needs to construct programs which can store their own code in a variable, and then process it as necessary: We construct a program which consists of three parts; *A*, *B*, and *Main*, which run in this order. When executed, part *A* stores the code of the other two parts, namely, a string of the form <*B*,*Main*>, into a variable *v*. Part *B* then starts running, and uses the string in *v* to construct the description of a partial program which stores the value that *B* sees in *v* into the variable *v*. Note that the partial program *B* prepares in this manner is *A* itself. *B* then appends <*A*>, which it has just constructed, with the current contents of *v*, stores the resulting longer string, which is none other than the code of our program itself, namely <*A*,*B*,*Main*>, in *v*, and passes control to part *Main*, which can use the program's code stored in *v* when needed. *Main* contains the rest of the code which makes the program accomplish whatever its designated task is; *A* and *B* are used just for implementing the recursion technique described above.

2.2.4 The Halting Problem Is Reducible to Hilbert's Tenth Problem

As the name suggests, Hilbert's Tenth Problem is the tenth of 23 problems which were announced in 1900 by the famous mathematician David Hilbert as a challenge to the mathematicians of the 20th century. It asks for an algorithm for deciding whether a given multivariate polynomial with integer coefficients has integer solutions. In 1970, Yuri V. Matiyasevich showed that no such algorithm exists, by demonstrating a method which can be used to construct, for any given Turing machine *T*, a polynomial *P* with integer coefficients, such that *P* has a solution in the natural numbers if and only if *T* halts on the empty input. As mentioned above, the original statement of the problem talks about the domain of integers, rather than natural numbers. However, this can be shown to be equivalent in difficulty to the version with the domain restricted to the natural numbers; see, for instance, [3].

2.2.5 Hilbert's Tenth Problem Is Reducible to Qualitative Simulator Input Consistency Checking

Yılmaz and Say have proven [10] that, even if the qualitative representation is narrowed so that only the *derivative*, *add, mult* and *constant* constraints can be used in

QDE's, and the simulation proceeds only in a single operating region, it is still impossible to build a sound and complete qualitative simulator based on this input-output vocabulary. This proof uses a reduction from Hilbert's Tenth Problem, namely, a technique that can be used to build, for any given multivariate polynomial *P* with integer coefficients, a qualitative simulator input *QI*, such that *QI* is consistent if and only if *P* has a solution in the integers. This means that a sound and complete qualitative simulator, if it existed, could be used to solve Hilbert's Tenth Problem. Although this proves that there can be no qualitative simulator which is both sound and complete, the transformation used for this purpose in [10] can also be used fruitfully to obtaining interesting results about sound, steadfast, responsive and naturally incomplete simulators, as will be seen in Section 3.2.

3. Every Sound, Steadfast and Responsive Qualitative Simulator Has a "Blind Spot"

We will now prove that every qualitative simulator which possesses the soundness, steadfastness, and responsiveness properties necessarily predicts a provably spurious behavior *B*, and that this same *B* can be recognized as spurious and filtered out easily by many other feasibly constructible qualitative simulators. Section 3.1 demonstrates this fact for qualitative simulators which support the operating region transition feature. In Section 3.2, we show that this feature is not required for the phenomenon we describe here to occur.

3.1.The Blind Spot Theorem: Multi-Region Version

We start by observing that qualitative simulator inputs can be designed to use a simple adaptation of the recursion technique of Section 2.2.3 to obtain and store their own code in a simulation variable. Such an input will consist of three submodels: *A*, *B*, *Main*. *A*, which consists of a single operating region, will contain a variable *V*, which it initializes to an integer encoding the string <*B*, *Main*>. Another variable in *A* is constrained to reach a landmark which will trigger a transition to the starting operating region of the multiple-region submodel *B*. Variable *V* inherits its value during all operating region transitions. *B* models a URM which uses its knowledge of the value in *V* to prepare the description of a qualitative input submodel, which models a URM that initializes variable *V* to the value *B* now sees in *V*, and then triggers a transition to the starting operating region of *B*. Note that this submodel description prepared by *B* is none other than <*A*>. *B* then combines <*A*> and <*B*, *Main*> to obtain <*A*, *B*, *Main*>, stores this value in *V*, and triggers a transition to *Main*, where the description of the entire input <*A*, *B*, *Main*> can be used as needed.

We now note that, given any qualitative simulator *C*, one can build a qualitative simulator input M_C as follows:

M_C contains the representation of a URM program. Upon starting execution, M_C first acquires its own code $<M_C>$ using the recursion technique described above, and then starts to simulate *C*, whose code has been embedded in that of the program of M_C, with $<M_C>$ as input. The simulation of *C* is performed until *C* gives its response about the

input, i.e. until C either declares inconsistency or prints the initial state as the root of the behavior tree. If C rejects the initial state of $<M_C>$, the program of M_C ends by arriving at an operating region where a variable increases until it reaches a bound of its legal range, constituting a successful termination of the corresponding branch of the behavior tree, meaning that $<M_C>$ was a consistent input. On the other hand, if C prints the initial state of its input, the program of M_C jumps to an instruction represented by the operating region OR_C, which causes a contradiction. This can be achieved by a variable, say, S, which is defined in all operating regions, and whose value is inherited in all operating region transitions. S is constrained to be constant in all operating regions and it is initialized to a positive finite value. In the operating region OR_C, S is constrained to be constant at zero. Therefore a transition into this region causes an inconsistent behavior.

Now let Q be any sound, steadfast and responsive qualitative simulator. We claim that the input M_Q is inconsistent, and yet Q does not reject this input; it starts printing a provably spurious prediction that begins with the initial state of M_Q. We justify this claim with the following analysis of the execution of Q on input M_Q:

To prevent confusion, let Q_0 denote the "outer" Q, and let Q_1 denote the "inner" Q, which will be simulated as described above by the program M_Q. Since Q_0 and Q_1 are implementations of the same qualitative simulator which are working on the same input (M_Q), their actions will be exactly the same.

There are two possibilities for the response of Q to the input M_Q: Q either rejects M_Q, or prints out the initial state of M_Q.

Let us first analyze the case where Q_0 rejects M_Q. Then Q_1 will also reject M_Q. But now consider what the program described by M_Q does: It simulates Q_1 for a finite number of steps to see how Q_1 responds to $<M_Q>$, and when it sees a rejection, it terminates successfully, without reaching a contradictory state. This is a perfectly valid behavior of the described system, and should of course be printed out by any sound qualitative simulator. Since Q is sound, we conclude from this argument that it cannot reject M_Q.

The remaining possibility is that both Q_0 and Q_1 will print out the initial state of M_Q. Since Q is steadfast, printing the initial state is an irreversible action, and means that Q announces the input M_Q to be consistent. Let us consider what the program of M_Q does in this case: It simulates Q_1 for a finite number of steps, and when it sees Q_1 print out the initial state of M_Q, it jumps to a contradictory operating region, making the branch of the behavior tree describing its entire execution a spurious one. Since the model is so constrained that no other nonspurious branches are possible, as explained in section 2.2.2, we conclude that M_Q is, after all, inconsistent. By the argument of the previous paragraph, Q must announce this inconsistent input to be consistent.

There exist other qualitative simulators which can correctly detect the inconsistency of this input and reject it: Consider, for example, a computationally universal version of QSIM, to which the "numerical" filters mentioned in section 2.2.2, that are required for the simulation of a URM to produce a single-branch behavior tree, have been incorporated. Such a simulator will start "running" the program of the input M_Q, which in turn will simulate Q on the input M_Q, see Q accept M_Q as proven above, and jump to the contradictory operating region, at which point the "outer" simulator will propagate the inconsistency all the way back to the initial state and reject the input M_Q. Interestingly, almost every sufficiently sophisticated qualitative simulator other than Q is capable of

rejecting M_Q in this manner. It is this fact which leads us to use the term "blind spot" in the title of this section.

As a somewhat frustrating thought exercise, one can show that some qualitative simulators can sometimes "understand" that their present input will cause a blind spot spurious prediction, but they just cannot announce it loud, so to speak: Assume that our sound, responsive and steadfast simulator *Q* has been written by someone who knows about the trick that we have been discussing above. The programmer has coded *Q* so that it obtains its own code using the recursion technique, and then uses this to construct the string $<M_Q>$, which will cause it so much trouble. *Q* can now compare its present input with $<M_Q>$, but even this capability does not save it: Even when *Q* "knows" that the input is $<M_Q>$, it cannot announce it to be inconsistent as proven above, and the only option available is to start print the spurious prediction.

The argument we use to prove the existence of blind spots in sound, steadfast and responsive qualitative simulators has been inspired by the proof of Gödel's incompleteness theorem, which states that a sound formal system of axioms and rules of inference cannot be complete if it satisfies some simple conditions. (A *sound* formal system is one in which one cannot prove a statement to be both true and false at the same time. In a *complete* formal system, every true statement is provable.) The key point of Gödel's proof is the sentence *T*="This sentence cannot be proven," which can be defined mathematically in any system *M* which satisfies the conditions. If this sentence *T* can be proven, then it is obvious that a false statement is provable; since *T* states that *T* itself cannot be proven. Since system *M* is sound, this is not a valid choice. The other possibility is the non-existence of a proof in system *M* for statement *T*. This means that *T* is true, and therefore *M* is incomplete. For more information on Gödel's proof, see [9]. The resemblance between the argument in this section and Gödel's proof is a result of self reference. In our proof, the input has to be consistent if the qualitative simulator rejects it, and in Gödel's proof, the statement has to be wrong if the system can be used to prove it.

3.2.The Blind Spot Theorem: Single-Region Version

In the proof of Section 3.1, the operating region transition feature of the QSIM vocabulary played a critical role, since it is due to that representational item that URM's can be modeled. In this section, we show that the problem demonstrated in that section persists even when the operating region transition feature is excluded from the qualitative simulation vocabulary.

Let *Q* be any sound, steadfast and responsive qualitative simulator which works with the restricted vocabulary described above. We will now demonstrate that there exists an input which is inconsistent but which is announced to be consistent by *Q*. For this purpose, we first construct a Turing machine named *T*.

T starts to simulate *Q*'s simulation of an input *M*, whose preparation will be described shortly. On the first response of *Q*, *T* stops the simulation. If *Q* rejects its input, *T* halts; otherwise, *T* loops forever.

T prepares the input *M*, which it feeds to *U*, as follows: *T* first obtains its own code $<T>$ by recursion, and then it sets up a polynomial D, which has a solution if and only if *T* halts with the empty string as input, (The details of this computation are explained in the next two paragraphs.) *T* then encodes D as a qualitative simulator input *M* using the

technique described in [10], (Section 2.2.5) setting the initial magnitudes of all the simulation variables representing the polynomial variables to (0, ∞).

The transformation used by *T* to encode its own halting status in a multivariate polynomial is realized in two stages. T first employs the techniques of [3] (Section 2.2.4) to produce a polynomial D_1 defined on the domain $\mathcal{N}$ (including zero). Since we will specify to the simulator that we are looking for a solution where all the polynomial variables are positive, as mentioned in the previous paragraph, what we really want here is a polynomial which has positive roots if *T* halts. So *T* transforms D_1 to another polynomial D in the following manner:

Given a polynomial $D_1(x_1, x_2 \cdots x_n)$, which is defined on $\mathcal{N}$, we will build a new polynomial D, which has a solution in the positive integers, if and only if $D_1(x_1, x_2 \cdots x_n)$ has a solution in the set of natural numbers. D is the product of all the variations of D_1. By a *variation* of D_1, we mean a polynomial which can be obtained by setting some of the variables of D_1 directly to zero. Since there are two possibilities (zero or not) for each variable, there are 2^n variations of D_1.

$$D = \prod_{i \in \{0 \cup 1\}^n} D_i, \text{ where } D_i = D_1(i_1 \times x_1, i_2 \times x_2 \ldots i_n \times x_n)$$

where i_j is the j^{th} character of the string i, i.e. 0 or 1.

Having described *T*, we immediately proceed to our proof. *Q* can either find the input *M* inconsistent, or can start to print out the initial state. Let us analyze these cases.

Assume that *Q* says that *M* is inconsistent, and that therefore *T* halts. But if *T* halts, then D has a solution, and *M* is consistent. So *Q* has incorrectly rejected a consistent behavior. Since *Q* is sound, this is impossible, so *Q* cannot reject *M*.

So *Q* accepts *M*. But then *T* is a TM that does not halt, meaning that D has no solution, and that *M* is inconsistent. So *Q* accepts an inconsistent model.

4. States Checkable with High Cost

In [10], it is shown that a qualitative simulator can simulate a URM. Now, we will use this fact to construct a qualitative simulator input whose consistency requires $\Theta(2^{n^k})$ time to be detected, where n is the size of the input and $k > 0$.

Consider any EXPTIME-complete language *A*. The fastest algorithm which decides *A* has superpolynomial time complexity, since all other languages in the class EXPTIME can be reduced to *A* in polynomial time, and it is known that $P \subset EXPTIME$ [8]. Since *A* is decidable, a URM which decides it exists, call this URM *U*.

We will use a modified version of the technique in [10], to encode *U* and its input as an input for a qualitative simulator *Q*. The only difference from [10] will be the number of finalization operating regions. We need two separate finalization operating regions. Since *Q* will simulate a decider, one of the regions will stand for *yes*, while the other will stand for *no*. The *no* operating region contains an inconsistency with regard to the other operating regions, so if the simulation reaches the *no* operating region, this will result in a spurious behavior, and since there will be at most one simulation branch, the input will be inconsistent in this case. The *yes* operating region does not contain an inconsistency, and

therefore in the case of reaching there, the simulation will output a single nonempty behavior successfully.

Now, let us construct a Turing machine *T* for deciding *A*. *T* reads its input string *x*, and uses the technique described in the previous paragraph to construct a qualitative simulator input *M*, which encodes the URM *U* working on input *x*, and then simulates *Q* on input *M* until *Q* gives its response to the initial state. If *Q* prints the initial state, this means that $x \in A$, and *T* prints *yes*; if *Q* rejects the input *M* due to inconsistency, this means $x \notin A$, and *T* prints *no*. (Note that this construction is guaranteed to be valid only if *Q* is steadfast.)

Let us calculate how fast the fastest possible qualitative simulator *Q* can respond to *M* in this scenario. Let the length of *x* be *n*. The length of the input *M* of the qualitative simulator is $\Theta(n)$, since the only operating region of *M* whose size depends on *n* is the starting region, where the value *x* is supposed to be encoded as the initial value of *U*'s first register, and this can be done using a set of constraints that can be expressed in $O(n)$ symbols. All the other operating regions have fixed lengths that do not depend on *x*. So $|M|$ is $\Theta(n)$. Now, we know that for some values of *x*, the fastest possible *T* will have to run for $\Theta(2^{n^k})$ steps. If one leaves the simulation of Q aside, it is clear that the remaining parts of T have a total runtime of $O(n)$. Since the total time is $\Theta(2^{n^k})$, this concludes that time required for Q should also be $\Theta(2^{n^k})$. Since $|M|$ is $\Theta(n)$, and n is $\Theta(|M|)$, Q is seen to require $\Theta(2^{\Theta(|M|)^k})$ steps, that is, an exponential amount of time in terms of the size of its own input, to decide about the consistency of its initial state.

If one thinks about QSIM (a version which has been augmented with the numerical filters to ensure a single branch while simulating the URM, and which has been guaranteed to act steadfastly, at least for the inputs it will encounter in this construction, by making sure that it starts printing the constructed state tree only when the simulation is over,) in this scenario, it is clear that the announcement of the verdict about the initial state will take $\Theta(2^{\Theta(|M|)^k})$ steps, since QSIM would construct the branch all the way to its end, and then, in case of a *no* answer, propagate the inconsistency all the way back to the initial state. The proof above shows that this runtime is the best that can be achieved by any qualitative simulator.

We conclude that, for any sound, responsive and steadfast qualitative simulator which has a practical (i.e. polynomial) upper bound on its runtime, there exist infinitely many provably inconsistent inputs, which require so much time for a consistency check that the simulator has to start printing out the spurious behaviors beginning with their initial states.

There exists an infinite hierarchy of languages which require worse and worse runtimes than those in EXPTIME [8]. All these can be used to demonstrate the existence of spurious behaviors which are eradicable in principle, but ineradicable in practice, by the same argument as above.

5. Conclusion

We proved that there is no single sound, responsive and steadfast qualitative simulator which can detect and eliminate all eradicable spurious predictions. Furthermore, when practical limits are imposed on the runtime, the set of spurious predictions that can be eliminated is a dramatically small subset of the set of all eradicable spurious predictions.

We acknowledge that the models involved in our arguments are not of the kind that would normally be submitted to a qualitative simulator by a sensible user. But getting rid of the occasionally predicted eradicable spurious behavior is a desirable thing for those normal users as well, and we hope that the findings reported here might be useful for researchers interested in constructing qualitative simulators with improved theoretical guarantees and additional filters of increasing mathematical sophistication.

References

[1] Nigel J. Cutland, 1980. Computability: An Introduction to Recursive Function Theory. Cambridge University Press.

[2] Benjamin J. Kuipers, 1994. Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge, MIT Press, Cambridge, MA.

[3] Yuri V. Matiyasevich, 1993. Hilbert's Tenth Problem. Cambridge, Mass.: The MIT Press.

[4] A. C. Cem Say and H. Levent Akın, 2003. Sound and complete qualitative simulation is impossible, Artificial Intelligence Vol. 149, pp. 251-266.

[5] A. C. Cem Say, 1998. L'Hôpital's filter for QSIM, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 20, pp. 1-8.

[6] A. C. Cem Say, 1997. Numbers representable in pure QSIM, in: Proc. Eleventh International Workshop on Qualitative Reasoning, Cortona, Italy, pp. 337–344.

[7] A. C. Cem Say, Selahattin Kuru, 1993. Improved filtering for the QSIM algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 15, pp. 967-971.

[8] Michael Sipser, 1997. Introduction to the Theory of Computation, PWS Publishing Company.

[9] Raymond M. Smullyan, 1992. Gödel's Incompleteness Theorems. Oxford University Press.

[10] Özgür Yılmaz and A. C. Cem Say, 2006. Causes of ineradicable spurious predictions in qualitative simulation, Journal of Artificial Intelligence Research Vol.27, pp. 551-575.

Plenty of Blame to Go Around:
A Qualitative Approach to Attribution of Moral Responsibility

Emmett Tomai **Ken Forbus**

Qualitative Reasoning Group
Northwestern University
2133 Sheridan Rd, Evanston, IL 60208
etomai@northwestern.edu

Abstract

We present a computational model of blame attribution. Recently Mao and Gratch, following Attribution theory, created a computational model that assigned blame to an agent for a negative occurrence. Their model made categorical judgments, and could only assign blame to a single agent. Our model extends this work, using QP theory to provide a continuous model for the parameters involved in attribution and directly capturing the constraints postulated by Attribution theory. This allows our model to infer relative amounts of blame in a situation in a manner that is consistently overall with relative amounts of blame attributed in a psychological experiment.

Who is to Blame?

Bad things happen, and an important capability of social agents is to understand who is responsible. From the affairs of nations to personal misfortunes, accountability is an important part of how we understand the world around us. But how does a person go from perceiving such a situation to making a judgment of blame? This question has been the topic of much research in social psychology. Driven by the need to create social agents that can interact with people for a variety of purposes (tutoring, entertainment, assistants), creating computational models to capture such judgments is receiving increased attention. Without an understanding of blame assignment, an agent cannot properly infer the implications of social interactions.

This paper describes how Qualitative Process theory [Forbus 1984] can be used in such modeling. We briefly summarize the elements of *Attribution theory* that address blame judgments. We then discuss the Mao and Gratch computational model [Mao & Gratch 2005][Mao 2006]. We present an alternative model for attribution of blame based on QP theory, which we claim better represents the underlying theory as well as human data. We present an evaluation of our model using data collected by Mao, showing that our model captures that data better, and makes additional predictions.

Attribution Theory

Attribution theory is an area of research in Social Psychology based on the founding work of Heider [Heider 1958] and advanced by Kelley [Kelley 1973] and Jones [Jones & Davis 1965]. Its goal is to identify the conditions that will lead a perceiver, through an *attribution process*, to attribute some behavior, event or outcome to an internal disposition of the agent involved, as opposed to an environmental condition. Attribution is, therefore, a judgment embedded in the point of view of the perceiver and subject to the epistemic state of that perceiver.

Further work in Attribution theory has directly addressed the question of the attribution of blame [Shaver 1985, Weiner 1995]. Our model is based primarily on the work of Shaver who makes the distinction between *cause*, *responsibility* and *blameworthiness*. For a given negative outcome, cause is defined as being an insufficient but necessary part of a condition which is itself unnecessary but sufficient for that result. Only causes which represent human agency are of interest to the theory. It is certainly possible that people attribute responsibility and even blame to animals or the inanimate, but in doing so they would have to give that target additional human qualities, to the extent that it would be no different than pretending it was a human agent from the point of view of the theory. Responsibility is a broad term with several senses; the one of interest in this process is referred to by Shaver as "moral accountability", distinct from legal responsibility, the responsibilities of a formal office or mental/emotional capacity (e.g. "He was not responsible for his actions"). Blame is a moral condemnation that follows from responsibility for a morally reprehensible outcome but may be mitigated by *justification* or *excuse*.

Shaver's attribution process begins with an outcome that has been judged negative and evaluates an involved agent for attribution of responsibility against five *dimensions of responsibility*, which are: *causality*, *intentionality*, *coercion*, *appreciation*, and *foreknowledge*. Shaver's process is sequential in its evaluation. We discuss the role of each dimension in the attribution process in turn.

Schultz and Schleifer [Schultz & Schleifer 1983] argue that a judgment of responsibility presupposes a judgment of cause. While Shaver points out that responsibility judgment may be driven purely by a desire for answerability, without a causal connection, his model of attribution adheres to the principle that there must be some judgment of cause for any responsibility to be attributed.

