Qualitative Simulation Based On A Logic Of Space And Time*

Z. Cui, A.G. Cohn and D.A. Randell

School of Computer Studies
University of Leeds, Leeds, LS2 9JT, UK
{cui,agc,dr}@scs.leeds.ac.uk

Abstract

We describe an envisionment-based simulation program. The program bears some design similarities to Kuipers' QSIM algorithm, but differs in the underlying ontology and in the implemented theory in the envisioning process. The program implements part of an axiomatic, first order theory that has been developed to represent and reason about space and time. Topological information is extracted from the modelled domain and is expressed in the theory as sets of distinct topological relations holding betwen sets of objects. These form the qualitative states in the underlying theory and simulation. Processes in the theory are represented in the envisionment as paths in the envisionment tree. We show the feasability of this particular ontology in the implementation of a simulation program derived from a logic-based formal theory. A description of the algorithm is given and the whole is illustrated with an example of a simulation of the processes phagocytosis and exocytosis - two processes used by unicellular organisms for garnering food and expelling waste material respectively. Finally we show how the program can be viewed as a specialized theorem prover by mapping program transformations to logical inferences in the modelling theory.

1 Introduction

Envisionment-based simulation programs used in Qualitative Reasoning (QR) are now well established. The notion of an envisionment originated in de Kleer's NEWTON (Weld and de Kleer 1990) program, but now appears as a central program design feature in many QR simulation programs - see Weld and de Kleer (1990). An envisionment takes a set of predetermined qualitative states, and expresses them in the form of graph or a tree. This represents a temporal partial ordering of all the qualitative states a modelled physical system can evolve into given some indexed state. The term "envisionment" refers to the generated tree of possible states of a modelled system, the term "envisioning" to the actual process of deriving this tree. Envisionments can be attainable or total. Attainable envisionments generate the tree from some particular initial state of the modelled system; total envisionments are generated from all possible states - see Weld and de Kleer (1990) for examples of both types in the literature. Our simulation program produces an attainable envisionment.

*This is an extended version of a paper presented at AAAI92.

The simulation program described below shares many of its general design features with Kuipers' (1986) QSIM approach to qualitative simulation. QSIM uses a set of symbols that represent physical parameters of a modelled system, together with a set of constraint equations (which are taken to be qualitative analogues of standard differential equations commonly used in mathematics and physics). The qualitative simulation starts with a structural description of the modelled domain (being the description of the parameters and constraint equations which relate the parameters to each other) and an initial state. The program produces a tree which represents the initial state of the system as the root node, and possible behaviours of the modelled system as paths in the tree from the root node to its leaf nodes.

In our simulation program, QSIM's physical parameters map to a set of mutually exhaustive and pairwise disjoint set of dyadic relations that can hold between pairs of regions. Similarly, QSIM's set of transition rules map to a set of transition rules in our theory (which determine the manner in which pairs of objects can change their degree of connectivity over time), and QSIM's constraint model maps to domain independent and dependent constraints that apply to states, and between adjacent states. While both QSIM and our simulation program take particular physical systems as a model, unlike QSIM, our simulation program first requires the user to abstract out a logical description of the physical model in terms of a set of topological relationships holding between the set of objects in the modelled domain. An analogue of QSIM's consistency filtering also appears in our simulation program.

The structure of the rest of the paper is as follows. In section 2 we outline that part of the underlying theory upon which the present simulation program is based. Section 3 discusses the simulation program. In section 4 we give an example model and resulting envisionment. The logical correctness of the program is dealt with in section 5. Finally in section 6 we discuss related and future work.

2 Overview of the spatial theory

The formal theory which underpins the simulation program (see Randell and Cohn 1989, 1992 and Randell 1991) is based upon Clarke's (1981, 1985) calculus of individuals based on "connection" and is expressed in the many sorted logic LLAMA - see Cohn (1987). The theory supports regions having either a spatial or temporal interpretation. Informally, these regions may be thought to be infinite in number, and any degree of connection from external contact to identity is allowed in the intended model.

The basic part of the formal theory assumes a primitive dyadic relation: $C(x, y)$ read as 'x connects with y' which is defined on regions. $C(x, y)$ is reflexive and symmetric. In terms of points incident in regions, $C(x, y)$ holds when regions x and y share a common point. Using the relation $C(x, y)$, a basic set of dyadic relations are defined. These relations are: '$DC(x, y)$' ('x is disconnected from y'), '$P(x, y)$' ('x is a part of y'), '$PP(x, y)$ ('x is a proper part of y'), '$x = y$' ('x is identical with y'), '$O(x, y)$' ('x overlaps y'), '$DR(x, y)$' ('x is discrete from y') '$PO(x, y)$' ('x partially overlaps y'), '$EC(x, y)$' ('x is externally connected with y)', '$TP(x, y)$' ('x is a tangential part of y'), '$NTP(x, y)$' ('x is a nontangential part of y'), '$TPP(x, y)$' ('x is a tangential proper part of y'), '$NTPP(x, y)$' ('x is a nontangental proper part of y'), '$TPI(x, y)$' ('x is the identity tangential part of y'), and, '$NTPI(x, y)$' ('x is the identity nontangential part of y'). The relations P, PP, TP, NTP, TPP

and NTPP support inverses. Of the defined relations, the set DC, EC, PO, TPP, NTPP, TPI, NTPI, and the inverses for TPP and NTPP form a mutually exhaustive and pairwise disjoint set. From now on we shall refer to this particular set, as the set of base relations defined solely in terms of the primitive relation C. A pictorial model for this set of base relations (excepting the relation NTPI) is given in Figure 1 [1]. Atomic formulae whose predicate symbol is a base relation will be called basic atoms. Note that all the relations described above can be expressed as disjunctions of sets of base relations.

For the temporal part of the theory assumed by the simulation program, we first introduce temporal regions into our ontology, which we call periods. Periods are subdivided into intervals and moments, where a moment is defined as a period that has no consituent parts such that one part is before another. In addition to periods, a new primitive relation of temporal precedence 'B(x,y)' read as 'x is before y' is added to the formalism and axiomatised to be irreflexive and transitive. A set of 13 dyadic temporal relations are then defined - see Randell (1991). These may be viewed as analogues of all the 13 interval relations common to interval logics - see e.g. Allen and Hayes (1985) and Hamblin (1971). However, for the purposes of this paper, only the relation Meets(x, y) which is irreflexive and transitive is needed. Two periods x and y are then said to meet if and only if x is before y and no other period z exists such that x is before z, and z is before y. The function 'next(x)' read as 'the next moment after (period) x' can then be defined.

In the general formal theory, an ontological distinction is made between physical objects (bodies) and the regions of space they occupy. Bodies and regions are represented in the formal theory as disjoint sorts. The mapping between the two is done by introducing a transfer function 'space(x, y)' read as 'the space occupied by x at y', that takes a body at a given moment in time, and maps this to the region of space it occupies, or to a null object if x does not exist at the moment y. In the latter case NULL(x, y) is true. The transfer function is used in the theory to define a set of ternary relations of the form $\Phi(x, y, z)$ which are used in a set of envisioning axioms and means that body x is in relation Φ to body y during period z. However, the temporal parameters in formulae used in the simulation program remain implicit, i.e. the formula NTPP(n, a) abbreviates the temporally indexed formula NTPP(n, a, t) - where t denotes a specific period during which the state obtains.

The formal theory contains a set of envisioning axioms and encodes a set of theorems (derivable from the part of the theory described above) in the form of a transitivity table - cf. Allen's (1983) transitivity table.