Shaver describes intention as a scale of deliberateness with intentional at one end and involuntary at the other. He describes it as the central concern in attribution of responsibility, and claims that a judgment of intention should result in the strongest judgment of responsibility. There are, however, exceptions to be found in the judgments of coercion and appreciation.

Coercion captures the force exerted by another agent which limits the available choices, from a social standpoint, for the agent in question. This could be through some direct threat or via an authority relationship. In Shaver's model coercion comes into play only once intentionality has been established – it therefore covers only that influence which leads to intentional obedience. An agent who is coerced is assigned less responsibility than one who acts intentionally in the absence of coercion.

In the appreciation dimension the perceiver judges whether the agent in question has the capacity to understand that the outcome in question is morally wrong. If the agent does not have such capacity, then they still bear some responsibility, but are held exempt from blame.

Foreknowledge is defined as the extent to which the agent was aware that a particular action would result in a particular outcome, prior to execution. As with all aspects of this process, it is the perceiver's judgment of the knowledge the agent possessed that is evaluated. Foreknowledge may also be the perceiver's judgment of what knowledge the agent should have had. In the absence of intention, foreknowledge becomes the driving question for responsibility. Shaver attributes less responsibility to an agent who should have known than to one that did know, and less to either than to an agent in the intentional cases above.

Once an agent has been judged responsible, blame follows unless there is a successful intervention by justification or excuse. Justification does not deny responsibility but presents a reason why responsibility for a negative outcome should not carry the negative attribution of blame. This would be the case where someone shot someone else dead, but did it in self-defense. Excuses deny responsibility by appealing the judgments of the dimensions. Examples are "I didn't do it", "I didn't know", "I didn't mean it", "He made me do it" and "She doesn't know it's wrong". Successful intervention by an excuse alters the assignment of responsibility.

Mao and Gratch Computational Model

Mao [Mao 2006], in collaboration with Gratch [Mao & Gratch, 2005] developed a computational model of responsibility assignment which models the judgments of *attribution variables* based on the dimensions of causality, intentionality, coercion and foreknowledge, and the attribution of blame[1] following from those judgments. It does not deal with justifications and excuses, thus blame follows directly from responsibility.

In Mao's model, actions are encoded via hierarchical plans. Non-primitive actions can have multiple decompositions, representing alternative ways the action can be achieved. Actions have propositional preconditions and effects, as well as slots to indicate the agent that could or did perform the action and the agent under whose authority the action falls. Communicative events are modeled as a sequence of *speech acts* [Austin 1962; Searle 1969] representing informing, requesting, and negotiations.

Mao describes a set of inference rules that takes these representations and assigns values to attribution variables that capture how each involved agent is judged with respect to each negative outcome. Causality is ascertained by performance of the primitive action that resulted in the outcome. For intention, a significant distinction is made between act and outcome intention, following from [Weiner 2001]. It is assumed that an agent intends any action that he or she performs or orders performed. However, act intention implies intention of at least one outcome, not all of the outcomes. When an action has multiple possible decompositions and the performing agent was allowed to choose the decomposition, outcome intention moves from the ordering agent to the performing agent if not all decompositions led to the negative outcome. Coercion is inferred from order negotiation; one agent ordering another to perform an action shows act coercion and possibly outcome coercion, depending on how much choice the performing agent had in carrying out their orders. The rules for determining outcome coercion follow the same logic as for determining outcome intention, with the additional constraint that an agent with prior intention is not coerced by being ordered to do what they already intended. Foreknowledge is strongly implied by communication of knowledge of the outcome to another agent before the action is executed. It is also implied by intention, as one cannot intend what one is unaware of.

Mao's work is an important step towards modeling blame attribution. However, there are three limitations we address here. First, as [Mao 2006] observes, it uses Boolean values for attribution variables, whereas Attribution theory describes the dimensions of responsibility in terms of scalar values. Second, all blame is assigned to a single agent (or group of agents in a joint action). This is inconsistent with the human data in Mao's own experiment. Third, the degree of blame assigned by the system is limited to a value of *high* for intentional action and a value of *low* in the absence of intention. These assignments do not match up with the human data.

[1] Social psychological research cited in [Mao 2006] indicates that there are differences in the processes used for responsibility for positive events and negative events, hence the exclusive focus on negative events here.

Qualitative Model of Attribution

We claim that these limitations can be addressed by an encoding of Attribution theory using the principles of Qualitative Process (QP) theory [Forbus 1984]. We claim that this model makes more informative distinctions between blame assignments within and across scenarios.

While physical domains have been a major focus of QR research, an increasing number of researchers have found these techniques useful in fields where theories are expressed in continuous parameters more generally, including organization theory (cf. [Kamps & Peli, 1995]), economics (cf. [Steinmann, 1997]), and political reasoning [Forbus & Kuehne 2005]. Qualitative reasoning, we believe, provides an especially appropriate level of representation for reasoning about social causality. Theories typically are expressed in terms of continuous parameters, such as "amount of intention" and "degree of foreknowledge", but there tend to not be principled ways to move to quantitative models and numerical values for such parameters. In those circumstances, qualitative modeling becomes the most rigorous way to proceed, and ordinal fits with human data becomes the most robust measure.

Our model takes as input the same attribution variables generated by Mao's planning and dialogue inference systems; we tackle neither of those issues, since it is not clear that QR has much to say about them. We extend the inferences on those variables to allow qualitative rather than just Boolean values, and support assignment of responsibility to multiple recipients. Finally, we model Shaver's attribution process to judge responsibility based on those values. Because we omit justification and excuse at this time, we speak in terms of responsibility rather than blame.

In our model, judgments of responsibility, as well as the attribution variables for intentionality, coercion and foreknowledge, are represented by nonnegative continuous parameters. Judgments of causality remain Boolean as that is the extent of their impact in Shaver's model of attribution. A value of zero is a lower limit point indicating the absence of responsibility, intentionality, coercion or foreknowledge in the judgment of the perceiver. Foreknowledge is a function of time: it is the knowledge about the outcome of an action held over an interval prior to, during and after the action. We represent both foreknowledge that the perceiver believes the agent had (epistemic) and should have had (expected). Intention is also evaluated with respect to time as it may be judged to vary over time. We use Allen's interval calculus relations *contains* and *overlaps* in our inference rules [Allen 1983].

For a given scenario, where Mao's system asserts a value of true for intention, coercion or foreknowledge, we assert a value greater than zero. Where epistemic foreknowledge is inferred in Mao's system by communication of the knowledge, we assert equality to an upper limit point of certainty.

In attribution theory, intention does not refer to desire. That is to say, an agent who points a gun and pulls the trigger may or may not have wanted that person to die, but they certainly intended for their action to produce that outcome. Even with that distinction, there is much philosophical discussion on the meaning of intention. According to Shaver, a judgment of intention presupposes epistemic foreknowledge, but not the other way around [Shaver 1985]. On the other hand, Bratman argues that epistemic foreknowledge, and the degree to which it is certain, combined with action must imply intention [Bratman 1990]. Acknowledging these differences in opinion, our model makes the weaker inference that when an agent is certain of an outcome and performs or authorizes the action, it implies only some non-zero level of intention.

Attribution of responsibility from the attribution variables begins with an assessment of eligibility. The agent that performed the action that caused the outcome is eligible of course. Where an agent is in a position of authority over the action that caused the outcome, that agent is also eligible. In both cases, the agent is *responsible by action.* In the case of coercion, the coercing agent is *responsible by coercion* and is also eligible for responsibility for the outcome. Note that R2 and R3 are mutually exclusive – an authority who coerces is responsible by coercion, not by action. These rules are as follows:

```
R1: causes(?action, ?outcome) ∧
    performedBy(?action, ?agent)
⇒ responsibleByActionFor(?agent, ?action, ?outcome)

R2: causes(?action, ?outcome) ∧
    authorizedBy(?action, ?agent) ∧
    performedBy(?action, ?coerced) ∧
    CoercionFn(?agent, ?coerced, ?outcome) = 0
⇒ responsibleByActionFor(?agent, ?action, ?outcome)

R3: causes(?coercion-action,
           CoercionFn(?agent, ?coerced, ?outcome) > 0) ∧
    performedBy(?coercion-action, ?agent)
⇒ resposibleByCoercionFor(?agent, ?coercion-action,
                          ?outcome)
```

Given our omission of the more special-case dimension of appreciation, Shaver's attribution process displays four distinct modes of judgment: causal without foreknowledge, causal without intent, intentional but coerced and intentional in the absence of coercion. Responsibility is strictly increasing across those modes, in that order. Within each state, responsibility is qualitatively proportional to a different attribution variable. These modes translate into six model fragments or views in our model.

The first two modes translate directly into two views. The third and fourth modes each translate into two views based on whether the agent being considered is responsible by action or coercion. In the former case, intention and foreknowledge are measured at the time of the causal action. In the latter case, they are measured at the time of the coercing action. The six views are as follows:

```
View: CausalWithoutForeknowledge
Conditions:
 responsibleByActionFor(?agent, ?action, ?outcome) ∧
 ¬∃?s1(KnowledgeFn(?agent,
                   causes(?action, ?outcome),
                   ?s1) > 0 ∧
       contains(?s1, ?action)) ∧
 ¬∃?s2(IntentionFn(?agent, ?outcome, ?s2) > 0 ∧
       contains(?s2, ?action)) ∧
 Knowledge-ExpectedFn(?agent, causes(?action, ?outcome),
                      ?s3) > 0 ∧
 contains(?s3, ?action)
Consequences:
 ResponsibilityFn(?agent, ?outcome) ∝Q+
   Knowledge-ExpectedFn(?agent,
                        causes(?action, ?outcome),?s3)

View: CausalWithoutIntent
Conditions:
 responsibleByActionFor(?agent, ?action, ?outcome) ∧
 KnowledgeFn(?agent, causes(?action, ?outcome),
             ?s1) > 0 ∧
 contains(?s1, ?action)) ∧
 ¬∃?s2(IntentionFn(?agent, ?outcome, ?s2) > 0 ∧
       contains(?s2, ?action))
Consequences:
 ResponsibilityFn(?agent, ?outcome) ∝Q+
   KnowledgeFn(?agent, causes(?action, ?outcome), ?s1)

View: IntentionalButCoerced
Conditions:
  responsibleByActionFor(?agent, ?action, ?outcome) ∧
  IntentionFn(?agent, ?outcome, ?s1) > 0 ∧
  contains(?s1, ?action))) ∧
  CoercionFn(?coercer, ?agent, ?outcome) > 0
Consequences:
  ResponsibilityFn(?agent, ?outcome) ∝Q+
    CoercionFn(?coercer, ?agent, ?outcome)

View: IntentionalByCoercionButCoerced
Conditions:
  responsibleByCoercionFor(?agent, ?coercion-action,
                           ?outcome) ∧
  IntentionFn(?agent, ?outcome, ?s1) > 0 ∧
  contains(?s1, ?coercion-action))) ∧
  CoercionFn(?coercer, ?agent, ?outcome) > 0
Consequences:
  ResponsibilityFn(?agent, ?outcome) ∝Q+
    CoercionFn(?coercer, ?agent, ?outcome)

View: Intentional
Conditions:
  responsibleByActionFor(?agent, ?action, ?outcome) ∧
  IntentionFn(?agent, ?outcome, ?s1) > 0 ∧
  contains(?s1, ?action))) ∧
  ¬∃?coercer(CoercionFn(?coercer, ?agent, ?outcome)
             > 0)
Consequences:
  ResponsibilityFn(?agent, ?outcome) ∝Q+
    IntentionFn(?agent, ?outcome, ?s1)

View: IntentionalByCoercion
Conditions:
  responsibleByCoercionFor(?agent, ?coercion-action,
                           ?outcome) ∧
  IntentionFn(?agent, ?outcome, ?s1) > 0 ∧
  contains(?s1, ?coercion-action))) ∧
  ¬∃?coercer(CoercionFn(?coercer, ?agent, ?outcome)
             > 0)
Consequences:
  ResponsibilityFn(?agent, ?outcome) ∝Q+
    IntentionFn(?agent, ?outcome, ?s1)
```

Given a scenario with a negative outcome and some number of agents, our system first infers which agents bear some level of responsibility for the outcome. For each agent in that set, it infers what mode of judgment to use and the qualitative proportionality that constrains the amount of responsibility attributed. Given a number of such scenarios, our system is able to infer ordinal constraints on responsibility for all pairs of agents both within and across the scenarios. Clearly for situations where two responsibility judgments being considered fall into different judgment modes, the inference is straightforward. For judgments within the same mode, we can infer relative amounts of responsibility when ordinal relationships between the control parameters are known.

In Mao's inference system, evidence of intention prior to coercion determines the strength of the coercion. If the agent in question intended the action or outcome prior to being ordered to do it, then there is no coercion. If the agent did not intend it, then there is strong coercion. If the agent's prior intent is unknown, then there is weak coercion. The strong/weak distinction is not used in Mao's attribution process, but is targeted towards a probabilistic extension of the system. We modify this rule to infer ordinal constraints on the amount of coercion as follows:

```
R4: causes(?coercion-action1,
           CoercionFn(?coercer1, ?agent1, ?outcome1)
                > 0) ∧
    IntentionFn(?agent1, ?outcome1, ?s1) = 0 ∧
    overlaps(?s1, ?coercion-action1) ∧
    causes(?coercion-action2,
           CoercionFn(?coercer2, ?agent2, ?outcome2)
              > 0) ∧
    ¬∃?s2(IntentionFn(?agent2, ?outcome2, ?s2) = 0 ∧
          contains(?s2, ?coercion-action2))
⇒ CoercionFn(?coercer1, ?agent1, ?outcome1) >
    CoercionFn(?coercer2, ?agent2, ?outcome2)
```

In the dimension of causality, Shaver argues that omission is just as blameworthy as commission. In our model we extend this allowance to the dimension of coercion. As stated in rule R2, an agent who is in a position of authority over a causal action is extended eligibility for responsibility. If the authority is aware of the possibility of a negative outcome from the underling's actions, yet does not coerce his or her underling away from that outcome, then he or she is guilty of abdicating authority. Under these circumstances the authority is subject to the same evaluation of intention as the underling. However, if the authority is unaware of the underling's actual intention to cause that outcome, then his or her outcome intention is constrained to be less than the intention of the underling. These rules are as follows:

```
R5: causes(?action, ?outcome) ∧
    performedBy(?action, ?underling) ∧
    authorizedBy(?action, ?authority) ∧
    CoercionFn(?authority, ?underling, ?outcome) = 0 ∧
    KnowledgeFn(?authority, causes(?action, ?outcome),
                ?s1) > 0 ∧
    contains(?s1, ?action)
⇒ abdicatedAuthority(?authority, ?underling,
                     ?action, ?outcome)
```

```
R6: abdicatedAuthority(?authority, ?underling,
                       ?action, ?outcome) ∧
    IntentionFn(?underling, ?outcome, ?s1) > 0 ∧
    contains(?s1, ?action) ∧
    IntentionFn(?authority, ?outcome, ?s2) > 0 ∧
    contains(?s2, ?action) ∧
    ¬∃?s4(KnowledgeFn(?authority,
                      (IntentionFn(?underling, ?outcome,
                                   ?s3) > 0 ∧
                       contains(?s3, ?action))
                      ?s4) ∧
          contains(?s4, ?action))
⇒ IntentionFn(?underling, ?outcome, ?s1) >
     IntentionFn(?authority, ?outcome, ?s2)
```

Finally, the outcome intention of an agent who chooses not to coerce, even one in authority, must be considered less than that of an agent who chooses to coerce. Again as follows:

```
R7: IntentionFn(?agent1, ?outcome1, ?s1) > 0 ∧
    contains(?s1, ?action1) ∧
    CoercionFn(?agent1, ?coerced1, ?outcome1) > 0 ∧
    causes(?action2, ?outcome2) ∧
    abdicatedAuthority(?agent2, ?underling,
                       ?action2, ?outcome2) ∧
    IntentionFn(?agent2, ?outcome2, ?s1) > 0 ∧
    contains(?s2, ?action2)
⇒ IntentionFn(?agent1, ?outcome1, ?s1) >
     IntentionFn(?agent2, ?outcome2, ?s2)
```

Evaluation

Mao presents an evaluation of her system against human data collected in a survey of 30 respondents. The survey presented four scenarios, variations starting with the "*company program"* scenario used by Knobe [Knobe 2003], replicated below. The scenarios involve two agents, a chairman and a vice president, and a negative outcome of environmental harm. Each scenario was followed by a set of Yes/No questions intended to validate the judgments of intermediate variables, including the attribution variables, and a final question asking the respondent to score the blame each agent deserved on a scale of 1-6. Due to space limitations, we refer the reader to [Mao 2006] for details on the data collection process.

Corporate Program Scenarios

Scenario 1. The vice president of Beta Corporation goes to the chairman of the board and requests, "Can we start a new program?" The vice president continues, "The new program will help us increase profits, and according to our investigation report, it has no harm to the environment." The chairman answers, "Very well." The vice president executes the new program. However, the environment is harmed by the new program.

Scenario 2. The chairman of Beta Corporation is discussing a new program with the vice president of the corporation. The vice president says, "The new program will help us increase profits, but according to our investigation report, it will also harm the environment." The chairman answers, "I only want to make as much profit as I can. Start the new program!" The vice president says, "Ok," and executes the new program. The environment is harmed by the new program.

Scenario 3. The chairman of Beta Corporation is discussing a new program with the vice president of the corporation. The vice president says, "The new program will help us increase profits, but according to our investigation report, it will also harm the environment. Instead, we should run an alternative program, that will gain us fewer profits than this new program, but it has no harm to the environment." The chairman answers, "I only want to make as much profit as I can. Start the new program!" The vice president says, "Ok," and executes the new program. The environment is harmed by the new program.

Scenario 4. The chairman of Beta Corporation is discussing a new program with the vice president of the corporation. The vice president says, "There are two ways to run this new program, a simple way and a complex way. Both will equally help us increase profits, but according to our investigation report, the simple way will also harm the environment." The chairman answers, "I only want to make as much profit as I can. Start the new program either way!" The vice president says, "Ok," and chooses the simple way to execute the new program. The environment is harmed.

	Human Data		**Mao Model**		
	Chair	VP	Chair	VP	Degree
Scenario1	3.00	3.73		Y	Low
Scenario2	5.63	3.77	Y		Low
Scenario3	5.63	3.23	Y		Low
Scenario4	4.13	5.20		Y	High

Table 1. Blame attribution results

Table 1 shows for each scenario the average blame attributed to each agent by the survey respondents, the single choice of the blameworthy agent made by Mao's system and the degree of responsibility for that agent asserted by Mao's system. In each scenario, Mao's model correctly selects the agent who receives the higher degree of blame, but with the incorrect implication that the other agent involved is free of responsibility. The degree of responsibility assertions made by Mao's model do not match the human data.

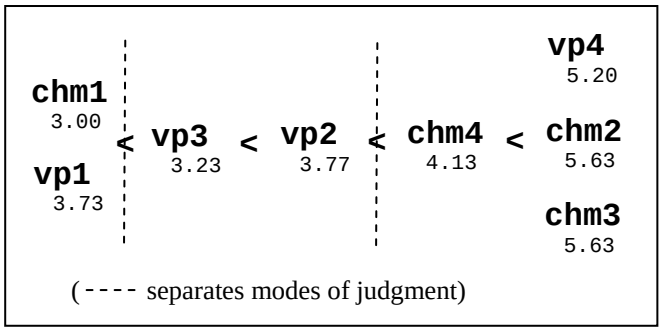


Figure 1. Ordinal constraints on responsibility and average participant attribution numbers

Figure 1 shows the ordinal constraints inferred by our model on the amount of responsibility for the agents in the four scenarios, together with labels indicating the average blame attributed to each by the survey respondents.

The eight agents being considered fall into three of the four modes of judgment from Shaver's attribution theory. The ordering of those modes establishes the ordinal relations between all pairs of agents in different modes. The chairman and vice president in scenario 1 both fall into the `CausalWithoutForeknowledge` view which is part of the causal without foreknowledge mode of judgment. Within this view, the responsibility of each agent is qualitatively proportional to only the amount of expected foreknowledge each agent is judged to have. As there are no inferred constraints on their expected foreknowledge, they remain unordered. The vice president in scenario 2 and the vice president in scenario 3 fall into the `IntentionalButCoerced` view which is part of the intentional but coerced mode of judgment. Their respective degree of responsibility is qualitatively proportional to the amount of coercion judged to have been applied. There is no indication of the outcome intention of the vice president in scenario 2 prior to the coercion action, while the vice president in scenario 3 clearly shows lack of outcome intention prior to being coerced. The vice president in scenario 3 is therefore judged to have a higher degree of coercion by rule R4 and thus a lower degree of responsibility. The chairman and vice president in scenario 4 fall into the `Intentional` view while the chairmen from scenarios 2 and 3 fall into the `IntentionalByCoercion` view, both of which are part of the intentional mode of judgment. Their degrees of responsibility are qualitatively proportional to their outcome intention. The chairman in scenario 4 abdicated his authority to the vice president as captured by rule R5. However, because he did not coerce the outcome nor did he have prior knowledge of the vice president's intention, he is constrained to have a lower degree of intention than the other three by rules R6 and R7. This results in a lower judgment of responsibility while the other three remain unordered.