The envisioning axioms (see figure 1) describe direct topological transitions that can be made between pairs of regions. Thus, for example, given two regions that DC in one state, a direct transition to EC is allowed, and from EC back to DC or to where the regions PO, and so on. These axioms rule out certain transitions - for example no direct transition between DC and PO is allowed: if the transition from DC to PO is sanctioned in the model, then the underlying transition must pass

[1] This particular model assumes all the regions to be topologically closed (i.e. including their boundaries). The relation NTPI is only satisfied if the regions it is defined on are open (i.e. not including their boundaries), and given that in the theory used by the simulation program, regions cannot both be simultaneously closed and open, the relation NTPI is not represented in the restricted model we use here. We justify this restriction by noting that physical objects are most naturally represented by topologically closed spatial regions.

$\forall xyz[\mathrm{DC}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{DC}(x,y,\mathrm{next}(z)) \vee \mathrm{EC}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{EC}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{EC}(x,y,\mathrm{next}(z)) \vee \mathrm{DC}(x,y,\mathrm{next}(z)) \vee \mathrm{PO}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{PO}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{PO}(x,y,\mathrm{next}(z)) \vee \mathrm{EC}(x,y,\mathrm{next}(z)) \vee \mathrm{TPP}(x,y,\mathrm{next}(z)) \vee$
$\mathrm{TPP}^{-1}(x,y,\mathrm{next}(z)) \vee \mathrm{TPI}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{TPP}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{TPP}(x,y,\mathrm{next}(z)) \vee \mathrm{NTPP}(x,y,\mathrm{next}(z)) \vee$
$\mathrm{PO}(x,y,\mathrm{next}(z)) \vee \mathrm{TPI}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{NTPP}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{NTPP}(x,y,\mathrm{next}(z)) \vee \mathrm{TPP}(x,y,\mathrm{next}(z)) \vee \mathrm{TPI}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{TPI}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{TPI}(x,y,\mathrm{next}(z)) \vee \mathrm{PO}(x,y,\mathrm{next}(z)) \vee$
$\mathrm{TPP}(x,y,\mathrm{next}(z)) \vee \mathrm{NTPP}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{TPP}^{-1}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{TPP}^{-1}(x,y,\mathrm{next}(z)) \vee \mathrm{NTPP}^{-1}(x,y,\mathrm{next}(z)) \vee$
$\mathrm{PO}(x,y,\mathrm{next}(z))\mathrm{TPI}(x,y,\mathrm{next}(z))]]$
$\forall xyz[\mathrm{NTPP}^{-1}(x,y,z) \wedge \Gamma \rightarrow [\mathrm{NTPP}^{-1}(x,y,\mathrm{next}(z)) \vee \mathrm{TPP}^{-1}(x,y,\mathrm{next}(z)) \vee$
$\mathrm{TPI}(x,y,\mathrm{next}(z))]]$
where $\Gamma \equiv \neg\mathrm{NULL}(x,next(z)) \wedge \neg\mathrm{NULL}(y,\ \mathrm{next}(z))$

Figure 1: The eight envisioning axioms.

through the intermediate EC state. A pictorial representation of of the envisioning axioms is illustrated in Figure 2.

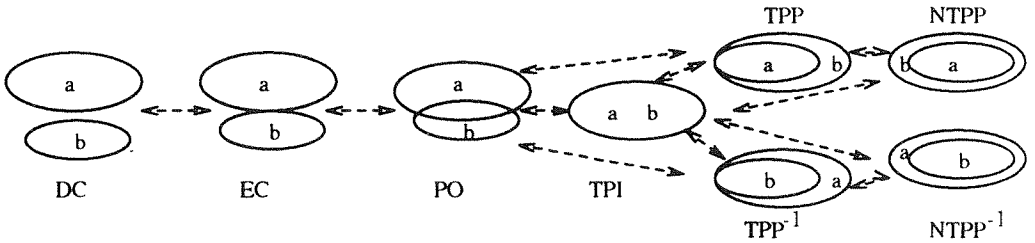

Figure 2: A pictorial representation of the base relations and their direct topological transitions.

The theory also uses a precomputed transitivity table (Table 1) for the set of dyadic base relations described above - for details see Randell, Cohn and Cui (1992). Each R3(a,c) entry in the table represents a disjunction of all the possible dyadic relations holding between regions a and c, for each R1(a,b) and R2(b,c) conjunction - where R1, R2, R3 are elements of the set of base relations in the theory. The transitivity table is used in the simulation program for checking consistency of state descriptions in the envisioning process.

As mentioned above, only a part of the general theory is actually implemented in the simulation program. For example, the general theory also includes an additional primitive function 'conv(x)' read as 'the convex hull of x', which is axiomatised and is used to generate a further set of dyadic relations. These additional relations are used to describe regions that are either inside, partially inside or outside other regions - see Randell, Cohn and Cui (1992). As with the set of relations defined solely in terms of C, the extended theory including the new set of inside and outside relations also admits the possibility of constructing several further sets of base relations, depending upon the degree of representational detail required by the user. For the basic extension to the theory, the set of base relations extend from 9 to 23. However, here we simply concentrate upon the set of base relations defined solely in terms of C which turns out to be sufficient to demonstrate the general utility of our approach.

R1(a,b) \ R2(b,c)	DC	EC	PO	TPP	NTPP	TPP^{-1}	$NTPP^{-1}$	TPI	NTPI
DC	no.info	DR,PO,PP	DR,PO,PP	DR,PO,PP	DR,PO,PP	DC	DC	DC	DC
EC	DR,PO,PP^{-1}	DR,PO TPP,TP^{-1}	DR,PO,PP	EC,PO,PP	PO,PP	DR	DC	EC	×
PO	DR,PO,PP^{-1}	DR,PO,PP^{-1}	no.info	PO,PP	PO,PP	DR,PO,PP^{-1}	DR,PO PP^{-1}	PO	PO
TPP	DC	DR	DR,PO,PP	PP	NTPP	DR,PO TPP,TP^{-1}	DR,PO PP^{-1}	TPP	×
NTPP	DC	DC	DR,PO,PP	NTPP	NTPP	DR,PO,PP	no.info	NTPP	NTPP
TPP^{-1}	DR,PO,PP^{-1}	EC,PO,PP^{-1}	PO,PP^{-1}	PO,TPP,TP^{-1}	PO,PP	PP^{-1}	$NTPP^{-1}$	TPP^{-1}	×
$NTPP^{-1}$	DR,PO,PP^{-1}	PO,PP^{-1}	PO,PP^{-1}	PO,PP^{-1}	O	$NTPP^{-1}$	$NTPP^{-1}$	$NTPP^{-1}$	$NTPP^{-1}$
TPI	DC	EC	PO	TPP	NTPP	TPP^{-1}	$NTPP^{-1}$	TPI	×
NTPI	DC	×	PO	×	NTPP	×	$NTPP^{-1}$	×	NTPI

Table 1: Transitivity table for the 9 basic relations. If $R_1(a, b)$ and $R_2(b, c)$, it follows that $R_3(a, c)$ where R_3 is looked up in the table. "×" entries mean that the corresponding conjunction $R_1(a, b)$ and $R_2(b, c)$ cannot be simultaneously satisfied, and "no info." that no base relation is excluded. Multiple entries in a cell are interpreted as disjunctions. Note that DR stands for DC and EC, PP for TPP and NTPP, PP^{-1} for TPP^{-1} and $NTPP^{-1}$, TP^{-1} for TPP^{-1} and TPI, and O for PO, TPP, NTPP, TPP^{-1}, $NTPP^{-1}$, TPI and NTPI.

3 The Simulation program

State descriptions in the simulation program are represented as conjunctions of ground atomic formulae. The program first of all takes an initial state description, then evolves successive states according to the restrictions imposed by direct topological transitions encoded in the envisioning axioms, by sets of constraints that apply within a state or between states, and by any sets of add or delete rules that sanction the introduction and deletion of named entities in the modelled domain respectively. A consistency check is made for each state, first for the initial state, and then for all potential evolved states generated in the envisioning process. The envisioning process terminates when for each path generated in the envisionment tree, the last state repeats an earlier one. Each path of states S_1, S_2, ... corresponds to a sequence of periods, t_1, t_2, ... such that Meets(t_i, t_{i+1}), and the state description of S_i obtains during t_i. Each complete path corresponds to a possible behaviour of the physical model as predicted by the program. However, because the transition rules always allow the possibility that the relationship between two entities continues indefinitely, each initial subpath also corresponds to a predicted behaviour.