In 23 of the 28 possible comparisons between agents our system correctly infers which agent should receive more blame. In 4 of the remaining comparisons, our system establishes a constraint between the degree of responsibility and the value of an attribution variable for each agent, but cannot infer an ordinal relation between those control variables. In the remaining case our model is inconsistent; comparing the vice president in scenario 1 with the vice president in scenario 3, the survey respondents attributed less blame to the agent who had foreknowledge but was coerced than to the agent with no foreknowledge at all. Interestingly, this constitutes a violation of the strict ordering of the modes of judgment assumed in Shaver's model. The vice president in scenario 1 has no foreknowledge of the environmental harm and thus no intention to cause it, placing the judgment of his responsibility in the causal without foreknowledge mode. On the other hand, the vice president in scenario 3 has foreknowledge but is strongly coerced, placing the judgment of his responsibility in the intentional but coerced mode. This overlap between these two states indicates that, under some circumstances, an agent acting under coercion with full foreknowledge of the consequences may be counted less responsible than one who simply does not know the outcome. The first scenario was worded differently than the others, in that the vice president is presented as initiating a new program that the chairman had no prior knowledge of. In the other scenarios the program is assumed to already be known to both participants. Further, the vice president in scenario 3 is explicitly shown to have expended some amount of effort to avoid the outcome. We suspect that these differences introduce a variable of personal desire to run the program or not on the part of the vice presidents, which is distinct from intention and not accounted for in the current model nor in the underlying theory.

Based on the four cases where our model infers a constraint with a free variable, we can make predictions about additional constraints in the attribution variables. Given that participants attributed a higher degree of blame to the vice president in scenario 1 over the chairman, our model predicts that they would also indicate that the vice president was more responsible than the chairman to know that environmental harm would result from the new program. Likewise, as the respondents attributed equal blame to the chairmen in scenarios 2 and 3, our model predicts that they would judge the outcome intention of the chairmen as being equal as well. This is consistent with the implicit claim in attribution theory that, while coercion mitigates the responsibility of the coerced, it has no such effect on the responsibility of the coercer, who is judged on his intention instead. Finally, as the respondents attributed less blame to the vice president in scenario 4 than to the chairmen in scenarios 2 and 3, our model predicts that they would judge the outcome intention of that vice president to be less than the outcome intention of each chairman, respectively.

Discussion

We have shown that QP theory can be used to formally encode a model for attributing responsibility for negative outcomes, based on Attribution theory. Our model explains the corporate scenario data better than Mao's model does, due to our use of qualitative representations instead of categorical, Boolean values. While a purely qualitative model would not be sufficient for all purposes – for example, deciding whether or not someone was blameworthy enough to report an action – our evaluation suggests that qualitative modeling captures an important level of reasoning about social situations. Even when numerical models are desired, working out qualitative models first could provide constraints on more detailed models. And, as our demonstration of the violation of an assumption of Shaver's theory indicates, formally

encoding qualitative models and examining ordinal fits with human data could provide social scientists with a new set of tools for exploring the consequences of their theories.

As part of our ongoing work on narrative understanding, we intend to incorporate this model into our Explanation Agent natural language understanding system [Kuehne 2004]. This work represents part of a larger effort to model and reason about moral decisions presented in folktales and the explanatory stories that people tell. In that context, we plan to expand the factors that go into judging attribution variables beyond plan recognition and order negotiation speech acts and do further evaluation of the validity of those judgments and the predictions made by this model regarding the attribution of blame.

Acknowledgements

This research was supported by the Air Force Office of Scientific Research. We thank Jon Gratch for useful conversations.

References

Allen, J. F. and D. Waltz. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26, 832-843.

Austin, J. 1975. *How to Do Things with Words*. Harvard University Press.

Bratman, M. 1990. What is intention? In P. Cohen, J. Morgan & M. Pollack eds., *Intentions in Communication*. MIT Press.

Forbus, K. 1984. Qualitative Process Theory. *Artificial Intelligence*, 24, 85-168.

Forbus, Kenneth D. & Sven Kuehne. 2005. Towards a qualitative model of everyday political reasoning. *Proceedings of the Nineteenth International Qualitative Reasoning Workshop*.

Heider, F. 1958. *The Psychology of Interpersonal Relations*. John Wiley & Sons Inc.

E. E. Jones and K. E. Davis. 1965. From Acts to Dispositions: The Attribution Process in Person Perception. In L. Berkowitz ed. *Advances in Experimental Social Psychology* (Vol.2), pp. 219-266. Academic Press.

Kamps, J. and Peli, G. 1995. Qualitative reasoning beyond the physics domain: The density dependency theory of organizational ecology. *Proceedings of the Ninth International Qualitative Reasoning Workshop.*

H. H. Kelley. 1973. The Processes of Causal Attribution. *American Psychologist*, 28:107-128.

J. Knobe. 2003. Intentional Action and Side-Effects in Ordinary Language. *Analysis*, 63:190-193.

Kuehne, S. E. 2004. On the Representation of Physical Quantities in Natural Language Text. *Proceedings of the Twenty-sixth Annual Meeting of the Cognitive Science Society*, Chicago, Illinois, USA, August.

Mao, W. 2006. *Modeling Social Causality and Social Judgment in Multi-Agent Interactions.* Doctoral dissertation, University of Southern California, Los Angeles, California.

Mao, W. and Gratch, J. 2005. Social Causality and Responsibility: Modeling and Evaluation. *Fifth International Conference on Interactive Virtual Agents*.

Shultz, T.R. & Schleifer, M. 1983. Towards a refinement of attribution concepts. In J. Jaspars, F. D. Fincham, & M. Hewstone (Eds.), *Attribution theory and research: Conceptual, developmental and social dimensions* (pp. 37-62). London: Academic.

J. R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

K. G. Shaver. 1985. *The Attribution Theory of Blame: Causality, Responsibility and Blameworthiness.* Springer-Verlag.

Steinmann, C. 1997. Qualitative reasoning on economic models. *Proceedings of the Eleventh International Qualitative Reasoning Workshop.*

B. Weiner. 1995. *Judgments of Responsibility: A Foundation for a Theory of Social Conduct*. Guilford Press.

B. Weiner. 2001. Responsibility for Social Transgressions: An Attributional Analysis. In B. F. Malle, L. J. Moses and D. A. Baldwin eds. *Intentions and Intentionality: Foundations of Social Cognition*, pp. 331-344. MIT Press.

Generating test-cases from qualitative knowledge – Preliminary report

Franz Wotawa*
Technische Universität Graz, Institute for Software Technology
Inffeldgasse 16b/2, A-8010 Graz, Austria
wotawa@ist.tugraz.at

Abstract

In this paper we introduce a methodology for extracting test-cases from qualitative knowledge which represents the expected behavior of the environment of a system under test. Usually in software engineering test-cases are only derived from the requirements documentation and do hardly consider environmental constraints. This is especially problematic if a system like a mobile device has interactions with the environment which cannot be foreseen in advance in all details. An approach that is based on the behavior of the world which is external to the system would help to generate test cases which are realistic and capture the whole range of interactions. The use of qualitative reasoning for representing for example the physical world is an advantage because the underlying models capture all possible behaviors and thus guarantee completeness of the generated test-case set to some extent.

Introduction

The complexity of systems and software increases every year which is mainly caused by a strong demand for smarter products that have to provide more and more functionality. For example in the automotive industry the number of CPUs with control software running on it is still increasing. You should not wonder that the number of such control units onboard of a vehicle is likely to more than 40. Because of the fact that control units even implement different functionalities there is also heavy communication between them. Hence, complexity of the whole system increases and makes it difficult to construct such systems and to validate and verify them. Moreover, there is a tendency that less and less effort is spent in verification and validation (V & V) due to market requirements, e.g., pre-defined dates for introducing a new product, and economical requirements, e.g., labor costs and expected revenue. Such considerations of course do not apply in the development of safety-critical systems like vehicles.

A proposed solution to overcome the mentioned problems is to automate testing to some extent. This can be done by providing tools for test execution and test-case generation. The former helps to reduce effort in running test-cases especially in cases of several product releases and where regression testing is necessary. Such test execution tools also provide statistical information regarding failing test-cases and other measures. Test-case generation tools are used to generate test-cases from specifications or from the source code which fulfill certain requirements like coverage or maximizing the mutation score. One important aspect of development of good test-cases especially for system tests as part of the validation procedure is that test-cases should reflect possible interactions between the system and its environment. For example, if there is no pre-defined order for entering data into a system, different sequences have to be tested. Moreover, unexpected but possible interactions have to be tested. For example, what happens when killing a process which has an open transaction with a database? If the database software is correct, the transaction is not allowed to be confirmed in order to ensure integrity. In this case a roll-back procedure would be necessary.

In this paper, we focus on the generation of test-cases from qualitative models. The reason for that is the need for test-cases which test not only specified requirements but also unexpected but still possible interactions of the system with the environment. Qualitative reasoning is appropriate for that purpose especially when generating test-cases for embedded systems that have to work more or less autonomously. One reason is the fact that QR models capture all possible behaviors which make them a perfect choice for explanation and in our case test-case generation. For test-case generation using QR models we have two application areas in mind:

1. Systems that have to have knowledge of its domain in order to fulfill a certain task. Such systems might be control or decision support systems. For example, if we want to have an automated advisor that helps us in deciding actions for protecting the ecosystem (or some parts of it like a river), we have to provide test-cases which describe a range of possible scenarios. For the generation of such test-cases someone obviously has to have knowledge about the considered ecosystems. If we want to automate test-case generation, the test-case generator has to use knowledge about possible behaviors of the ecosystem.

*This work has been supported by the FIT-IT research project *Self Properties in Autonomous Systems project (SEPIAS)* which is funded by the Austrian Federal Ministry of Transport, Innovation and Technology and the FFG.

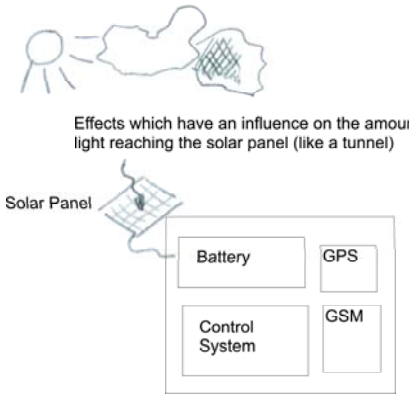


Figure 1: An mobile embedded system with a solar panel for providing electricity

2. The second scenario considers systems that implement a certain functionality and interact with the surrounding environment. In this case knowledge about the behavior of the environment can be used to generate test-cases that are unexpected. In this case the environment knowledge is not used to test the basic functionality of the system but to test the interaction with its environment. This is especially important for system tests.

In the paper we will use an example from the second application area but the ideas should be applicable as well for the other area.

Basic idea

To illustrate the basic idea of our approach we use a small example which is depicted in Figure 1. In this example a solar panel is used to provide electrical power of mobile embedded systems. The system comprises a battery which is loaded whenever the solar panel is providing energy, and other components like a GPS for measuring the global position and a GSM module for communicating with a server. Because the systems have to fulfill a task, e.g., sending its position to the server every minute, it requires electrical power. During the night the power is provided only by the battery and thus the battery is discharged. Of course the system should be designed in a way that the capacity of the battery is large enough to provide electrical power for the whole night. However, the system's designer might not considered all possible situations when computing the required capacity. For example, there can be several cloudy days where the solar panel does not provide enough power. The solar panel might produce no electricity because the device is in a building or a tunnel. Because of aging the battery capacity and the solar panel capabilities are decreasing. Moreover, the device is used in the north of the globe during winter where there is almost no sun light during a day.

A qualitative model describing the mentioned situation has to describe the relationships between the important entities. In particular, we introduce the following variables together with their domains:

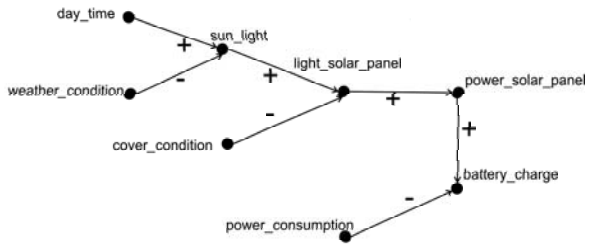


Figure 2: The cause-effect relationships of the solar panel example

`day_time`	{day, night}
`weather_condition`	{sunny, cloudy }
`sun_light`	{ no, medium, full }
`cover_condition`	{ yes, no }
`light_solar_panel`	{ no, medium, full }
`power_solar_panel`	{ no, medium, full }
`battery_charge`	{ empty, medium, full }

The `day_time` variable indicates whether it is currently day or night. `weather_condition` is for stating the current weather situation where in our case only the amount of clouds is required. From `day time` and `weather_condition` we can derive the amount of sun light (`sun_light`) at the ground level of the earth. If there the solar panel is not covered by anything,e.g., `cover_condition` is false, `sun_light` has to be equivalent to the light available directly at the panel (`light_solar_panel`). Otherwise, there is no light available for the solar panel to produce electrical power. Depending on the light at the solar panel more or less power (`power_solar_panel`) can be produced. And from the energy we can derive the charging status of the battery. Figure 2 depicts the cause-effect relationship between the variables. Note that this model is of course simplified and does not handle all possibilities. The model can be extended to represent parts of the internal behavior of the mobile embedded system, e.g., the influence of the subsystems to the status of the battery charge capacity.

This model can be formally represented using a qualitative reasoning methodology like Qualitative Process Theory (Forbus 1984) or Qualitative Simulation (Kuipers 1986). A qualitative reasoning engine like Garp 3 (Bredeweg *et al.* 2006) can be used to produce simulation runs. For this paper we assume that the qualitative model together with a reasoning engine is available.

For example, we know that the weather condition has a negative impact on the amount of sun light if we assume the following order `sunny<cloudy` and `no<medium<full` defined for the domains of the corresponding variables. If the weather condition increases (which means the amount of clouds is increasing), then the sun light is decreasing. Using the QSIM representation we would write M^{-}(`weather_condtion`, `sun_light`). For other relationships between variables we are able to define similar relationships. Moreover, we specify a relationship between the state of the charge of the battery and the power consumption of the device in order to compute the

discharging of the battery whenever a consumer is added. If a device is added to a battery there is a negative impact on the battery's charge capacity:

$M^{-}(\texttt{power\_consumption},\texttt{battery\_charge})$.

Hence, we might be interested in finding a behavior that leads to a value of `no` for `battery_charge` when assuming that the battery is fully charged at the beginning.

In order to compute such a behavior we have to have information about the values of some variables, the model, and a test condition. The latter specifies the behavior to be searched by simulating the model. For our example, one test-case might be of the form:

$$\left\langle \begin{array}{c} (\texttt{day\_time},\texttt{night}) \\ \ldots \\ (\texttt{power\_consumption},\texttt{yes}) \\ (\texttt{battery\_charge},\texttt{full}) \end{array} \right\rangle_0$$

$$\left\langle \begin{array}{c} (\texttt{day\_time},\texttt{night}) \\ \ldots \\ (\texttt{power\_consumption},\texttt{yes}) \\ (\texttt{battery\_charge},\texttt{medium}) \end{array} \right\rangle_1$$

$$\left\langle \begin{array}{c} (\texttt{day\_time},\texttt{night}) \\ \ldots \\ (\texttt{power\_consumption},\texttt{yes}) \\ (\texttt{battery\_charge},\texttt{empty}) \end{array} \right\rangle_2$$

Note that the values of other variables over time are not given for this example. The test-case states that the battery can be discharged during the night. Such a test-case is an abstract test-case because it does not provide any information about the quantities, e.g., the length of night. Hence, in order to create an executable test-case we have to instantiate or refine the abstract test-case. For example, we might define that a night lasts for 12 hours and thus we have to test whether the device is designed to fulfill its expected functionality during the night or even longer.

Definitions and algorithms

Generating test-cases automatically requires the availability of a formal model (or the source code). In our case we assume that the formal model is a qualitative model which captures the important aspects of the environment and of the system to be tested. In the example given in the previous section we introduced variables for stating properties of the solar panel, the battery as well as the power consumption which correspond to parts of the system. In the following definition of the model to be used for test-case generation we distinguish the environment model from the system model. The purpose of separating the model is that re-use of models is supported. It is very likely that the environment part of the model can be used together with models of different systems under test.

Definition 1 (Test-case model) *A test-case model is a tuple $(QM \cup QS, C_I, C_T)$ where QM is the qualitative model of the environment, QS is the partial qualitative model of the system, C_I is a set of constraints of input variables, and C_T is a set of test-case constraints.*

Note that in the definition of a test-case model a test purpose can be specified using C_T. The test purpose gives information about which test-case to generate. In our previous example we wanted to have a test-case which leads to an empty battery. Formally, we would write the input and test-case constraint of our example as follows:

$$C_I = \{(\texttt{day\_time},\texttt{night})\} \cup C$$
$$C_T = \{(\texttt{battery\_charge},\texttt{empty})\}$$

In this example C denotes the set of constraints stating that it is not possible for variables to have more than one value assigned at the same time.

Given a test-case model we now are interested in specifying a test-case. We do this by defining a test-case as the outcome of a qualitative simulation where all input constraints have to be fulfilled and where the test purpose is reached.

Definition 2 (Abstract test-case) *Given a test-case model $(QM \cup QS, C_I, C_T)$ and a qualitative simulator QEXEC. An abstract test-case t is the result of a run of the simulator on the model where the inputs do not contradict C_I and t covers all requirements of C_T, i.e., $t = QEXEC(QM \cup QS, i)$ where $i \cup C_I$ is consistent and $endState(t) \models C_T$.*

In the definition of abstract test-cases the constraints which apply on the test-case, i.e., the test-case requirements C_T are somehow stronger that the constraints for the inputs. A reason is that we expect a test-case to entail all requirements whereas consistency checks are enough for the input. The test purpose has to be reached in all cases.

The following algorithm computes an abstract test-case from the given test-case model. The algorithm does not only compute one test-case for each run of the QR simulation engine but combines different runs. A reason for extending the computation is to generate larger test-cases which are likely to capture a more complex behavior. Moreover, we further be able to represent information about possible inputs over time using C_I. For example, we might be interested in testing the system over several days. Hence, we have to specify that after the night we start with a day which is followed by a night and so on. Such knowledge is assumed to be captured by C_I.

Algorithm *AbstractTestCase*
Inputs: A test-case specification $(QM \cup QS, C_I, C_T)$
Output: An abstract test-case

1. Let t be the empty test-case sequence and let b be the empty behavior.
2. Choose inputs i which are consistent with the input constraints C_I and the last state of the behavior b computed in the previous run.
3. Select a behavior b which is computed by calling a QR engine on model $QM \cup QS$ and inputs i.
4. If no new behavior b exists, then return t.
5. Let t be t extended with b, i.e., $t := t + b$.
6. If t fulfills the criteria for test-cases C_T, then return t. Otherwise, go to 2

The algorithm ***AbstractTestCase*** obviously computes an abstract test-case accordingly to our definition. The algorithm halts if a test-case that entails the test purpose C_T can

be found or if the whole search space has been explored. The latter cannot be guaranteed in general. However, in practice someone would specify a boundary which when reached terminates the computation. The complexity of the algorithm depends on the complexity of the simulator and the size of the model in terms of variables and their domains.

AbstractTestCase can be used to compute a set of test-cases by calling it more often. In this case we might change the input constraints and test purpose. If we do not change the constraints, we also expect that ***AbstractTestCase*** returns different solutions because we assume the selection of inputs and behaviors to be random. In a practical implementation someone might make this selection deterministic. In this case the algorithm can also be extended to compute different solutions by exploring the search space in a breadth-first manner.

Refining test cases

Abstract test-cases cannot be directly used because of the abstraction of time and domain knowledge. In order to have a test-case or a test suite which can be directly used we have to convert the abstract test-case to a concrete one. For this purpose we have to define a function which maps qualitative values to their corresponding quantitative values. This function has to be adapted for specific qualitative models of systems in order to capture the relevant aspects of a system. In particular, the function has to distinguish cases where the state sequence which represents an abstract test-case has to be mapped to a sequence of quantitative variable values from cases where information about dense time is required. In our example, the states carry information about the time during the day.

The function which maps abstract test-cases to their concrete counterparts is called a refinement function because it has to refine the abstract knowledge in order to lead to a specific and executable test-case. In the following, we discuss requirements of refinement functions:

- The refinement function has to preserve the order of events. Consider for example two immediately succeeding states at the qualitative level where we have events x and x' respectively. If we map x to y and x' to y' at the quantitative level, then the occurrence of y has to be before y'. For events occurring at the same state of the qualitative level no ordering can be ensured at the quantitative level.
- The mapping from quantitative values to qualitative ones has to preserve the order relation.
- The refinement function should ignore those events which are neither used to stimulate the system under test nor to check whether the system behaves correctly or not. Hence, events which are only relevant to compute a certain environment behavior but cannot be used to test the system should be ignored.

Given such a refinement function f_R we can compute a concrete test-case from an abstract one. However, even if f_R fulfills all requirements, the test-case at the quantitative level needs not to be a valid test-case in terms of being executable. This problem is similar to the one in verification. In (Ball & Rajamani 2002) a program is compiled into a representation using predicate abstraction. If a counter-example can be obtained from the abstract version of the program, the real program might be correct. Such problems always occur when using abstraction and correspond to the used abstraction mechanism which might ignore knowledge which is necessary to avoid the problem.

The methodology for generating test-cases from qualitative models has to have the following steps:

1. Compute abstract test-cases for the given qualitative model, the input constraints and the test purpose.
2. Apply the refinement function f_R to all abstract test-cases to obtain a set of concrete test-cases.
3. Evaluate the set of concrete test-cases by (i) executing the system under test on those tests and (ii) manually checking the test-cases for plausibility. Test-cases which are not plausible or which cannot be executed because of other reasons can be removed from the test suite.

An advantage of this methodology is that we obtain a lot of tests which based on a firm ground and all of them represent a certain interaction between the system and its environment. Hence, it is very unlikely to miss a test-case because of missing requirements.

The concrete test-case for our example would be of the form:

Simulate a night for a duration of 10 hours. The system is not allowed to run out of power during the simulation.

The specific value for the duration depends on the expected area of operation of the system. Hence, such values have to be specified when defining the refinement function f_R. The assertion that the system is not allowed to run out of power would come from a system engineer and is equivalent to the negation of the qualitative test purpose, i.e., `battery_charge` reaches the value `empty`.