The program requires a complete n-clique as the initial state, i.e. n(n-1)/2 atomic formulae. This requirement is needed for consistency and constraint checking by the program to function correctly.[2]

3.1 Constraints

The simulation program supports two types of constraints. These are *intrastate* and *interstate* constraints respectively. Intrastate constraints are constraints that

[2]However, in practice, a partial description of the initial is usually supplied by the user and a) the program computes the complete description or descriptions, if more than one is consistent, and b) only those state descriptions that arise from evolved transitions from pairs of entities described in the initial state are explicitly represented in the envisionment. Actually, our program is slightly different to the procedure specified in section 3.4 but this need not concern us here.

apply within a state, and interstate constraints between adjacent states - that is to say, between consecutive states, or states which meet. For example, in the physical system which is used to illustrate this simulation program below - namely modelling phagocytosis of unicellular organisms - an intrastate would be the assertion that the cell's nucleus is always part of the cell, and an interstate would be the fact that once the food is ingested during phagocytosis and becomes a part of the amoeba, it will remain so. Formally, both types of contraints assume the following forms:

Intrastate constraint: Φ, where Φ is a quantifier free formula, and all terms are variables or constants (all variables are implicitly quantified). Note that Φ must be composed of basic atoms.

Interstate constraints:

$$\Phi \rightarrow (R_0 \Longrightarrow (R_1 \vee ... \vee R_n)) \qquad \Phi \rightarrow (R_0 \not\Longrightarrow (R_1 \wedge ... \wedge R_n))$$

where Φ is as above, and the R_i are basic atoms predicating the same terms. In the first case, where Φ holds, if R_0 then in any next state the disjunction $R_0 \vee R_1 \vee ... \vee R_n$ holds, while in the second case the disjunction $R0 \vee R'_1 \vee ... \vee R'_{n_i}$ must hold where each R'_i is a base atom predicating the same terms as R0 and $R'_i \neq R'_j$ for any i, j. The presence of an interstate constraint does not force a transition to take place.

3.2 Add and delete rules

In addition to the set of constraint rules described above, the simulation program also supports add and delete rules. Both sets of rules can be viewed as another kind of inter-state constraint. In this case add rules simply sanction the introduction of new objects into the domain at the next state, and delete rules the elimination of particular objects in the next state. In the model used to illustrate our program, an example of an add rule is where, having enveloped the food, a vacuole is formed in the amoeba, while an example of a delete rule is where the vacuole containg waste material passes out of existence as it opens up and discharges its contents into the amoebal environment.

Add and delete rules assume the following forms: **add** o_1, ... o_n **with relations** Ψ_1 **when** Ψ_2; **delete** o_1, ..., o_n **when** Ψ_2 . Ψ_1 is a conjunction of basic atoms, and Ψ_2 is a quantifier free Boolean composition of atoms. o_1, ... o_n must be ground terms (at least in the current implementation). An add rule is fired when the 'when' condition is true for some instantiation of free variables in the condition, and will add o_1, ... o_n to all next states with the specified relations. Similarly, delete rules will be fired when the 'when' condition is true and will delete all the specified objects in all next states.

3.3 The Algorithm

The algorithm first of all takes an initial state of the modelled physical system, then proceeds to generate the envisionment. Each state in the envisioning process is checked for intrastate consistency before the next state in the envisionment is generated. The completed tree representing the envisionment has the initial state as the root node, and paths tracing to leaf nodes as distinct sequences of transitions undergone by the set of modelled objects.