The question of how to execute the obtained test-cases is a different one and is not in the focus of this paper. Moreover, it is not always possible to automate test-case execution. In our example we have to simulate the conditions of a night which can be hardly automated.

Related research

Classical white-box or black-box testing and test-case generation techniques (Beizer 1990) assumes both the existence as well as the accessibility of a source code, or a specification from which test-cases can be extracted. In most cases the objective is to prove the correct implementation of functions. Since specifications are hardly ever complete there might be cases where interactions of systems with their environment lead to harmful situations. This becomes even more critical if the complexity of systems is increasing. One solution to this problem is to provide models of the environment and its interaction with the system to be developed and extract possible interaction sequences for testing. For example, (Auguston, Michael, & Shing 2005) follow this solution.

In classical white-box testing and test-case generation the objective is to generate test-cases which fulfill a certain program coverage criterion like statement or path coverage. Such critera are very usefull for verification purposes because they allow to judge the quality of the used test suite to some extend. However, in black-box testing where a specification is available someone is more interested in test-cases covering the specification. This is the case for our approach where the test-cases are extracted from QR models using possible simulation runs. The quality of the computed test-cases wrt. to program coverage is left for future research.

(Auguston, Michael, & Shing 2005) introduced the use of attributed event grammars for generating test-cases from environment models for reactive systems. In the paper the authors use the grammar for representing an event-based model. Possible execution traces of the model form the test-cases. Insofar the underlying idea for test-case generation as described in this paper is very similar and can also be found in other papers, e.g., in (Fraser & Wotawa 2006a; 2006b). However, the mentioned papers can be distinguished with respect to the underlying modeling language. Whereas Auguston et al. are using attributed event grammars, Fraser et al. are using temporal logic and model-checking techniques, and in this paper we are proposing the use of qualitative models for test-case generation.

(Esser & Struss 2007) focused on test-case generation from finite state machines and on the underlying theory. In their work the authors want to create distinct test-cases, i.e., test-cases that allow for distinguishing different faults. For that purpose, the finite state machine model is compiled into a constraint representation. In contrast to Esser and Struss the methodology proposed in this paper does not require a transformation of the model. Moreover, we are more interested in obtaining test-cases from environment models (like (Auguston, Michael, & Shing 2005)) and not from behavior models of the system.

Conclusion

In this paper we followed the basic idea of (Auguston, Michael, & Shing 2005), i.e., using information about the environment of a system in order to generate test-cases for validating the system. The focus on the behavior of the system's environment is important because this potentially leads to test-cases which would not be generated when considering only the system's functional requirements. In contrast to (Auguston, Michael, & Shing 2005) we introduced the use of qualitative models for describing the behavior of the environment. We argued that qualitative models provide the right means for describing the behavior of the environment in terms of physical laws and causal relationships and the important parts of the system under test. The models themselves can be simulated and thus reveal possible interactions of systems with their environment. Hence, simulation results induce potential test-cases. Because the obtained test-cases present only an abstraction of the real behavior they have to be refined.

Advantages of the proposed methodology are:

- Test-case generation relies on well defined modeling paradigms and simulation engines are available.
- Because QR methods provide all possible behaviors (something which is a drawback in certain cases) even rare interactions of the system with its environment can be found.
- The restricted value domains in QR allows for taking into account all possible inputs. Those input values have to be refined, i.e., the abstract values have to be mapped to their corresponding quantitative values when computing the real test-cases for a system. This refinement step has to be specified by the user of the test-case generator. This is not a drawback but ensures flexibility. Moreover, the same happens when using abstraction which is sometimes necessary for formal verification and test-case generation using classical methods.
- QR is very well adapted for representing environment models and thus makes it very attractive for generating test-cases for reactive systems which interact with the environment.

Future research has to provide case studies for generating test-cases from QR models.

References

Auguston, M.; Michael, J. B.; and Shing, M.-T. 2005. Environment behavior models for scenario generation and testing automation. In *International Workshop on Advances in Model Based Testing (A-MOST 2005)*. St. Louis, Missouri, USA: ACM.

Ball, T., and Rajamani, S. K. 2002. The slam project: Debugging system software via static analysis. In *Annual ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL)*, 1–3.

Beizer, B. 1990. *Software Testing Techniques*. Van Nostrand Reinhold.

Bredeweg, B.; Bouwer, A.; Jellema, J.; Bertels, D.; Linnebank, F.; and Liem, J. 2006. Garp3 – a new workbench for qualitative reasoning and modelling. In *Proceedings of the 20th International Workshop on Qualitative Reasoning (QR-06)*.

Esser, M., and Struss, P. 2007. Fault-model-based test generation for embedded software. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, 342–347.

Forbus, K. D. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Fraser, G., and Wotawa, F. 2006a. Property relevant software testing with model-checkers. In *Proceedings of the Second International Workshop on Advances in Model-based Testing (A-MOST '06)*.

Fraser, G., and Wotawa, F. 2006b. Using model-checkers for mutation-based test-case generation, coverage analysis and specification analysis. In *Proceedings of the International Conference on Software Engineering Advances (ICSEA 2006)*.

Kuipers, B. 1986. Qualitative simulation. *Artificial Intelligence* 29:289–388.

Learning Qualitative Models through Partial Derivatives by Padé

Jure Žabkar and **Ivan Bratko** and **Janez Demšar**
AI Lab, Faculty of Computer and Information Science,
University of Ljubljana, SI-1000 Ljubljana, Slovenia

Abstract

Padé is a new method for learning qualitative models from observation data by computing partial derivatives from the data. Padé estimates partial derivatives of a target function from the learning data by splitting the attribute space into triangles or stars from Delaunay triangulation, or into tubes, and computing the linear interpolation or regression within these regions. Generalization is then accomplished by any attribute-value learning method. The methods for estimating partial derivatives differ regarding their resistance to noise, ability to handle noisy and missing values, computation speed and other properties. The experiments show these methods to be quite accurate, fast and robust. Being well integrated into our general machine learning and data mining suite Orange, Padé should also prove useful in practice.

Introduction

One of the goals of attribute-based machine learning is to explain the roles of individual attributes. An efficient way of achieving this in regression problems is to observe the change of function value corresponding to changes in individual attribute values. In mathematics, this is called partial derivative and has been — from its invention by Newton and Leibniz on — a most fundamental tool for describing relations in physics. In this paper we develop a machine learning method that computes partial derivatives and combines well established principles from mathematics and physics with the robustness and flexibility of typical machine learning algorithms.

In qualitative modeling, the task is often limited to only predicting the sign of the derivative and not its magnitude. In this paper we propose a set of methods with a common name Padé (an acronym for "partial derivative", and the name of a famous French mathematician). The methods assess qualitative or quantitative partial derivatives for points in the attribute space. We can then use machine learning algorithms to induce a predictive model, or venture into exploratory analysis and manually discover relations in the data.

Our goal was not only to design a fast, robust and conceptually clean algorithm with a small number of parameters, but also make it well integrated into our general ML platform Orange (Zupan, Leban, & Demšar 2004) with its substantial arsenal of machine learning and data mining techniques.

Learning method

We assume the following learning problem: the input data is a set of variable-value vectors, each consisting of attribute values and a class-value. The attributes normally correspond to independent variables in our problem space, and the class corresponds to a dependent variable. The task is to find a qualitative model that explains this data. The model may be a formal structure, such as a set of qualitative rules or a tree, or visual, for instance with a scatter plot. The model obtained from the data should enable predictions of the dependent variable value when given the values of the variables.

The class variable is continuous, and the attributes may be either continuous or discrete. In the context of learning models of physical systems, typically at least some of the attributes are real-valued. In our case, a qualitative model will consist of a set of qualitative proportionality constraints that will appear in if-then rules or decision trees. For example, let y be a function of x: $y = x^2$. The learning data would consist of a sample of pairs of values (x, y) where x is the attribute (independent variable) and y is the class (dependent variable). A correct qualitative model induced from this data would be:

$$\text{if } x > 0 \text{ then } y = Q(+x)$$
$$\text{if } x < 0 \text{ then } y = Q(-x)$$

The constraint $y = Q(+x)$ is read as y is qualitatively proportional to x. Roughly, this means that y increases with x. More precisely, in Padé this means

$$\frac{\partial y}{\partial x} > 0.$$

The notation $y = Q(-x)$ means that y is inversely qualitatively proportional to x (i.e. the partial derivative of y w.r.t. x is negative).

As another example that involves a discrete variable, we may state the qualitative relations between the price of a product, and the product's type and size:

$$\text{if } ProductType = car \text{ then } Price = Q(+ProductSize)$$
$$\text{if } ProductType = computer \text{ then } Price = Q(-ProductSize)$$

We will also be using an abbreviated notation when referring to several qualitative proportionalities. For example,

two constraints $z = Q(+x)$ and $z = Q(-y)$ will be abbreviated to $z = Q(+x, -y)$.

Using Padé for learning of qualitative models of this kind consists of three stages:

1. For all the given data points, use Padé to assess numerically the partial derivative of the class variable w.r.t. a chosen continuous attribute.
2. Perform a qualitative abstraction: Convert the computed numerical approximations of partial derivatives at all the data points into their signs. These signs are then used as discrete class values in the next step.
3. Use any attribute-value learning method (such as if-then rule learning, or decision tree learning) to produce a classifier that maps the points in the attribute space into qualitative proportionality constraints. This classifier is our qualitative model induced from the data.

Padé is not a single algorithm but a suite of methods performing this task: approximation of partial derivatives of a sampled unknown function f. The input for all the methods is a set of examples described by a list of attribute values and the value of a continuous dependent variable. An example of a function with two attributes is depicted in Fig. 1(a): each point represents a learning example and is assigned a continuous value. Our task is to compute a partial derivative at each given point. In our illustrations, we shall show the computation at point (5, 5), which is marked with a hollow symbol.

In the following paragraphs, we describe four novel methods of numerically estimating partial derivatives, which we use in Padé.

First Triangle method was the initial venture point of development of Padé. It models the function's behavior by dividing the attribute space into simplices (we shall refer to them as "triangles") by using the standard Delaunay triangulation as shown in Fig 1(b). If the samples are sufficiently dense the function's behavior within triangles is approximately linear.

Being able to compute the value of function in point $f(P + dx)$ with a simple interpolation between points P, A and B, First triangle method can apply the textbook definition of the partial derivative:

$$\frac{\partial f}{\partial x} = \frac{f(P + dx) - f(P)}{dx}.$$

Star Regression is based on the First triangle method, but improves its noise resistance by assuming the function's linearity across the entire star (a topological term for the set of triangles surrounding a point) instead of just a single triangle. It finds the value of the partial derivative that minimizes the square error, which translates into computing the univariate linear regression across the points of the star.

In the case shown in Fig. 1(c), the derivative would be computed as the coefficient of the linear regression on points A, B, C, D, E and F.

Triangles' Path method copes with more noise by smoothing the function more. To keep the computation focused, we do not simply widen the star but instead follow the triangles in the direction in which we compute the derivative (1(d)). The partial derivative is then again computed by minimizing the square error for the points lying on that path. In each shaded triangle, we choose an arbitrary point and assign it a function value by linear interpolation between triangle's vertices.

This method was actually never implemented as described here, but only as a simplified version of another, more complicated algorithm Qing, which includes many other improvements that will be published elsewhere.

Tube Regression is an approximation of the Triangles' Path method. It avoids computing the triangulation altogether, but instead considers a certain number of examples nearest to the axis in the direction in which we compute the derivative. These examples lie in a (hyper)tube which approximates (or, better, *mimics*) the Triangles' path (Fig. 1(e)).

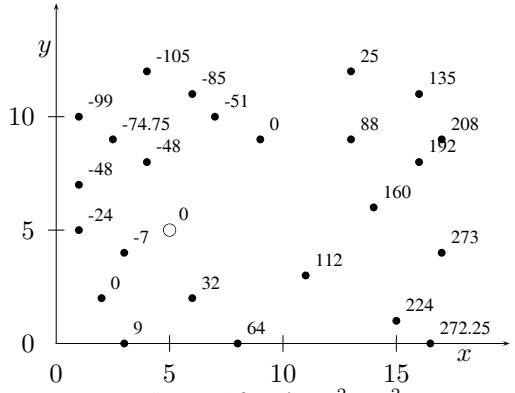
The tube can also contain points that lie quite far from the point P. To observe the local behavior of the function, Tube regression weights examples by their distances from P along the tube (that is, ignoring all dimensions but x). The method is thus similar to computing 1-dimensional LWR within the tube and taking the coefficients as partial derivatives.

All described methods are implemented as preprocessors, which get a sampled function, described by values of arguments and the function value, and return the corresponding numerical or qualitative partial derivative at each point. These derivative data can then be modeled with regression or classification trees (in the latter case one can model derivatives for each attribute separately or all attributes together) or by any other appropriate machine learning algorithm. It is usually even more interesting and useful to observe the data by visualizing it in scatter plots or other visualizations.

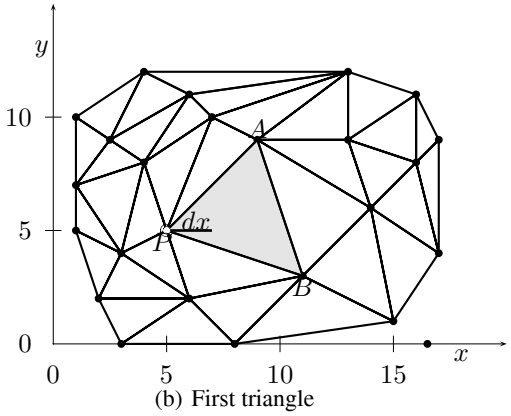
Experiments

The described methods were implemented within data mining and machine learning framework Orange (Zupan, Leban, & Demšar 2004), so they can be used with its huge arsenal of machine learning and visualization techniques.

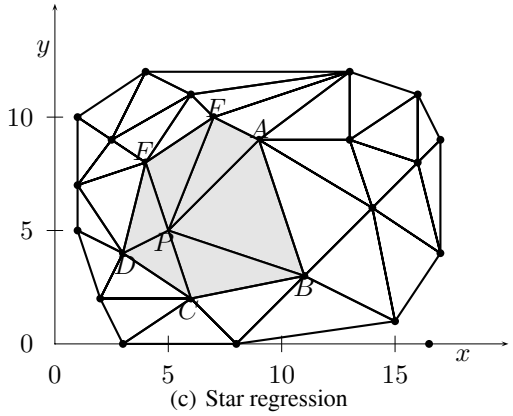
We will illustrate the interesting qualities and shortcomings of the algorithms with several experiments. We commence with inverted pendulum: we use a simple visualization that reveals the qualitative behavior of the function and also helps choosing a suitable modeling algorithm. We then show a simple artificial domain where the correct model depends on using a discrete attribute. We continue with another artificial example, $\sin(x)\sin(y)$ over a few periods, where the visualization turns out to be the only sensible "model". We then investigate Padé's ability to cope with noise, and conclude with an example with data from a 6th Framework European research project XPERO.



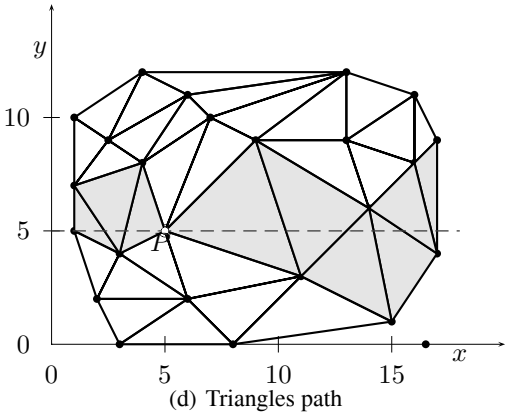
(a) Sampled function $x^2 - y^2$



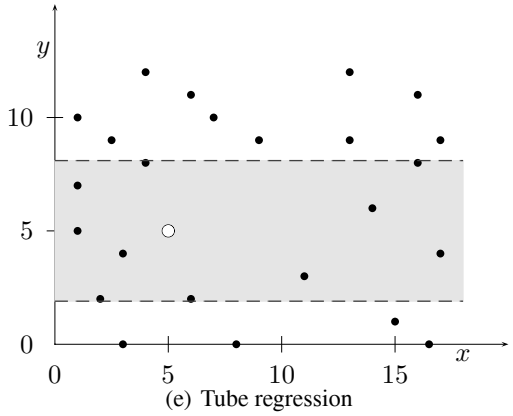
(b) First triangle



(c) Star regression



(d) Triangles path



(e) Tube regression

Figure 1: Illustration of Padé's methods

Inverted pendulum

The inverted pendulum is a well known dynamic domain often used in evaluation of algorithms for learning and control. The physical model is determined by equations for $\ddot{x}$ and $\ddot{\varphi}$.

$$\ddot{x} = \frac{4F + 2lm\dot{\varphi}^2 \sin\varphi - 1.5mg \sin 2\varphi}{4M + 4m - 3m\cos^2\varphi}$$

$$\ddot{\varphi} = \frac{(M+m)g\sin\varphi - F\cos\varphi - \frac{1}{2}ml\dot{\varphi}^2 \sin\varphi\cos\varphi}{\frac{1}{6}(4M + 4m - 3m\cos^2\varphi)l}$$

The variables x and φ are the horizontal position of the cart and the angle of the pole w.r.t. vertical axis. F is the horizontal force applied to the cart. The parameters M, m and l are the mass of the cart, the mass of the pole and the length of the pole. We here demonstrate our algorithm on the problem of modeling $\ddot{x}$, which is more difficult than the modeling of $\ddot{\varphi}$. We generated a set of 1000 examples by random sampling with parameters set to: $F = 0$ (free movement), $M = 1$ (mass of cart), $m = 0.1$ (mass of pole), $l = 1$ (length of pole). Independent variables are $\varphi \in [-\pi/2, \pi/2]$ and $\dot{\varphi} \in [-10, 10]$.

We approached the problem with the First triangle method because the data consists of only continuous attributes without any noisy or missing values. Its results can be nicely visualized with a scatter plot, which shows that $\ddot{x}$ is negative for smaller values of φ and $\dot{\varphi}$, and positive for the larger (Fig. 2a). The boundary between the two areas suggests the unsuitability of classification trees for modeling the domain. We instead used Orange's implementation of naive Bayesian classifier which uses LOESS for estimating the conditional probabilities for continuous attributes. Its visualization with the nomogram (Fig. 2b) shows that it can fit the boundary perfectly (see (Jakulin *et al.* 2005) for a detailed explanation of nomograms).

We obtained similar results with other Padé's methods, though they somewhat distorted the ellipse.

Discrete Attributes

We checked the Tube Regression's handling of discrete attributes with a function nastily defined as

$$\text{IF } s = 1 \text{ THEN } f = -x/10 \text{ ELSE } f = 10x.$$

Besides the continuous attribute x and Boolean attribute s, the data set also included an attribute r with random values and no influence on f. Variables x and r were from the same definition range, [-10, 10]. The function was sampled in 400 points.

Tube Regression, whose results we used to construct a classification tree using C4.5 (Quinlan 1993) included in Orange, found the correct solution (Fig. 3). We also tried other Padé's methods, which, as expected, mostly failed to recognize the role of s (which they were given as a continuous attribute). This confirms that replacing discrete attributes with dummy variables, like in statistical regression methods, will not work with triangulation-based Padé's methods.

Visualization

There are domains in which most machine learning algorithms fail to produce any meaningful results without a strong help from the expert. In such cases, using a good visualization is a much better choice than blindly inducing a model. Padé works with many visualization algorithms, from a simple distribution graph or scatter plot to state-of-the-art methods of intelligent visualization (Leban *et al.* 2006).

To illustrate such a domain, we generated a data set of 10000 sampled points for function $\sin(x)\sin(y)$, $x, y \in [-3\pi, 3\pi]$ (Fig. 4(a)). Such periodic functions of two variables are quite common in the real world. We computed derivatives by x; results for y are analogous.

The obvious candidate for this data is the First triangle method: there is no noise and all attributes are continuous. Knowing the complexity of the modeled function in advance, we can expect the noise reducing methods to almost certainly "oversimplify" the data.

First triangle (Fig. 4(b)) performed perfectly. The edges are perfectly sharp, which is due to the very high density of the samples. We checked that the algorithm still performs very well with 500-1000 samples, except for the edges which then evidently follow the individual sample points.

Star regression (Fig. 4(c)) exhibits some smearing at the corners, yet its results are still excellent and useful. Our suspicions that Tube regression (Fig. 4(d)) is unable to model this data were proved correct: it merged the left-most and the right-most two columns, and performed miserably in between.

Figures 4(e) and 4(f) visualize numerical derivatives.

Noise

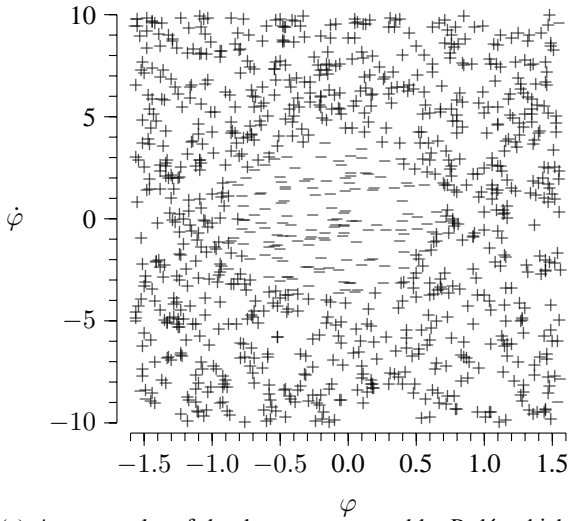
As an example of a very noisy function, we sampled the function $f(x, y) = x^2 - y^2$ on interval $[-100, 100] \times [-100, 100]$ to which we added uniform random noise of up to ± 2000. The data set consisted of 1000 random samples. Fig. 5 shows the intersection of the "noised" surface $x^2 - y^2$ with the plane $y = 0$ to illustrate the magnitude of the added noise around the point where the qualitative behavior of the function changes.

With such extreme noise, the method of choice is Tube regression. The assessed qualitative derivatives were used to induce a decision tree (Fig. 6). The induced models are correct and the split thresholds are quite accurate given the huge relative noise at around $x = 0$ and $y = 0$.

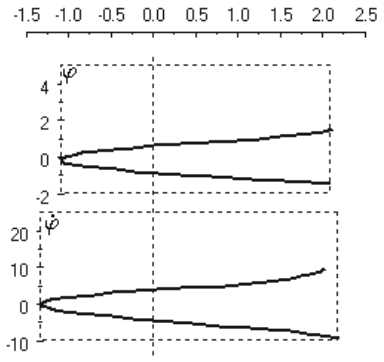
XPERO robot

For the final example, we used Padé on data from the ongoing European project XPERO (IST-29427). A simulated robot with a camera observes a ball. The task in this particular case was to discover the relation between the area of the ball in the picture, and the robot's angle and distance from the ball.

Figure 7 shows the corresponding trees. Padé performed perfectly regarding qualitative proportionality between distance and area: whenever the ball is visible (that is, the angle is approximately between -28.9 and +27.8 degrees), the area



(a) A scatter plot of the data set generated by Padé, which visualizes qualitative behaviour of $\ddot{x}$ with regards to φ.



(b) A naive Bayesian nomogram from Padé's data that models the qualitative behavior depicted in the scatter plot on the left.

Figure 2: A visualization and a qualitative model for inverted pendulum.

decreases with distance. Otherwise, is does not depend upon the distance (denoted by Q()).

The tree for the correspondence between the angle and the area correctly discovered – although with rather loose threshold values – that the area increases when the robot is turning towards the ball (that is, when the negative angle increases and when the positive angle decreases). When the entire ball is visible, the angle plays no role; the prevalent class with a slim majority is Q(-angle), yet the tree continues to split further and further. These further splits make no sense and, admittedly, the tree in Figure 7 was manually pruned at the middle leaf. For the angles in between, the area thus sometimes increase and sometimes decrease with the angle, which is the artifact of particular learning traces.

Discussion

All Padé's methods are fairly easy to understand and implement. They, however, differ in many important aspects.

Noise Handling

First triangle's beauty is in its pure use of concepts from topology and analysis. Its results on noiseless data are as good as the density of samples permit, while with increasing the noise level they soon degrade to useless. Noise canceling algorithms from topology are being added as a part of the Qing algorithm mentioned earlier.

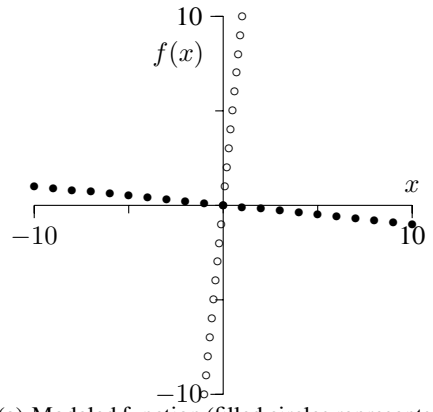
Tube regression, on the other side of the spectrum, is highly noise resistant, which will, as usual, also make it smear fine details in noiseless data. The actual degree of smoothing is in principle regulated with two arguments. The width of the tube should balance between having enough examples for a reliable estimation of the coefficient on one side and not observing the examples where the values of other attributes could significantly affect the function value (too much) on the other. However, if the tube is symmetrically covered by the examples (this is probably true except on the boundaries of the covered attribute space) and if the function which we model is negatively symmetrical with respect to other attributes' values in the part of the space covered by the tube,[1] impacts of other attributes can be expected to cancel out. Wide tubes therefore should not (and empirically do not) cause too much of a problem.

There is a similar balancing along dimension x: if the kernel function for the weight is too wide, the derivative will not be local enough, while a narrow kernel will not be resistant to noise. This dilemma is the same as in LWR, with the only difference that while LWR computes a function value, we here observe the regression coefficient.

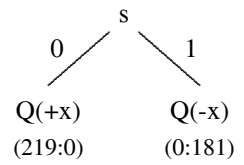
We experimentally observed that the method's parameters do not have considerable impact on the results and fixed the width of the tube to 30 points. Examples are weighted using a Gaussian kernel fitted so that the point farthest along the tube has a negligible coefficient of 0.001. The method is thus effectively without user-definable parameters.

Star Regression's resistance to noise is in between those of the First Triangle and the Tube Regression. We would also expect the Triangles' Path to be close to that.

[1] Formally, $f(\mathbf{x}+\mathbf{y})-f(\mathbf{x}) \approx f(\mathbf{x})-f(\mathbf{x}-\mathbf{y})$, where $\mathbf{x}$ is a point on the axis and $\mathbf{y}$ is a vector perpendicular to the axis and smaller than the tube's diameter. Linear functions, for instance, have this property, and most other functions we model are also locally linear enough.

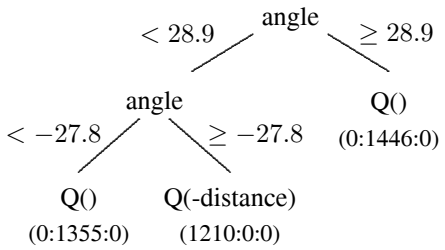


(a) Modeled function (filled circles represents points with $s = 1$, hollow circles represent points where $s = 0$)

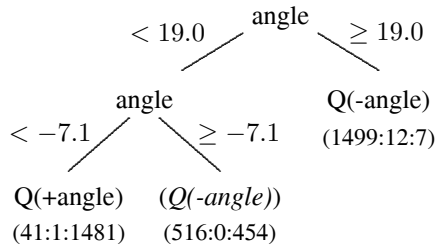


(b) Qualitative tree with Tube Regression and C4.5

Figure 3: Experiment with discrete attributes, function IF $s = 1$ THEN $f = -x/10$ ELSE $f = 10x$.



(a) Change of area w.r.t. distance



(b) Change of area w.r.t. angle (pruned)

Figure 7: Padé's modeling of XPERO data. Q() signifies negligible changes. Numbers in the leaves represent the number of examples with Q(-), Q() and Q(+), respectively.

Discrete Attributes and Unknown Values

The methods based on triangulation cannot handle unknown attribute values. Discrete attributes can be considered only if they are converted to dummy continuous variables, yet this gives awkward triangulations and meaningless results. Although not used in computation of derivatives, they can still be used in further processing of the data provided by Padé.

Tube regression can handle unknown values of attributes (except for the attribute on which we compute the derivative). This is, however, done through implicit imputation in distance computation procedure. Discrete values seem to pose no problems, as shown in experiments.

Total Derivatives

It may sometimes be interesting to observe the behavior of a function in a particular given direction not orthogonal to the coordinate axes. The adaptations of Padé's methods for that purpose are obvious. In First Triangle we align dx with the given direction, and in Tube regression we do the same with the tube. For Star Regression we can rotate the star in a similar fashion or, differently from the above tricks, compute multiple regression instead of univariate and treat the coefficients as a gradient. We then get the total derivative by multiplying the gradient with a (normalized) direction vector. None of these methods were implemented and evaluated yet.

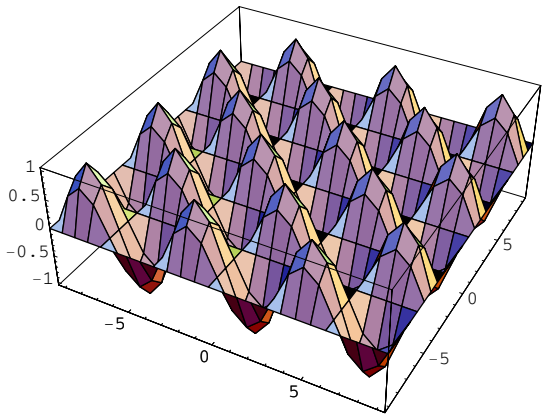
Time Complexity

For First triangle method, the most time consuming step is finding the triangle lying in the desired direction, which requires computing the determinant of a d-dimensional matrix (where d is the number of attributes). Such a triangle needs to be found for every point in space, for every attribute by which we compute the derivative. The running time strongly depends on the number of triangles that surround each point, which usually rises exponentially with the number of dimensions. In practice, the method is fast on low dimensional data and gets slower when the number of dimensions increases.

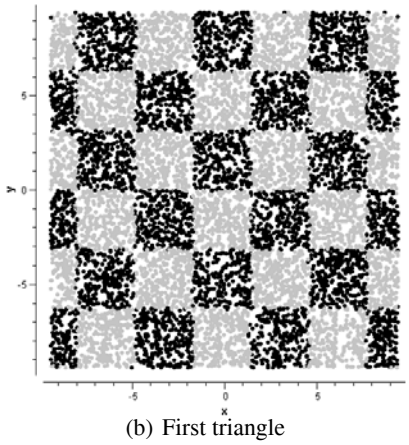
Tube regression's time complexity is linear in the number of dimensions and quadratic in the number of examples. It is consistently the slowest of all methods, except, possibly the Triangles' path, whose run time we have not measured.

Star regression always outran all other methods.

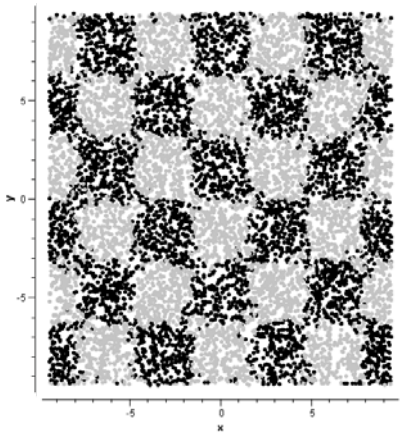
Table 1 sums up the running times of First triangle, Star regression and Tube regression for all experiments performed in the previous section.



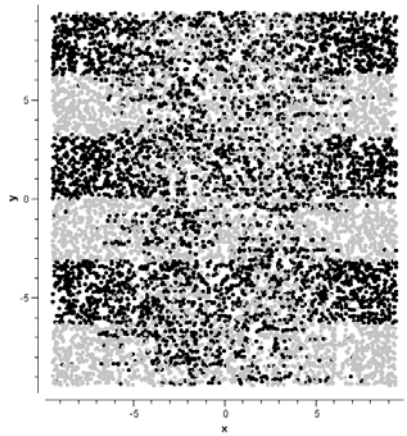
(a) Function $\sin x \sin y$



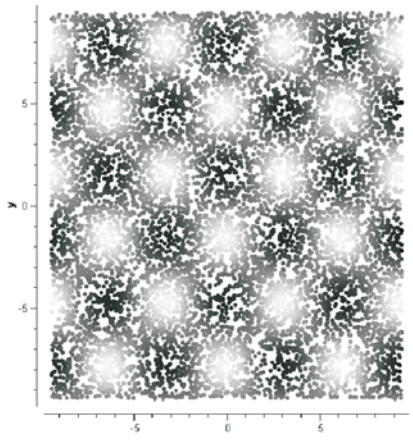
(b) First triangle



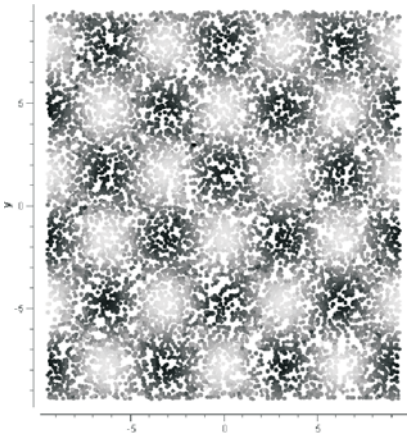
(c) Star regression



(d) Tube regression



(e) Numerical derivatives with First triangle



(f) Numerical derivatives with Star regression.

Figure 4: Function $\sin(x)\sin(y)$, its qualitative behavior and numeric derivatives by x of . Brighter and darker colors represent positive and negative derivatives. Color scales for graphs (e), (f) and other graphs are not comparable.

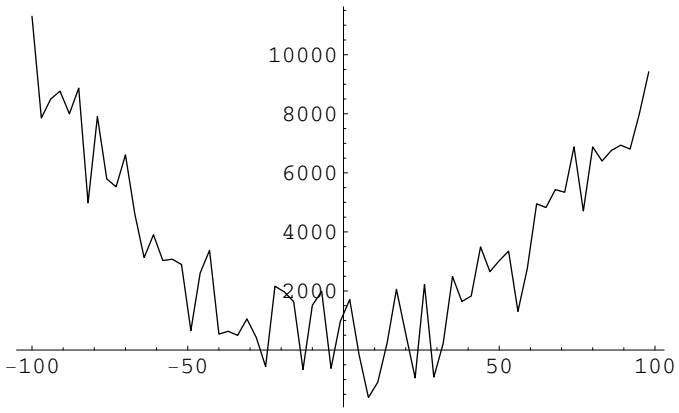


Figure 5: The intersection of the surface $x^2 - y^2$ with the plane $y = 0$ to illustrate the added noise.

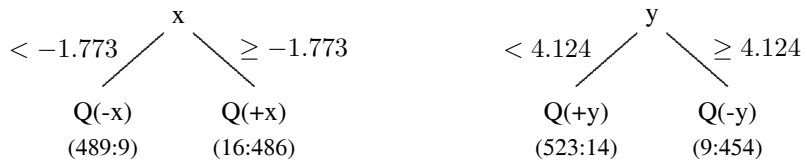


Figure 6: Qualitative models of function $x^2 - y^2$ with added random uniform noise. The data set contains 1000 randomly sampled examples.

	#attr	examples	Triangle	Star	Tube
pendulum	2	1,000	2	< 1	4
discrete	2	400	1	< 1	1
sine	3	10,000	35	8	496
noise	2	1,000	2	< 1	4
XPERO	2	4,011	15	3	80

Table 1: Run times (in seconds of CPU on a 2 GHz laptop) of First triangle, Star regression and Tube regression in the experiments.

Related Work

Many algorithms have, in one way or another, tackled the problem of qualitative model induction from observation data. Recently, Gerçeker and Say (Gerçeker & Say 2006) proposed algorithm LYQUID which fits polynomials to numerical data and use them to induce qualitative models. Other systems include QMN (Džeroski & Todorovski 1995), LAGRANGE (Džeroski & Todorovski 1993) and LAGRAMGE (Todorovski 2003). Other approaches that mostly induce models in the form of QDEs include GENMODEL for the induction of QSIM-type models (Hau & Coiera 1997), and SQUID (Kay, Rinner, & Kuipers 2000) which focuses on trends and extreme points in numerical data and use envelopes that bound the trajectories of variables.

An important difference between these algorithms and Padé is that Padé is essentially a preprocessor while other algorithms produce a model. Padé outputs a data set which can later be used by appropriate algorithms for induction of classification or regression models, or for visualization. So Padé in the context of learning qualitative models is of interest mainly in combination with other ML systems. To our knowledge, most other algorithms for learning qualitative models only handle numerical attributes, except QDE learners that take qualitative behaviors as input. In Padé, Tube regression can also use discrete attributes, whereas other methods are limited to continuous attributes. However, discrete attributes can be used by machine learning algorithms applied to Padé's output, which means that the final model can include discrete attributes.

We shall compare our work in more detail with the well-known algorithms QUIN and epQUIN (Šuc & Bratko 2001; Šuc 2003; Bratko & Šuc 2003). Examining the differences between Padé and QUIN will also be helpful for better understanding of the design of Padé itself.

The common property of Padé and QUIN (and, for that sake, any other algorithm for estimation of derivatives from sampled functions) is that they observe the local behavior of the function by summing up the information from sampled points in the vicinity of the point of interest. QUIN does this by comparing the attributes and class value at each pair of near data points, and constructs a vector of qualitative changes. These vectors are used to determine how well the learning data in various regions comply with possible qualitative constraints. epQUIN differs from QUIN by considering every pair of examples, not only near neighbors, but

weighting the evidence by the distance. The results are used to induce a qualitative tree, that is a decision tree with the qualitative constraints that fit well the corresponding data in the leaves.

The most obvious difference between Padé and QUIN is that Padé computes numerical derivatives, which can be (and in most of our experiments indeed were) later used qualitatively. QUIN, on the other hand, sums up the qualitative changes. While Padé estimates the magnitude of change, QUIN estimates the probabilities of various changes. These probabilities can be rather unreliable since they may be computed from small subsets of examples only.

There is an important difference in the definitions of qualitative proportionality constraints in Padé (denoted by Q), and monotonic qualitative constraints in QUIN (denoted by M). Padé's Q-constraints correspond to qualitative partial derivatives. QUIN's M-constraints, on the other hand, have a different definition (see (Šuc, Vladušič, & Bratko 2004)) illustrated by the following example. The M constraint $z = M^{+,-}(x, y)$ means: for all the points (x_1, y_1, z_1) and (x_2, y_2, z_2) in the region in which the constraint holds, we have: if $x_2 > x_1$ and $y_2 < y_1$ then $z_2 > z_1$. According to the continuous reification theorem (Šuc, Vladušič, & Bratko 2004), if x, y and z are continuous variables then if $z = M^{+,-}(x, y)$ holds then e.g. $z = M^{+}(x)$ cannot hold. This is obviously different from the Q-constraints. The difference comes from the fact that Padé only considers changes along the independent variables (which corresponds to partial derivatives), whereas QUIN considers changes in any direction (e.g. changes in both arguments x and y). This leads to a less apparent, yet crucial difference in the definition of vicinity in both systems.

A practical difference between the methods is that QUIN is implemented as a tree learning algorithm, while Padé is a data preprocessor which can be used with any learning or visualization algorithm. This is further simplified by Orange's versatile graphical interface for connecting various methods.

We noticed that QUIN is considerably slower than learning with Padé's and a typical chosen ML method, even when Padé is run with its slowest method - Tube Regression. It is though difficult to tell whether the difference comes from the algorithms themselves or only from their implementations.

Conclusion

We presented a novel method for learning qualitative models based on estimating partial derivatives from data. We developed an algorithm for estimation of partial derivatives from a sampled continuous function. The basic version of the algorithm, First Triangle, is based on splitting the attribute space into regions defined by Delaunay triangulation and the reasonable assumption that the function sample density is high enough to exhibit sufficiently linear behavior within the regions. The method is beautifully simple, but unfortunately unable to cope with any significant noise. To amend this, we developed several modifications of the method – Star Regression, Triangles' Path and Tube Regression. The methods are parameter-free, except for the threshold defining the negligible change if the numerical derivatives are transformed into qualitative changes. The only potential parameters would occur in Tube Regression, but since modifying them has no significant impact on the results, we – preferring simplicity over "tweakability" – froze the parameters and hid them from the user.

In experiments on a few artificial and semi-artificial data sets the algorithms behaved according to expectations, so we believe that they will also be useful in practice.

Padé has been implemented inside the general machine learning and data mining environment Orange, which can be freely downloaded (either as sources or in binary format for Windows or Linux) at http://www.ailab.si/orange.

Acknowledgment

This work was supported by the Slovenian research agency ARRS, and by the European project XPERO: Learning by Experimentation (IST-29427).

References

Bratko, I., and Šuc, D. 2003. Learning qualitative models. *AI Magazine* 24(4):107–119.

Džeroski, S., and Todorovski, L. 1995. Discovering dynamics: from inductive logic programming to machine discovery. *Journal of Intelligent Information Systems* 4:89–108.

Džeroski, S., and Todorovski, L. 1993. Discovering dynamics. In *International Conference on Machine Learning*, 97–103.

Gerçeker, R. K., and Say, A. C. C. 2006. Using polynomial approximations to discover qualitative models. In *In Proceedings of the 20th International Workshop on Qualitative Reasoning*.

Hau, D., and Coiera, E. 1997. Learning qualitative models of dynamic systems. *Machine Learning Journal* 26:177–211.

Jakulin, A.; Možina, M.; Demšar, J.; Bratko, I.; and Zupan, B. 2005. Nomograms for visualing support vector machines. In *KDD-2005: proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Kay, H.; Rinner, B.; and Kuipers, B. 2000. Semi-quantitative system identification. *Artificial Intelligence* 119:103–140.

Leban, G.; Zupan, B.; Vidmar, G.; and Bratko, I. 2006. Vizrank: Data visualization guided by machine learning. *Data Mining and Knowledge Discovery,* In press.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Todorovski, L. 2003. *Using Domain Knowledge for Automated Modeling of Dynamic Systems with Equation Discovery*. Ph.D. Dissertation, University of Ljubljana, Ljubljana, Slovenia.

Šuc, D., and Bratko, I. 2001. Induction of qualitative trees. In De Raedt, L., and Flach, P., eds., *Proceedings of the 12th European Conference on Machine Learning*, 442–453. Springer. Freiburg, Germany.

Šuc, D.; Vladušič, D.; and Bratko, I. 2004. Qualitatively faithful quantitative prediction. *Artificial Intelligence* 158(2):189–214.

Šuc, D. 2003. *Machine Reconstruction of Human Control Strategies*, volume 99 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, The Netherlands.

Zupan, B.; Leban, G.; and Demšar, J. 2004. *Orange: Widgets and visual programming, A White Paper*. Faculty of Computer and Information Science, Ljubljana, Slovenia.

Induction of qualitative models using discrete Morse theory

Jure Žabkar and **Gregor Jerše** and **Neža Mramor** and **Ivan Bratko**
Laboratory of Artificial Intelligence, Faculty of Computer Science, SI-1000 Ljubljana, Slovenia

Abstract

Qualitative models are often a useful abstraction of the physical world. Learning qualitative models from numerical data is a possible way to obtain such an abstraction. We present a new approach to induction of qualitative models from numerical data which is based on discrete Morse theory (DMT). Our algorithm QING (Qualitative INduction Generalized) has a firm theoretical background in computational topology. This makes it possible to extend the capabilities of state-of-the-art algorithms for qualitative modelling substantially. The output of QING is a labeled graph, which enables a visualisation of the qualitative model. Induced qualitative models can also be used for numerical regression by applying the Q^2 method. To illustrate the power of QING we present its application on an artificial function, add noise, and finally show how it performs on a dynamic domain such as inverted pendulum.

Introduction

Every day more and more data from real-life processes, such as measurements of weather variables, measured data from simulations, technological processes, chemical reactions etc is being recorded. Only a small subset of this data is later analyzed in hope to obtain the models that would imitate the processes from which the data was gathered. Such models enable the experts to run simulations and make predictions about something before it really happens. Numerical prediction and quantitative modelling, both suited for such a purpose, are common tasks in machine learning. Their quality is usually judged on the numerical accuracy they achieve on yet unseen data. It often happens that numerically accurate model fails to explain the underlying processes hidden in the data or the explanation is too complex. Recently, quantitative machine learning has been combined with qualitative learning in the method called Q^2 learning (Šuc, Vladušič, & Bratko 2004) which turned out to be very successful. In Q^2, a qualitative model is induced first and is later used to force the numerical model to be consistent with the induced qualitative constraints. This usually contributes to better accuracy of numerical predictions while qualitative models themselves are useful as comprehensible models that intuitively explain how the system works.

Qualitative models have been neglected for several reasons. Not only the induction of a qualitative model is a complex task but it is also not possible to estimate the true value of the induced model. How good is it? How does it compare to the model induced by another algorithm? There are several estimates for numerical models but none for qualitative models. More or less it is the matter of one's taste and habit when one decides which algorithm to use.

In this paper we present an algorithm QING (Qualitative INduction Generalized) which is based on discrete Morse theory (DMT) (Forman 2001) from the field of computational topology. We consider this powerful theoretical background in mathematics an advantage. Given a learning set of examples with numerical attributes and a numerical class variable, the goal of QING is to perform qualitative analysis of class variable w.r.t. attributes. The output of QING is a qualitative field (qfield), a set of critical points and a labeled qualitative graph (qgraph), which is a visualisation of the qualitative model. Detailed definitions of these terms are given in section 'Algorithm QING'. Induced qualitative models can also be used for numerical regression by applying the Q^2 method. The main difference between QING and other algorithms for induction of qualitative models is in attribute space partitioning. Unlike algorithms that split on attribute values (e.g. trees, rules), QING triangulates the space (domain) and constructs the qualitative field which for every learning example tells the directions of increasing/decreasing class. Doing so it finds all maxima, minima and saddles, so called *critical points*. One of the main features of QING is *canceling*, a direct way to handle noisy data. Another important advantage over state-of-the-art algortihms is that monotonic qualitative constraints are generalized so that most of the qualitative ambiguity is removed. This paper is mainly focused on the theoretical background of our approach that greatly contributes to many features of QING. However, we also present some experiments to show how QING works in practice, how it handles noise and how it compares to state-of-the-art algorithms for induction of qualitative models.

The most relevant of related work is algorithm QUIN which we briefly summarize in 'Related work'. Algorithm QING is described and accompanied with a simple example in section 'Algorithm QING'. In section 'QING with inverted pendulum' we aplly QING to the dynamic system of inverted pendulum. For mathematically oriented readers we summarize discrete Morse theory in section 'Discrete Morse theory'.

Related work

The problem of automatic induction of qualitative models has been addressed several times (Bratko & Šuc 2003; Kuipers 1994). In one way or another, most of the approaches use mainly background knowledge and not learning examples. The first algorithm for induction of qualitative trees from numerical data was QUIN.

QUIN (QUalitative INduction) looks for qualitative dependencies in numerical data and induces qualitative trees to express such dependencies. The induction process is similar to the induction of decision trees (Breiman *et al.* 1984; Quinlan 1992). In a qualitative tree the leaves are labeled with MQCs (monotonic qualitative constraints), a kind of monotonicity constraints that are widely used in the field of qualitative reasoning (Kuipers 1994).

An MQC is best described by an example, let's say $y = M^{+}(x)$. This says that y monotonically increases whenever x increases. In general, MQCs can have more than one argument, e.g. $z = M^{+,-}(x, y)$ says that z monotonically increases whenever x increases and z monotonically decreases when y increases. Each qualitative constraint in an MQC requires a strict increasing/decreasing dependency in its variable while keeping the other variables constant. Therefore, an MQC may be qualitatively ambiguous. Qualitative ambiguity occurs when the qualitative value of the constraint cannot be predicted (e.g. the qualitative change in $z = M^{+,-}(x, y)$ cannot be determined in the case of x and y both changing). The degree of fit between the data and an MQC is evaluated by two measures: *qualitative consistency* and *qualitative ambiguity*. Qualitative consistency of an MQC is the percentage of the learning examples that are qualitatively consistent with the MQC. Qualitative ambiguity is the percentage of examples for which the MQC allows ambiguous predictions.

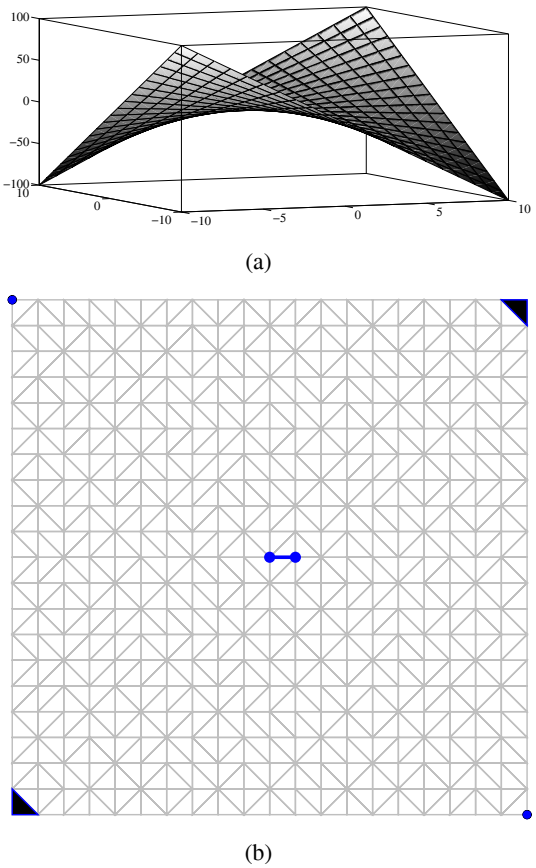
The QUIN algorithm has quite a high complexity. Empirical results (Bratko & Šuc 2003; Šuc, Vladušič, & Bratko 2004) show that QUIN can handle noisy data and, at least in simple domains, produces qualitative trees that correspond to human intuition.

Algorithm QING

QING's task is to perform qualitative analysis of continuous class variable f w.r.t. given attributes $(x_1, \ldots, x_n)$, where n is the dimension of the attribute space. For simplicity we will in this paper restrict ourselves to two attributes. Theoretically, QING works for any dimension n but is practical for $n \leq 5$ due to the complexity of triangulation. The input to QING is a set of learning examples with continuous attributes. Its output is:

- a qualitative field, (*qfield*)
- a set of *critical points* – minima, maxima and saddles of f, where in the case $n > 2$ the saddle are of different types,
- a qualitative graph, (*qgraph*)

Definition A qfield is a qualitative model represented as a set of pairs (p_i, p_j) which determine vectors pointing in the direction of incresing f. The points p_i in attribute space can be either data points or midpoints between the data points, i.e. centers of mass of the segments and triangles forming the triangultaion. An example of qfield is shown in Fig. 2.



(a)

(b)

Figure 1: Function $f(x, y) = xy$ and the triangulation of its domain with two minima (circles), two maxima (triangles) and a saddle in the middle.

The qfield determines the critical points of f. They are simply the points which do not appear in any one of the pairs (p_i, p_j).

Definition A qgraph is a labeled graph describing the qualitative behaviour of f. The vertices are in the critical points and two critical points are connected if a path along which the function values monotonically increase. It is an abstraction of qfield, ment as a visualization of the qualitative model. An example of qgraph is shown in Fig. 3.

To be more illustrative, the description of the algorithm is accompanied with an example $f(x, y) = xy$ defined on an orthogonal mesh (see Fig. 1(a)) on the domain $[-10, 10] \times [10, 10]$.

Before we continue, let us slightly extend the notation of an MQC: $f = M^{c}_{(x)}$ means that f stays constant with increasing x. We also note here that the specific qualitative ambiguity described in section 'Related work' is removed in QING – the values of all the variables may change simultaneously.

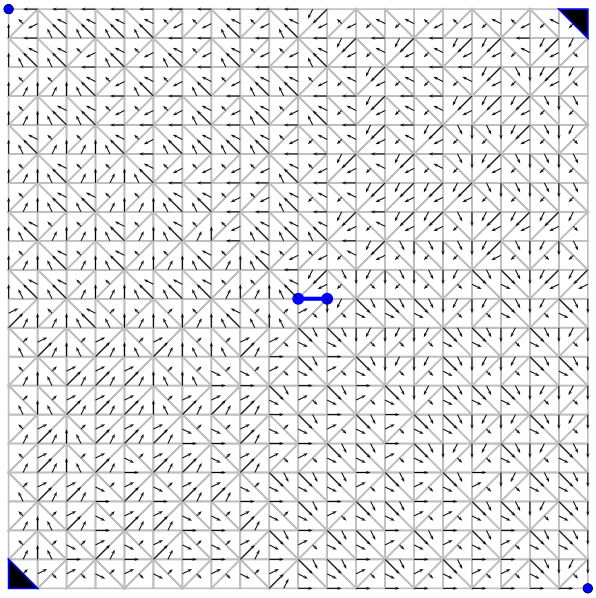


Figure 2: Qualitative field for $f(x, y) = xy$. The arrows point in the direction of function decrease.

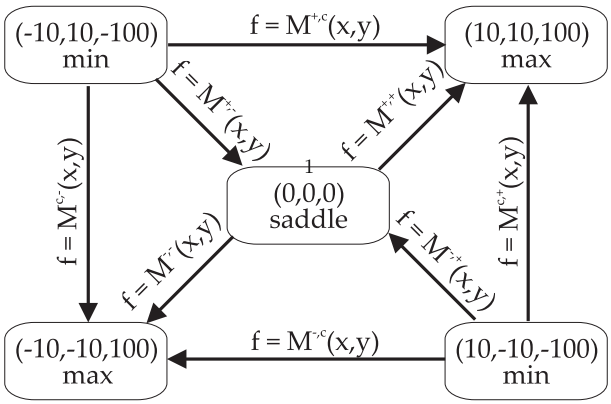


Figure 3: Qualitative graph for $f(x, y) = xy$.

The outline of the QING algorithm is as follows. Learning examples are represented as points in the attribute space, each point having assigned a value of its class variable. The domain is triangulated in order to be analysed with discrete Morse theory. Critical points are reconstructed using the algorithm of (King, Knudson, & Mramor Kosta 2005). Canceling is performed to remove the noise.

In the following paragraphs we explain each main step of the algorithm in more detail followed by examples. We finish this section with the analysis of QING's complexity.

Preprocessing

In the topological setting, learning examples are represented as points in $\mathbb{R}^n$, where n is the number of attributes, and the class variable f represents the values of a smooth Morse function in these points. In the case $n = 2$, a set of points $\{(x_i, y_i, z_i), i = 1, \ldots, k\}$ which represent sampled values of a function $z = f(x, y)$ over some domain $D \subset \mathbb{R}^2$ is given, and our goal is to analyse the function f using DMT to obtain a qualitative behaviour of f. To do so we first triangulate our domain D. The class values at these points are extended to a discrete Morse function defined on the triangles. In QING we use Delaunay triangulation implemented in a free software library Qhull (Barber, Dobkin, & Huhdanpaa 1996) which is very robust and works in arbitrary dimension. Since triangulation is the basis for further analysis it is worth using it carefully. Delaunay triangulation triangulates the convex hull of the given points causing some undesired effects on the edge, namely, triangles connecting distant points appear. To avoid this we embed our points in an artificial polygon, triangulate and remove the triangles that connect to the points on the polygon.

Obtaining qualitative model

To calculate the critical points of a function on a discrete set of points we use discrete Morse theory of Forman (Forman 2001). Critical points are reconstructed from the qualitative field which is obtained using the algorithm of (King, Knudson, & Mramor Kosta 2005). Possible pairs of critical points with function values differing by less than a given margin (parameter persistance) are cancelled. This becomes useful in noisy domains to set the threshold for noise reduction, where persistance is set to the value of the measuring tolerances at data acquisition.

Critical points together with the qfield represent a qualitative model of our function, the class variable. So described, the qualitative model could be used in Q^2 learning but it still lacks a comprehensive explanatory power. Especially in higher dimensions, it is too complex for a human to comprehend. Therefore we abstract the qualitative field to a qualitative graph which serves as a visualization tool.

Algorithm complexity

The algorithm consists of three major steps: constructing a triangulation and a discrete vector field on it, and constructng the qgraph connecting the critical points. In the first step, an additional feature is the possibility of cancelling neighboring pairs of critical points where the values differ

by less than a given margin, which is an efficient method for dealing with noise. The complexity of this first step is $O(h)$ without canceling, and $O(h^{2\times\lceil\frac{d}{2}\rceil})$ with cancelling, where h is the number of points and d is a dimension of the attribute space. The second step requires for each critical point a search through the paths leading through this critical point. The complexity of this step is $O(N)$, where N is the number of triangles. The last step requires a linear search through the points and therefore has the complexity of $O(h)$ where h is a number of learning examples (i.e. points).

How QING handles noise

Noise is disturbing but inevitable in real data. Therefore it is very important that the algorithm is able to deal with it and still induce a usefull model. QING has a straightforward solution to this problem. Its only parameter, *persistence*, cancels the pairs of critical points that differ in function values for less than the persistence.

To demonstrate canceling in practise we added 10% noise to our artificial domain $f(x, y) = xy$. Fig. 4 shows how different values of the parameter persistence influence the qualitative field. Starting with persistence 0, which corresponds to assuming that there is no noise, we encounter many critical points in the qfield. Increasing persistence we finally come to the point where the qgraph very much resembles the one on Fig. 3 with no noise. Both qgraphs are isomorphic, i.e. qualitatively equal. Inspite of noise we managed to discover the correct qualitative model. In practice, domain experts can usually asses the persistence value (e.g. the measuring tolerances) very well.

QING with inverted pendulum

The inverted pendulum (also known as 'pole and cart') is a well known dynamic domain that is, due to its simplicity, often used in experimenting with new algorithms. The system is shematically shown in Fig. 5. Equations 1 and 2 give its physical model. To build a qualitative model of the inverted pendulum we would have to model both equations. Since the procedure is the same, we choose to present only the more complex half of the qualitative model, $\ddot{x}$, and omit $\ddot{\varphi}$.

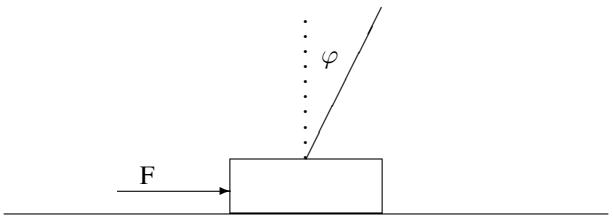


Figure 5: Inverted pendulum, also known as pole and cart.

$$\ddot{x} = \frac{4F + 2lm\dot{\varphi}^2 \sin\varphi - 1.5mg \sin 2\varphi}{4M + 4m - 3m\cos^2\varphi} \quad (1)$$

$$\ddot{\varphi} = \frac{(M+m)g\sin\varphi - F\cos\varphi - \frac{1}{2}ml\dot{\varphi}^2 \sin\varphi\cos\varphi}{\frac{1}{6}(4M + 4m - 3m\cos^2\varphi)l} \quad (2)$$

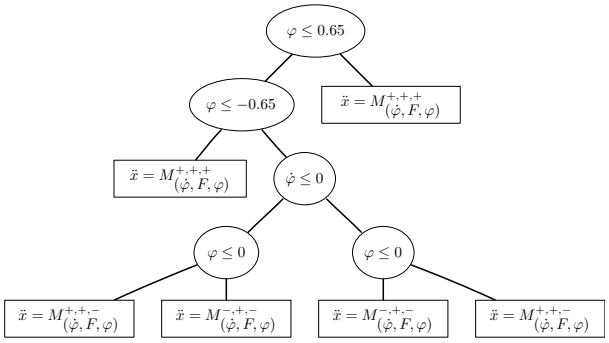


Figure 6: Qualitative tree for $\ddot{x} = \ddot{x}(\varphi, \dot{\varphi})$ built analytically from Eq. 1.

Since the equations are known, a straightforward way to obtain the qualitative model, would be to calculate the derivatives $\frac{\partial\ddot{x}}{\partial F}$, $\frac{\partial\ddot{x}}{\partial\varphi}$ and $\frac{\partial\ddot{x}}{\partial\dot{\varphi}}$ and look for the areas where they are positive/negative. By hand, with some approximations, we can get the qualitative tree shown in Fig. 6. Approximations are necessary because the area in $\mathbb{R}^2$ where $\frac{\partial\ddot{x}}{\partial\varphi}$ is close to 0 is an ellipse and using a qualitative tree, we can only approximate it with a rectangle.

Analytical solutions are nice to play with but in practise we often have only data, obtained by a sampling some process. For the sake of experiment, we use Eq.1 to obtain a data sample. Without loss, we neglect F. Our domain is therefore a plane spanned by φ and $\dot{\varphi}$, specifically, a rectangle $[-\pi/2, \pi/2] \times [-10, 10]$. To keep things simple we again have an orthogonal mesh and no noise.

On this data we use QUIN to construct a qualitative tree of depth 6 with 27 nodes, of which 14 are leaves – 7 $M^-(\dot{\varphi})$ and 7 $M^+(\dot{\varphi})$. The root splits on $\dot{\varphi} \leq -0.5$. All internal splits are made on different values of φ. As QUIN says, the coverage is perfect and there is no qualitative ambiguity in this tree. We can of course tell QUIN to build a smaller tree. The one of depth 3 has 8 leaves and its splits are the same as those to the third level in the larger tree.

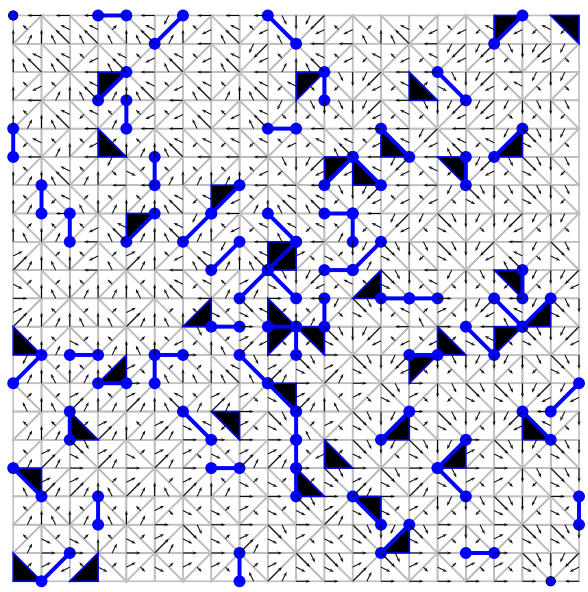
At the end, we use QING on the same data. The induced qualitative graph, Fig. 7, has 8 nodes (critical points) and 15 segments (MQCs) between them.

Technically speaking all three models are graphs so we can compare them simply by looking at their complexity.

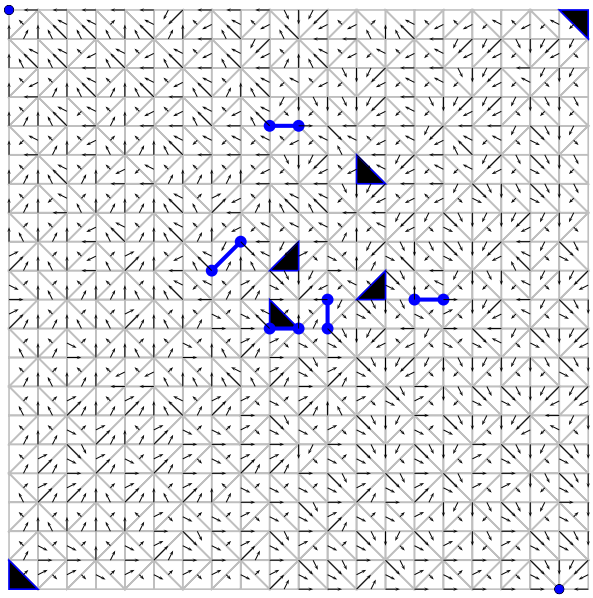
Discrete Morse theory

In this section we review the basics of Forman's discrete version of Morse Theory (Forman 2001).

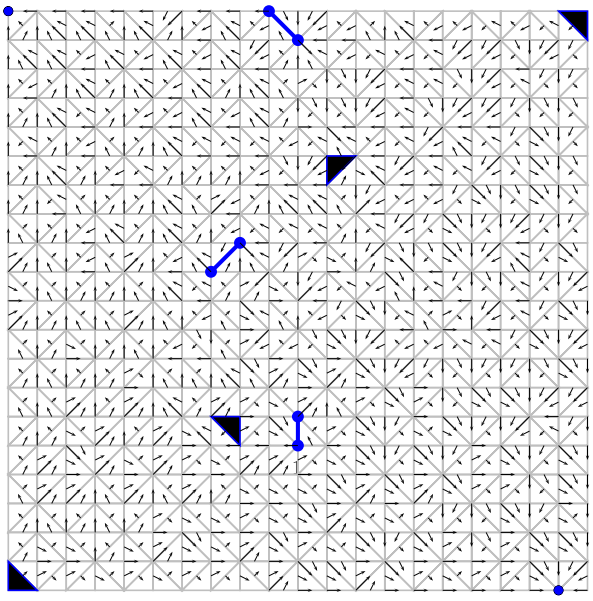
In the classical, smooth version of Morse Theory, a Morse function is a function defined on a smooth manifold M of dimension n, which has only nondegenerate critical points. In our case, M will be a domain in Euclidean space $\mathbb{R}^n$, and in this case a critical point p of a function $f : M \to \mathbb{R}$, is a point where $\mathrm{grad}\, f = 0$, i.e. the linear term in the Taylor expansion of f around p is 0. A critical point is nondegenerate if the second degree term in the Taylor expansion is nonzero. In the neighbourhood of a nondegenerate critical point, the



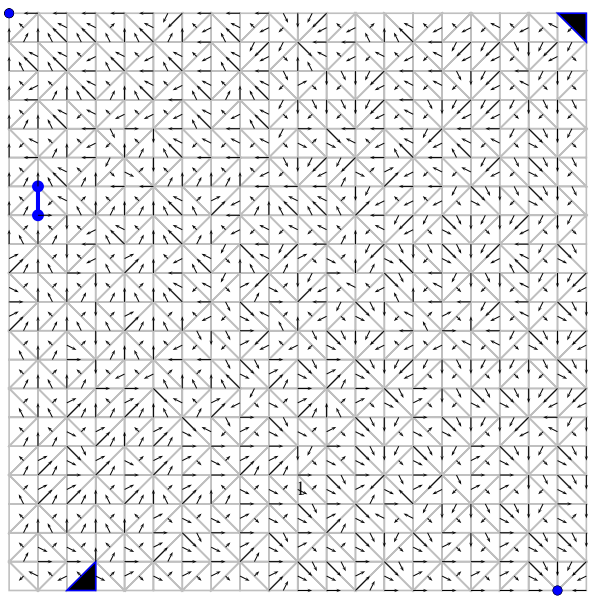
(a) persistence $= 0$



(b) persistence $= 10$



(c) persistence $= 15$



(d) persistence $= 23$

Figure 4: Domain $f(x, y) = xy$ with added 10% noise. Different values of parameter persistence are used to show how noise is removed through canceling of critical points.

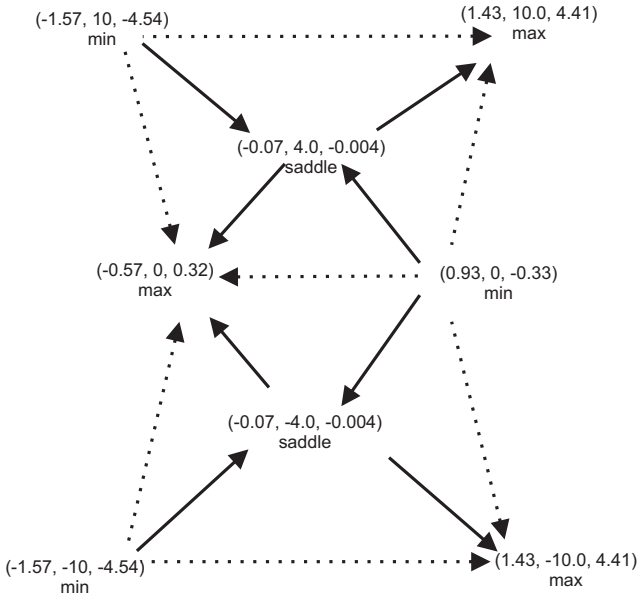


Figure 7: Qualitative graph for $\ddot{x} = \ddot{x}(\varphi, \dot{\varphi})$ built by QING.

function f can be expressed as $-\sum_{i=1}^{k} u_i^2 + \sum_{i=k+1}^{n} u_i^2$, where the number of negative terms k is called the *index* of p. In the case $n = 2$, a critical point of index 0 corresponds to a minimum, a critical point of index 1 to a saddle, and a critical point of index 2 to a maximum. In higher dimensions, saddles of different types exist. A nondegenerate Morse function determines a flow on the manifold M which corresponds to the vector field $\mathrm{grad}\, f$. A good introduction to Morse theory is (Milnor 1963).

In the discrete version of Morse theory, a triangulation of the domain M is given. A discrete Morse function f associates a value to each simplex in the triangulation, and satisfies the following conditions. For each simplex α there is at most one simplex $\beta^{(k+1)}$ which contains α as a face such that $f(\beta) \leq f(\alpha)$, and there is also at most one faces $\gamma^{(k-1)}$ of α such that $f(\gamma) \geq f(\alpha)$.

As we can see from these two conditions, the values of a Morse function generally increase with dimension, with one possible exception. It is easy to see that the two conditions above are exclusive, and so each simplex appears in at most one pair $(\alpha^{(k)}, \beta^{(k+1)})$, where α is a face of β and $f(\beta) < f(\alpha)$. A simplex $\alpha^{(k)}$ is a *critical simplex of index* k, if it does not appear in any such pair, i.e. if the function values on all its faces are lower, and the function values on all simplexes which contain it as a face are higher.

The collection of pairs $F = \{(\alpha^{(k)}, \beta^{(k+1)})\}$ with α_k face of β_{k+1} and $f(\beta) \leq f(\alpha)$ is the discrete analogue of the gradient vector field of a smooth function f. The discrete analogue of a trajectory of the gradient vector field is a V-path which is a sequence of simplices

$$\alpha_0^{(k)}, \beta_0^{(k+1)}, \alpha_1^{(k)}, \beta_1^{(k+1)}, ..., \beta_r^{(k+1)}, \alpha_{r+1}^{(k)}$$

such that pair $(\alpha_i, \beta_i) \in F$, for each $i = 0, 1, .., r$, $\alpha_i \neq \alpha_{i+1}$ and α_{i+1} (as well as α_i) is a face of β_i. Then $f(\beta_i) < f(\alpha_i)$ because (α_i, β_i) belongs to F and $f(\alpha_{i+1}) < f(\beta_i)$ because α_{i+1} is a face of β_i (but (α_{i+1}, β_i) does not belong to F). A V-path corresponds to a path through the simplices in M along which f decreases.

A discrete gradient vector which has no nontrivial closed paths, i.e. no V-paths such that $r \geq 0$ and $\alpha_0 = \alpha_{r+1}$ corresponds to a discrete Morse function (Forman 2001). So if we want to extend a function given on set of vertices to a discrete Morse function on the entire triangulation, we only have to find a discrete vector field that has no nontrivial closed paths (King, Knudson, & Mramor Kosta 2005).

Conclusions and further work

We applied the discrete Morse theory, which is a 'hot issue' in the field of computational topology, to qualitative machine learning. We used it to induce a qualitative model from numerical data. Qualitative rules are used to describe the qualitative constraints of class variable using given attributes. We focused mainly on the theoretical issues yet showing how QING performs in practise. We are aware of the fact that QING's true power should be tested on real domains but still believe that all the theoretical background should be carefully considered first.

References

Barber, C. B.; Dobkin, D. P.; and Huhdanpaa, H. 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software* 22(4):469–483.

Bratko, I., and Šuc, D. 2003. Learning qualitative models. *AI Magazine* 24(4):107–119.

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Trees*. CRC Press.

Forman, R. 2001. A user's guide to discrete morse theory. In *Proceedings of the 2001 International Conference on Formal Power Series and Algebraic Combinatorics, A special volume of Advances in Applied Mathematics.*

King, H. C.; Knudson, K.; and Mramor Kosta, N. 2005. Generating discrete morse functions from point data. *Exp. math.* 14(4):435–444.

Kuipers, B. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Massachusetts.

Milnor, J. 1963. *Morse Theory*. Princeton University Press.

Quinlan, J. 1992. Learning with continuous classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, 343–348.

Šuc, D.; Vladušič, D.; and Bratko, I. 2004. Qualitatively faithful quantitative prediction. *Artificial Intelligence* 158(2):189–214.

Qualitatively Constrained Equation Discovery

Jure Žabkar and **Aleksander Sadikov** and **Ivan Bratko** and **Janez Demšar**
Artificial Intelligence Laboratory, Faculty of Computer and Information Science,
University of Ljubljana, SI-1000 Ljubljana, Slovenia

Abstract

Equation discovery is a very lively area of artificial intelligence which deals with explaining phenomena by mathematical formulae induced from the data. One successful approach to the problem are algorithms which construct thousands of formulae and report the simplest ones with the best fit to the data. Another, sub-symbolic, fits (piecewise) regression hyper-planes; their advantage is that they may be made to conform to qualitative constraints. We propose an algorithm that shares the qualities of the two approaches: EDGAR searches for simple qualitatively faithful equations which fit the data well. The algorithm performs very well on simple problems, but in its current implementation fails to solve more complex ones.

Introduction

The field of equation discovery can be defined as "given a set of (numerical) observations, find a set of laws, expressed as mathematical equations, which govern the observed system". An amazing example of such a venture is Kepler's use of Brahe's data to discover the rules of planetary motion. The task is far easier if the researcher knows what he is looking for, that is, if he wants to discover the relation between a set of independent variables and a dependent variable.

The physicist's approach is to derive a new law from the known laws. For instance, the motion of planets is a direct consequence of Newton's universal laws of gravity. This will fail when the laws are not there yet (like they were not in Kepler's time) or, as is more often the case nowadays, if the domain is too complex, non-linear, or has too many variables to be analytically solvable. A typical example of such a problem is modeling weather. In such cases, the expert may be able to come up with an approximate model where the constants are fit to the existing data by applying statistical methods like minimization of squared error. When the domain is not understood well enough, even this may be unfeasible. Some well known examples of this kind occur in ecological modeling (Langley *et al.* 2002; Kompare, Todorovski, & Džeroski 2001).

Machine learning and statistics offer two alternatives. One is to generate numerical models, such as piecewise linear regression or LOESS (Cleveland, Devlin, & Grosse 1998). The property of this approach, which is of particular interest for our paper, is that it can be implemented to also conform to qualitative constraints (Šuc, Vladušič, & Bratko 2004) given by an expert or by algorithms like QUIN (Bratko & Šuc 2003) or Padé (Žabkar, Bratko, & Demšar 2007). These models may be very accurate, but they are useful only for *predicting* and not *explaining* the domain and so fail to fulfill our goal of finding "a set of laws governing the system".

The alternative, symbolic models are better in this respect. Algorithms like Goldhorn (Križman, Džeroski, & Kompare 1995) and Lagramge (Todorovski & Džeroski 1997) produce a number of equations with missing constants, fit the constants to the data and rank the equations by their simplicity and fit. For better control, they may also allow the expert to define a grammar for the equations (Todorovski & Džeroski 1997; Langley *et al.* 2002).

The problem with symbolic models is their ignorance of qualitative constraints, which can lead to meaningless results. For a simple test we modeled the free fall acceleration at different distances from the Earth. The correct equation (if the experiment is done above the Earth surface) is

$$g = G\frac{M}{r^2} = \frac{3.99 \times 10^{14}}{r^2}$$

where G is the gravitational constant, M is the Earth's mass and r is the distance from the Earth's center at which we measure the acceleration.

We generated artificial experimental data by sampling the function $g(r)$ with step 200 in the interval $[6371, 39971]$ (from the ground to the height of satellites) obtaining 169 samples. We added Gaussian noise with $N(0, 0.5)$. We tried to reconstruct the formula as a linear combination of terms obtained by generating all subsets of elementary functions

$$\{1, r, r^{-1}, r^2, r^{-2}, r^3, r^{-3}, \sin r, \log r, \cos r, \exp r\},$$

i.e. we were fitting the coefficients of functions like $a + br^2 + \sin r$ and $a \log r + br^{-2}$. We sorted the functions using the state-of-the-art combination of root mean squared error (RMSE) and minimum description length (MDL) measures from (Todorovski & Džeroski 1997). The optimal fit was a constant function, and the second best fit was (RMSE=0.5013):

$$g(r) = \frac{3.928 \cdot 10^{14}}{r^2} - 0.124\cos(r).$$

The first term is quite correct, while the second term only fits the (random) noise. The problem with this solution is that it suggests that free fall acceleration oscillates with r — which we (today) know is not true. The obvious remedy to this problem is to exclude the sine and cosine from the list of base functions. We can also tune the scoring function's bias on description length, but this can only be done if we know the correct formula in advance. Besides, the emphasis on MDL may already be too high, as witnessed by the fact that the best ranked function is simply a constant.

In this paper we propose a new algorithm, EDGAR, that offers a third approach, combining the advantages of numeric and symbolic approaches: it searches for symbolic equations by fitting the template functions constructed as a combination of terms (like in the example above) or from a grammar given by the expert, but at the same time also ensures that the solutions match the prescribed qualitative constraints.

Algorithm EDGAR

EDGAR (Equation Discovery with Grammars And Regression) is an algorithm for discovery of equations from a set of measurements of independent and dependent variables, a set of qualitative constraints, and the grammar specifying the templates of equations. The constraints may also specify a region, like in "y increases with x for all positive values of x". The algorithm consists of the following four steps.

1. Use a function generator to generate general forms of functions (templates). For instance, $a + bx + cx^2$ is a template for second degree polynomials in x.
2. Compute a symbolic derivative of each generated function, *e.g.*
$$\frac{\partial(a + bx + cx^2)}{\partial x} = b + 2cx.$$
3. Symbolically solve the system that puts the constraints on the coefficients of the initial function, respecting the qualitative constraint. For instance, if we know (from an expert or a qualitative model) that the function increases with x for all positive x, the algorithm needs to find the values of b and c which satisfy
$$\forall x, x > 0 : b + 2cx > 0.$$
The solution is:
$$(b = 0 \wedge c > 0) \vee (b > 0 \wedge c \geq 0).$$
4. Finally, fit the coefficients of the function to minimize RMSE, with respect to the constraints on the coefficients that were computed in the previous step to guarantee that the induced function will satisfy the given qualitative constraints. For instance, the algorithm would find the values of a, b and c within $(b = 0 \wedge c > 0) \vee (b > 0 \wedge c \geq 0)$, for which $a + bx + cx^2$ fits the data as close as possible.

For the first step, the algorithm currently supports two forms of specifying the function templates. One is to provide a set of elementary (basic) functions from which we can automatically generate candidate functions for further processing, like we did in the example in the introduction. For instance $\{1, x, x^2\}$ is used to generate all possible second degree polynomials. The alternative is to use context free grammars to generate candidate functions. This approach has several advantages over the first one, among them offering a simple way for the user to provide background knowledge and the use of declarative bias (Todorovski & Džeroski 1997).

The second step, computing the symbolic derivative of the function from the previous step, is trivial.

The overall simplicity of the idea is unfortunately spoiled by the extremely difficult realization of the third step. Its task translates to the problem of quantifier elimination and is generally insolvable. We used the state-of-the-art algorithms coded in Mathematica's (Wolfram Research, Inc. 2005) function Reduce. For polynomials, it uses cylindrical algebraic decomposition (Collins 1975). Algebraic functions are translated into equivalent purely polynomial systems. For transcendental functions, Reduce generates polynomial systems composed with transcendental conditions, then reduces these using functional relations and a database of inverse image information. Piecewise functions are symbolically expanded to construct a collection of continuous systems. The user can also help by adding some background knowledge into the logical formula.

The remaining step, minimization of RMSE given the constraints from the previous step, is generally a nonlinear constraint satisfaction problem, which we solve using Nelder-Mead methods (Luersen & Le Riche 2002).

The first step of the algorithm was partially implemented in Prolog. Everything else was implemented in Mathematica, which already contains the derivation, methods for quantifier elimination, and nonlinear minimization.

Experiments and Discussion

We tried the algorithm on the problem of modeling the gravitational acceleration with artificial data generated as described in the introduction. The Gaussian noise was again N(0, 0.5). We generated the function templates with a grammar that can induce symbolic rational functions up to the second order, e.g.: $ax^2 + bx\sin(c + dx)$, or $ax/[\sin(b + cx) - dx]$. The sine terms were included only for the sake of comparison, although it was obvious that all functions with such terms would be discarded in the third step. As a qualitative constraint, we told EDGAR that the gravitation decreases with the distance, $g = Q(-r)$.

The generated function with the optimal RMSE was
$$g(r) = -0.0259 + \frac{4.096 \cdot 10^{14}}{r^2}$$
with a RMSE of 0.4968.

Acting as domain experts, we noted that the formula, despite obeying the given qualitative constraints, still made no physical sense, since the negative term reverses the sense of gravitation for distances above 125,000 kilometers.

EDGAR makes it easy to add new constraints. We thus additionally stated that $g(r)$ should always be positive, which reported
$$g(r) = \frac{4.070 \cdot 10^{14}}{r^2}$$

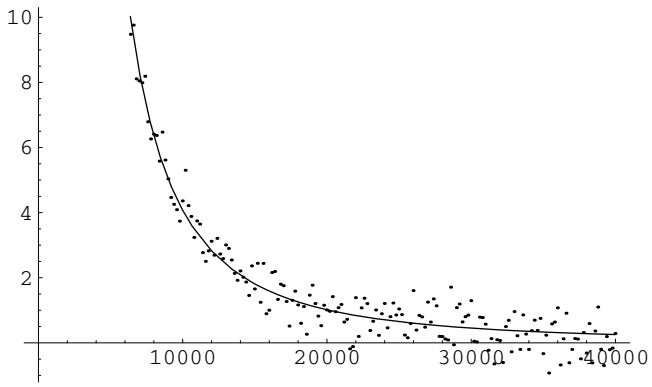


Figure 1: Best fit by EDGAR with enforced $Q(-r)$ and $\forall r : g(r) > 0$.

as the best ranked function with a RMSE of 0.4972 (see Fig. 1). This function is correct, except for the 3.6% error in the constant due to the noise.

We repeated the experiment with different amounts of noise: N(0, 1) and N(0, 0.2). EDGAR's results were the same (correct) as in the experiment with N(0, 0.5), except for the constant slightly varying due to different amounts of noise in the data. On the other hand, RMSE alone always selected an overly complex overfitted function, and adding MDL to the scoring function resulted in always preferring a constant as a solution.

Yet, despite this success — and a few others on similarly simple domains, for instance on the XPERO robot data described in (Žabkar, Bratko, & Demšar 2007) — there remains a lot of further work to make the algorithm practically useful. We describe the problems and our proposed solutions below.

Depending on the complexity of the templates (or, more accurately, their derivatives) the task of the third step may be too complex. In the current implementation, this would result in a suboptimal, yet still qualitatively faithful solution. We are working on replacing the Reduce function with probabilistic alternatives.

When the solution includes periodic functions, these can generate a lot of local minima, which the minimization procedure can fall into. We do not yet know whether this will cause any real problems and whether restarting the minimization from different initial points will amend them.

The algorithm needs a few minutes on an average PC for solving rather simple problems (gravitational acceleration, XPERO robot data) and does not seem to scale well. This is again due to the complexity of the Reduce function. Besides replacing it, the algorithm can also be accelerated by using exact or heuristic methods to eliminate as many functions as possible before they reach the third step of the algorithm.

Conclusion

We described an algorithm called EDGAR which discovers symbolic equations that fit the given data as well as possible and, at the same time, match the given qualitative constraints. The algorithm is conceptually simple and was easy to implement using the existing functions for derivation, quantifier elimination and minimization available in Mathematica. The successful tests on a few simple domains show the algorithm as promising, yet there remain quite a few technical problems to be solved before it will also be practically useful.

Acknowledgments

This work was supported by the Slovenian research agency ARRS, and by the European project XPERO: Learning by Experimentation (IST-29427).

References

Bratko, I., and Šuc, D. 2003. Learning qualitative models. *AI Magazine* 24(4):107–119.

Cleveland, W.; Devlin, S.; and Grosse, E. 1998. Regression by local fitting. *Journal of Econometrics* 37:87–114.

Collins, G. E. 1975. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. *In Lecture Notes In Computer Science* 33:134–183.

Kompare, B.; Todorovski, L.; and Džeroski, S. 2001. Modeling and prediction of phytoplankton growth with equation discovery : Case study - Lake Glumsø, Denmark. *Verh. - Int. Ver. Theor. Angew. Limnol.* 27:3626–3631.

Križman, V.; Džeroski, S.; and Kompare, B. 1995. Discovering dynamics from measured data. *Electrotechnical Review* 62:191–198.

Langley, P.; Sanchez, J.; Todorovski, L.; and Džeroski, S. 2002. Inducing process models from continuous data. In *Proceedings of The Nineteenth International Conference on Machine Learning*.

Luersen, M. A., and Le Riche, R. 2002. Globalized Nelder-Mead method for engineering optimization. In *ICECT'03: Proceedings of the third international conference on Engineering computational technology*, 165–166. Edinburgh, UK: Civil-Comp press.

Todorovski, L., and Džeroski, S. 1997. Declarative bias in equation discovery. In *In Proceedings of the 14th International Conference on Machine Learning*.

Šuc, D.; Vladušič, D.; and Bratko, I. 2004. Qualitatively faithful quantitative prediction. *Artificial Intelligence* 158(2):189–214.

Žabkar, J.; Bratko, I.; and Demšar, J. 2007. Learning qualitative models through partial derivatives by Padé. In *Proceedings of the 21st International Workshop on Qualitative Reasoning (submitted)*.

Wolfram Research, Inc. 2005. *Mathematica, Version 5.2*. Champaign, IL, USA.

The riverine landscape Kamp (Austria): an integrative case study for qualitative modeling of sustainable development

Andreas Zitek, Susanne Muhar, Sabine Preis & Stefan Schmutz

Institute of Hydrobiology and Aquatic Ecosystem Management, University of Applied Life Sciences, Vienna;
Andreas.Zitek@boku.ac.at

Abstract

This paper presents basic features for modeling some important aspects of sustainable development of the riverine landscape Kamp. We used the QR ontology to collect and organize expert knowledge on ecological effects of water abstraction on fish and the integration of stakeholder interests for successful and sustainable implementation of (ecological) river engineering measures. Following a standardized QR-modeling framework, a concept map served as the basis for the structural model of the Kamp system. Based on this, two causal models are presented expressing system behaviors. Based on the most relevant entities, interacting static and process model fragments are presented. Conclusions and remarks on ongoing work are given.

Introduction

Sustainability and the NaturNet-Redime project

Sustainable development means that the needs of the present generation should be met without compromising the ability of future generations to meet their own needs. It is an overarching objective of the European Union set out in the Treaty governing all the Union's policies and activities[1]. One main important target of the renewed EU-Strategy for Sustainable Development (EU-SSD) is the involvement of citizens in the sustainability decision-making process (enhancing the participation in decision-making, promoting education and public awareness of sustainable development, informing citizens about their impact on the environment and their options for making more sustainable choices). The NaturNet-Redime project (www.naturnet.org) is charged with development of new education and decision support models for active behavior in sustainable development based on innovative web services and qualitative reasoning. Different case studies (Cioaca et al., 2006; Salles & Rios Caldas, 2006; Uzunov et al., 2006) (and also Cioaca et al., Salles et al., Nakova et al., and Noble et al., submitted to this QR workshop) are representing sustainability issues using QR and provide model fragments stored in and freely available at an online model fragment library[2]. To support integration of these case studies into a curriculum for learning about sustainability (Nuttle et al., 2006), a new software program (Garp3; see footnote 2) and a standardized QR modeling approach were developed (Bredeweg et al., 2007). Within the NaturNet-Redime-project the presented case study serves as a basis for the development of learning material for the QR-portal focusing on ecosystem, social, economic and cultural/political processes and integrated management related to catchment planning and river restoration in Austria. Main issues treated within the models are stakeholder integration as a crucial basis for a sustainable development of the whole river basin and the ecological restoration of river sites affected by water abstraction with regard to the EU-Water Framework Directive.

The EU water framework directive

In 2000, the European Union launched new water legislation, the Water Framework Directive (WFD, 2000). Within this framework, a program of measures is developed aimed at rehabilitation of degraded aquatic ecosystems across Europe. One of the key objectives of the WFD is to achieve "good ecological status" of running waters by 2015. Four organism groups (fish, macrozoobenthos, algae, macrophytes) are used as indicators to describe the ecological status.

The Kamp valley case study

Catastrophic floods and inundations in August 2002, a nearly 2000-year event, set new conditions for life and economy in the in the Kamp valley (Austria) facing flood control management, landscape architecture and land use planning with essential and future challenges. The high-water event represents a chance to develop the riverine landscape together with the local population as well as with the concerned scientific disciplines considering social, economic and ecological claims with regard to the EU-WFD. Within the whole valley there is a long tradition in water power use for grain and saw mills. Some power plants abstract water from the river for hydropower production and cause significant problems to fish by creation of residual flow stretches. The first river

[1] http://ec.europa.eu/environment/eussd/, accessed 14 February 2007.

[2] http://hcs.science.uva.nl/QRM/

engineering measures besides local bank protection were carried out around 1900. This paper presents preliminary steps in developing a QR model of the Kamp system following the structured methodology (Bredeweg et al., 2007).

Main model goals

Based on this description of the main issues facing the Kamp riverine landscape, we identified the main model goals to represent basic processes for a sustainable development of riverine landscapes:

- To develop a better understanding and representation of entities and processes involved into the very complex task of sustainable development and management of riverine landscapes in industrialized countries.
- To develop a QR-approach representing river restoration with regard to fish and the EU-WFD.

System Structure

To describe the most important concepts of sustainable development of the Kamp landscape, a concept map was developed (Zitek, 2006). This concept map includes the basic concepts of sustainable development like human society (with its sub-concepts of legislation, infrastructure, culture), institutions, nature and economy. From this, we describe the system structure, including the main entities and their structural relationships (Fig. 1). This sets the system boundaries for the modeling approach, representing interactions between energy production, flood protection and the river. Entities involved are human, infrastructure, hydropower production, economy, flood protection, vegetation, land, river, animal, river features, legislation and institution.

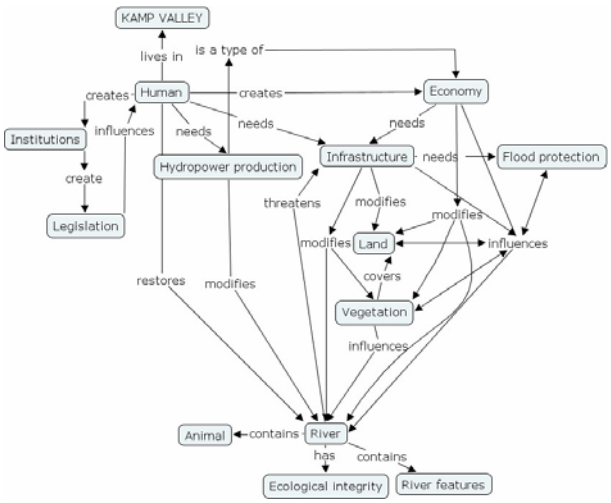


Figure 1: System structure of the Kamp valley (without restoration activity).

Two sub-systems were selected for the modeling process:

- development and implementation of measures with regard to information and participation processes with the acceptance of a measure as an indicator for sustainability (Model A)
- restoration of river sites impacted by water abstraction and channelization with regard to the EU-WFD (Model B).

Model A: Acceptance of a measure

Entities overview. The most relevant entities for the model A are "environment" (local environment, social environment), "human" (stakeholder, local population, politician, planner), "management action" (information, participation, development of measures, implementation of measures), "economic unit" (money) and "indicator" (acceptance of a measure).

Configurations overview. An initial list of entities and their configurations is presented below. If new entities are to be included, new configurations may be required.

- Human *lives* in Environment
- Planner *sets* Management action
- Economic unit *influences* Management action
- Information *informs* local population and stakeholders
- Participation *integrates* stakeholders
- Management action *influences* indicator

Agents. Agents are used to model processes that affect the system of interest, but are external to it. A catastrophic event sets the pre-requisition for the development of measures and is treated as an agent, or external influence.

Assumptions. Assumptions represent something about the system of interest, which makes them conceptually different from both entities and agents. E.g. the WFD defines the role that ecological targets have within planning activities; environmental sustainability due to measures should be reached following the approach of minimizing economic loss. It is assumed that the participation process creates multipliers that have a high influence on the acceptance of a measure within the local social environment. But additionally, official information is still important to increase the integration of the local environment to reach a high acceptance of the measures.

Model B: River restoration focusing on channelization and water abstraction

Entities overview. The most relevant entities for the model B are "water body" (river, residual flow stretch), "river feature" (water, habitat, substrate, shoreline vegetation), "driver" (hydropower production, flood protection), "technology" (hydropower plant), "human pressure" (water abstraction, channelization), "indicator" (fish, ecological integrity), "management action" (restoration), "economic unit" (money).

Configurations overview. An initial list of entities and their configurations is presented below. If new entities are to be included, new configurations may be required.

- Water body *contains* river features
- Human pressure *modifies* river features
- River features *influence* indicators
- Management action *modifies* human pressure
- Management action *influences* economic unit

Assumptions. The WFD directive is influences the whole modeling approach (5-level scheme, economic commensurability of measures, indicators, etc.). Furthermore it is assumed, that flood protection of a riverine landscape is often achieved by river channelization together with the construction of levees. But only channelization is treated as a direct impact on habitat heterogeneity within the models neglecting the importance of lateral connectivity for fish that is lost due to levees. It is further assumed that the WFD status reflects the degree of the impact. Temperature changes due to the impoundment upstream are not integrated into models yet. Also the effect of the interruption of longitudinal connectivity is not integrated.

Causal Models

Human occupation of the Kamp valley has substantially altered the riverine landscape and the river features reducing the ecological integrity of the river. Hydropower production and channelization for flood protection cause the most important pressures to the riverine system. Sustainable restoration activities integrating all stakeholder interests are an important task, especially with regard to the EU-WFD (Harrison et al., 2001). To illustrate these typical situations in the Kamp valley, two causal models are presented: one for model A and one for model B.

Causal model A: Acceptance of a measure

Fig. 2 shows the causal model for the acceptance of a measure. The success and sustainability of a measure largely depend on high agreement of the local population (integration of the local environment) and other stakeholders to the proposed measures. Acceptance of a measure is mainly influenced by information, participation, integration of stakeholder interests and of the local environment (including typical habits of the local population, landscape history, etc.). Catastrophic events, increasing the motivation of the local population influencing political interest for development and implementation of measures is treated as an important external influence (agent).

According to the causal model some of the relations might read as follows:

- Fear from catastrophic events increases the motivation of local population for actions (P+) which increases the pressure on politicians (P+) which positively influences the political interest for actions (I+); this propagates positively the money available (P+) and the development of measures (P+) as a pre-condition for the following steps.
- The integration of scientific know-how positively influences the success of the measures (P+).
- Participation and Information processes increase the integration of stakeholder interests and the integration of the local environment (I+).
- Both affect the acceptance of the measure (P+).
- If the acceptance of the measure is low, resistance against measures is high (P-).
- If resistance against measures is high, pressure on politicians is high (P+) which increases the pressure on planners (P+) which activates the information and participation process (P+).

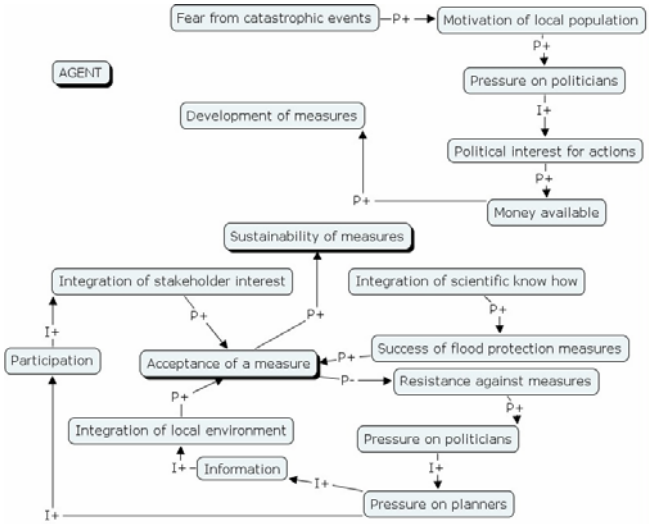


Figure 2: Causal model "acceptance of a measure" with "catastrophic event" as agent.

Causal model B: River restoration with regard to water abstraction and channelization

Water abstraction and river channelization are generally known as two of the main pressures to Austrian rivers (BMLFUW, 2005) and restoring river sections impacted by reduced flow and a changed flow regime is known to be a challenging task (Scruton et al., 1998; Erskine et al., 1999). Fig. 3 shows the causal model for two different possibilities of river restoration activities to restore the ecological integrity of impacted rivers in compliance with the WFD. According to the two pressure types, two restoration activities (Restoration I and II) might reduce the pressures, positively influencing related river features and indicators.

In this causal model some relations can be described as follows:

- The Water abstraction rate positively influences the amount of abstracted water (I+).
- The higher the amount of abstracted water, the lower is the amount of water in the river (P-), lowering the depth and flow velocity (P+), but increasing temperature (P-); these factors are known to be relevant factors influencing fish biomass, density and species diversity (P+), representing indicators for the ecological integrity (P+).
- River channelization reduces habitat heterogeneity (P-); habitat heterogeneity is positively proportional to fish reproduction, biomass, density and species diversity (P+), which are all indicators for the ecological integrity of a river (P+). Channelization is often accompanied with a reduction of shoreline vegetation (P-) which increases the temperature of a river section (P-).
- Restoration opportunities (I and II) can be seen as single or combined processes. Restoration I reduces (I-) river channelization and increases (I+) the amount of shoreline vegetation.. Restoration II reduces (I-) the water abstraction rate and positively influences the naturalness of the discharge regime (I+) which decreases substrate clogging that is negatively linked to fish reproduction (P-).

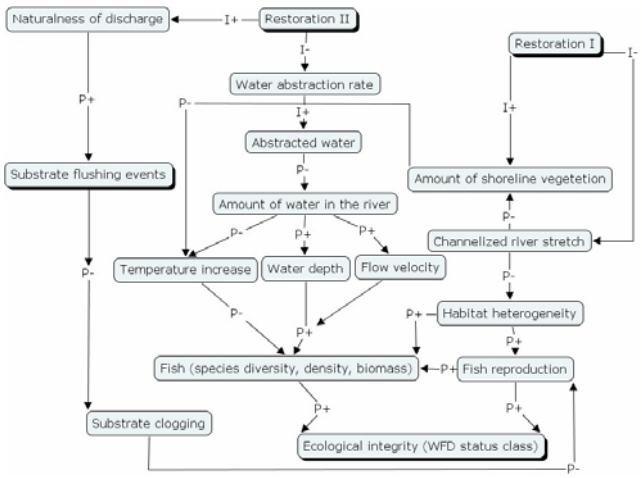


Figure 3: Causal model "river restoration with regard to the WFD".

Detailed system structure and behavior

QR models generally comprise a hierarchical library of model fragments.. In this section, the basic model fragments for the River Kamp case study are defined. The model fragments are classified as static fragment, process fragment and agent fragment. These implement the ideas presented in the causal models. Some examples for both models are given:

Model A: Static model fragments

The purpose of static model fragments is to define structural relations between entities as well as to indicate propagation of changes from one quantity to another by using proportionalities (Bredeweg et al. 2006).

Sustainability of measures.

- Conditions:
  - Entities: Indicator, Human
  - Configurations: influences
- Consequence
  - Quantities: Acceptance of a measure, Resistance against a measure, Sustainability of measures
  - Causal dependencies: Acceptance of a measure propagates negatively to resistance to a measure (P-) and positively to sustainability of a measure (P+).

Model A: Process model fragments

Process model fragments describe how values of quantities cause changes to occur in other quantities via direct influences (I+ and I-).

Participation process.

- Conditions:
  - Entities: Planner, local population, stakeholders, management action, indicator
  - Configuration: sets, participates, influences
- Consequence
  - Quantities: Participation, Integration of stakeholder interests, Acceptance of a measure
  - Causal dependencies: Participation process has a positive influence (I+) on Integration of stakeholder interests.

Model A: Agent model fragments

Agent model fragments are a special kind of process model fragment (containing direct influences I+, I-), that model how external influences cause changes in a system. They generally relate to processes that humans can potentially exert some control over, as opposed to natural processes, that humans generally cannot or do not directly control.

Pressure on politicians/political interest.

- Conditions:
  - Entities: Local population, politician
  - Configuration: influences
- Consequence
  - Quantities: Pressure on politicians, Political interest
  - Causal dependencies: Pressure on politicians has a positive influence (I+) on political interest for actions.

Model B: Static model fragments

River feature and fish.

- Conditions:

- Entities: Water body, River feature, Indicator
- Configurations: contains, influences

- Consequence
  - Quantities: Amount of water, Impact on water depth, Temperature increase, Impact on flow velocity, Species diversity, Size of fish, Loss of sensitive species, Biomass.
  - Causal dependencies: Impact on water depth and flow velocity negatively propagate to species diversity, size of fish, and biomass (P-) and positively to loss of sensitive species (P+).

Model B: Process model fragments

Process model fragments describe how values of quantities cause changes to occur in other quantities via direct influences (I+ and I-).

Water abstraction.

- Conditions:
  - Entities: Water body, River feature, Human pressure
  - Configuration: contains, influences
- Consequence
  - Quantities: Water abstraction rate, Water abstracted, Water in the river,
  - Causal dependencies: The water abstraction rate positively influences the amount of abstracted water (I+) which negatively influences the amount of water in the river (I-).

Model B: Agent model fragments

No agent model fragments are currently used within model B.

Discussion

Guided by the standardized QR modeling framework (Bredeweg et al., 2007) we were able to develop the presented models in QR language capturing important problems related to a sustainable development of riverine landscapes related to the EU-WFD. During the model implementation phase, the two models presented and scenarios will be further developed and specified. Model fragments will collaboratively developed within the collaborative model-building workbench together with the case study from UK (Noble, 2006). The collaborative model-building workbench allows for the exchange of sketches of ideas and the re-use of model fragments produced by other case studies available at the repository at the QRM portal by simply copying and pasting (see Liem et al., also presented in this workshop). At the end both presented models will represent basic aspects of a sustainable development of riverine landscapes with validity throughout Europe. To evaluate the efficacy and efficiency of the models, an interactive workshop with various stakeholders of the valley (fishermen, local water authorities and river engineers, energy producers and planners) will be organized.

Public participation is seen as perhaps the most pressing and problematic issue in ensuring the prompt and adequate implementation of the Water Framework Directive (WFD) and the achievement of integrated river basin management planning (Harrison et al., 2001). Therefore and in order to manage water resources in a more sustainable manner, new planning methodologies/ processes for river basin management need to be developed especially to achieve participation and integration in a decision-making or planning process (Hedelin, 2007). Integration of interests at various dimensions has to be achieved, including the consideration of multiple issues and stakeholders, the key disciplines within and between the natural and human sciences, multiple scales of system representation and behaviour and cascading effects both spatially and temporally. The trend to more integrative or holistic assessment and management of our resources requires the corresponding development of our science (Jakeman & Letcher, 2003). Participatory approaches to natural resource use planning and management have significant implications for managers, planners and researchers (Walker et al., 2001). Especially communication is suggested to be crucial to achieve integrated environmental management, integrated modelling, integrated assessment, or integrated knowledge (Parker et al., 2002).

Model-generated information might help in the process of stakeholder integration (Olsson & Berg, 2005). The causal models and graphic presentations as produced by Garp3 may effectively summarise a large quantity of information and will help to understand and communicate processes and relationships relevant for sustainable river restoration besides statistical relationships currently dominating in aquatic science.

Acknowledgements. This research was funded by the European Commission's Sixth Framework Program for Research and Development (project number 004074, project acronym Naturnet-Redime). More information on the project is available at http://www.naturnet.org. Furthermore the authors would like to thank Tim Nuttle and Paulo Salles for helpful comments and suggestions on an earlier version of the manuscript.

References

BMLFUW, 2005. EU-Water Framework Directive 2000/60/EG: Austrian report on the current situation of water bodies and rivers. Vienna. BMLFUW. 205.

Bredeweg, B., P. Salles, A. Bouwer, J. Liem, T. Nuttle, E. Cioaca, E. Nakova, R. Noble, A.-L. Rios Caldas, Y. Uzunov, E. Varadinova and A. Zitek, 2007. Towards a Structured Approach to Building Qualitative Reasoning

Models and Simulations. Ecological Informatics. **accepted**: xx-xx.

Cioaca, E., S. Covaliov, C. David, M. Tudor, L. Torok and O. Ibram, 2006. Textual description of the Danube Delta Biosphere Reserve case study focusing on basic biological, physical, and chemical processes related to the environment. Deliverable 6.2.1 of the NaturNet-REDIME project. Tulcea, Romania. Danube
Delta National Institute for Research and Development. 39.

Erskine, W. D., N. Terrazzolo and R. F. Warner, 1999. River rehabilitation from the hydrogeomorphic impacts of a large hydro-electric power project: Snowy River, Australia. Regulated Rivers-Research & Management **15**: 3-24.

Harrison, A., G. Schmidt, C. Avis and R. Hauser, 2001. WWF's preliminary comments on Public Participation in the context of the Water Framework Directive and Integrated River Basin Management. WWF. 15.

Hedelin, B., 2007. Criteria for the Assessment of Sustainable Water Management. **Environmental Management 39**: 151-163.

Jakeman, A. J. and R. A. Letcher, 2003. Integrated assessment and modelling: features, principles and examples for catchment management. Environmental Modelling & Software **18**: 491–501.

Noble, R., 2006. Textual description for the UK case studies on the River Trent and Great Ouse focusing on the links between the ecological and socio-economic aspects of sustainable river rehabilitation and management. Deliverable of the NaturNet-REDIME project. Hull. International Fisheries Institute. 54.

Nuttle, T., P. Salles, A. Bouwer, B. Bredeweg, A. L. R. Caldas, E. Cioaca, S. Covaliov, J. Liem, E. Nakova, M. Neumann, R. Noble, E. Varadinova, Y. Uzunov and A. Zitek, 2006. Navigating a landscape of sustainability concepts: towards progressive learning routes within and among qualitative reasoning models. Proceedings of the 5th ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, part of the 17th European Conference on Artificial Intelligence, 28 August - 1 September 2006, Riva del Garda, Italy.

Olsson, J. A. and K. Berg, 2005. Local Stakeholders' Acceptance of Model-generated Data Used as a Communication Tool in Water Management: The Rönneå Study. AMBIO: A Journal of the Human Environment **34**: 507–512.

Parker, P., et al., 2002. Progress in integrated assessment and modelling. Environmental Modelling & Software **17**: 209–217.

Salles, P. and A. L. Rios Caldas, 2006. Textual description of Riacho Fundo case study. Deliverable 6.3.1 of the NaturNet-REDIME project. Brasilia. University of Brasilia. 42.

Scruton, D. A., T. C. Anderson and L. W. King, 1998. Pamehac Brook: A case study of the restoration of a Newfoundland, Canada, river impacted by flow diversion for pulpwood transportation. Aquatic Conservation-Marine and Freshwater Ecosystems **8**: 145-157.

Uzunov, Y., E. Nakova and E. Varadinova, 2006. Textual description of river Mesta case study. Deliverable 6.3.1 of the NaturNet-REDIME project. Sofia, Bulgaria. Central laboratory of general ecology. 28.

Walker, D. H., S. G. Cowell and A. K. L. Johnson, 2001. Integrating research results into decision making
about natural resource management at a
catchment scale. Agricultural Systems **69**: 85–98.

WFD, 2000. Directive 2000/60/EC of the European Parliament and Council of 23 October 2000 establishing a framework for Community action in the field of water policy.

Zitek, A., 2006. Textual description of the river Kamp case study focusing on basic ecological and socio-economic features for an integrative and sustainable development of the riverine landscape. Deliverable of the NaturNet-REDIME project. Vienna. Institute of Hydrobiology and Aquatic Ecosystem Management, BOKU. 31.