The algorithm is as follows. First we put the initial state s0. in a set S of unexpanded states; then the following steps are executed:

1. If S is empty then stop.
2. Select and remove a state Si from S.
3. Check consistency of Si, if Si is inconsistent, then go to 1.
4. Select applicable transition rules by applying interstate constraints.
5. Apply all the selected transition rules to produce a set of possible states.
6. Apply add and delete rules.
7. Delete any states that violate an intrastate constraint.
8. Add remaining states generated to S.
9. Go to 1.

We discuss the details of steps 3 to 7 in the subsections below.

3.3.1 Consistency checking

In step 3 the algorithm uses a simple form of consistency checking step to filter out sets of atomic formulae (being a potential state in the simulation and thus in the physical model) whose conjunction is inconsistent in the underlying theory, and thus supports no model. In this instance, we use the results encoded in the transitivity table. Given n-objects in the modelled domain, there are exactly n(n-1)/2 atomic formulae in a state. In particular for each tuple of objects x,y,z. there are three atomic formulae of the form $R_1(x, y)$, $R_2(y, z)$ and $R_3(x, z)$. Consistency checking simply consists of checking that each $R_3(x, z)$ formula is logically implied by $R_1(x, y)$ and $R_2(y, z)$ for each y $\notin$ {x, z}. In use this is effectively the same as Allen's (1983) constraint satisfaction algorithm, except that our algorithm can be simplified since we have no "disjunctive labels" because we have restricted state descriptions to conjunctions of base relations.

3.4 Generating next states

In steps 4 through to 7, the algorithm takes a state produced in step 3. and procedes to generate a new state. The selected state Si composes of a set of basic atoms. For each atom there are between 1 and 5 applicable transition rules - see Figure 1. In step 4 possible transitions for each atom which violate an interstate constraint are filtered out. In step 5 the remaining transitions are applied in all possible combinations to yield a set of posible next states. In step 6 the add and delete rules are then applied in that order. Finally, in step 7 any next states which violate an intrastate constraint are deleted.

4 An Example

By way of a simple example, we shall demonstrate the simulation program by modelling cellular behaviour - in particular, the processes known as phagocytosis and exocytosis. Phagocytosis is the process by which cells surround, engulf and then

digest food particles. It is the feeding method used by some unicellular organisms of which the amoeba is an example, and which is adopted here. The same process is used by white blood cells in an attempt to deal with invading micro-organisms. Exocytosis is the name given to a similar inverse process where waste material originally contained in a cell is subsequently exprelled from the cell.

In the proposed model, an amoeba is depicted in a fluid environment containing other organisms which are its food. Each amoeba is credited with vacuoles (being fluid filled spaces) containing either enzymes or food which the animal has ingested. The enzymes are used by the amoeba to break down the food into nutrient and waste. This is done by routing the enzymes to the food vacuole. Upon contact the enzyme and food vacuoles fuse together and the enzymes merge into the fluid containing the food. After breaking down the food into nutrient aand waste, the nutrient is absorbed into the amoebal protoplasm, leaving the waste material in the vacuole ready to be expelled. This is achieved by letting the waste vacuole pass to the exterior of the protozoan's body, which opens up, letting the waste material pass out of the amoeba and into the amoebal environment.

The formal description of the physical model is as follows. We assume six physical objects: a, f, n, e, nt, w and v, standing for the amoeba, its food, the amoeba's nucleus, a packet of enzymes, nutrient, a body of waste material, and a vacuole respectively. In the simulation, the vacuole, the nutrient and the waste are generated dynamically as the process is undergone.

The initial state is represented by the conjunction of the following atomic formulae: DC(a,f), NTPP(n,a), NTPP(e,a), DC(n,e) and DC(e,f). [3]

Next we introduce our set of domain constraints for the physical model. First the interstate constraints:

$1) EC(f,a) \not\Rightarrow DC(f,a)$ $\quad$ $8) DC(nt,v) \Longrightarrow DC(nt,v)$
$2) PO(f,a) \not\Rightarrow EC(f,a)$ $\quad$ $9) EC(nt,v) \not\Rightarrow PO(nt,v)$
$3) TPP(f,a) \not\Rightarrow PO(f,a)$ $\quad$ $10) PO(nt,v) \Longrightarrow EC(nt,v)$
$4) TPP(f,a) \not\Rightarrow TPI(f,a)$ $\quad$ $11) TPP(nt,v) \not\Rightarrow TPI(nt,v)$
$5) NTPP(f,a) \not\Rightarrow TPI(f,a)$ $\quad$ $12) NTPP(nt,v) \not\Rightarrow TPI(nt,v)$
$6) EC(w,a) \not\Rightarrow PO(w,a)$ $\quad$ $13) EC(e,f) \not\Rightarrow DC(e,f)$
$7) PO(w,a) \Longrightarrow EC(w,a)$ $\quad$ $14) PO(e,f) \Longrightarrow TPP(e,f)$

Constraints 1 to 3, 6 and 7, and 13 and 14 respectively impose a unidirectionality of movement between the food and the amoeba, between the waste material and the amoeba and between the enzyme packet and the food. In the first case when the food is in contact with the amoeba it is always ingested to become a proper part of the animal; in the second case once the waste material is in external contact with the animal, it will never be reingested, and in the last case once the enzyme packet contacts the food, it will always pass into it becoming a part. Constraints 4 and 5, and 6 and 7 respectively impose the conditions that once the food is ingested (and is thus a proper part) it will remain a proper part of the animal, and that nutrient once produced (being a proper part of the vacuole) remains a proper part.

[3] In the initial state, since there are 5 objects, there are really 10 relationships to be specified. As mentioned earlier, the program expands a user supplied partial description to a complete description. In fact although the formula DC(e,f) is formally derivable in the general theory from the first four atomic formule, it is represented explicitly in the input languge here, otherwise no relation between e and f will be generated in subsequent states in the envisioning process - see earlier footnote.

Without these constraints the transition from being a proper part to being identical sanctioned by the envisioning axioms is not violated; this would simply result in a possible state being generated in the envisionment with the amoeba being part of the food, and the vacuole part of the nutrient!

The intrastate constraints are all straightforward to understand and just impose the obvious static topological constraints between the domain entities.

NTPP(n, a), NTPP(e, a), PP(nt, a), PP(v, a), DR(n, e), DR(n, v)
PP(w, v), PP(f, v), (PP(w, a) — PP(v,a))

There are two add-rules. The first rule introduces nutrient and waste into the food vacuole when the enzyme packet is a proper part of the food, while the second rule sees the creation of the vacuole when the food is a proper part of the amoeba. The delete rules govern the deletion of the enzyme and food, and vacuole respectively. Since the first add rule below contains no basic atoms in the 'with relations' component, it is actually schematic for 4 rules in which only basic atoms are used.

add nt, w **with relations** $PP(nt, v) \wedge PP(w, v)$ **when** $TPP(e, f)$
add v **with relations** $TPP(v, a) \wedge TPP(f, v)$ **when** $TPP(f, v)$
delete e, f **when** $P(e, f)$
delete v **when** $TPP(v, a) \wedge PP(w, v) \wedge DR(nt, v)$

The simulation program produces an envisionment with 76 distinct states. Our constraints are sufficiently strong because each complete path corresponds to the English description of phagocytosis and exocytosis given above. A pictorial representation of two paths generated in the envisionment is given in Figure 3.

In both paths generated we can see that the food is ingested by the amoeba, a vacuole is formed which then contains that food, digestion takes place transforming the food into nutrient and waste, and finally the waste is exprelled. Note that in one path the enzyme packet begins to be absorbed into the food before the food is completely enveloped by the amoeba, while in another path the vacuole is formed before the enzyme packet is similarly absorbed.

Altogether there are 6 terminal states although there are 264 paths leading from the initial state to these final states representing different ordering of the topological transformations. However all the complete paths predict that phagocytosis and exoctosis will be undergone. Some of the paths exhibit oscillatory behaviour.

5 Correctness

As mentioned above, the program terminates when for each path generated in the envisioning process, the last state repeats an earlier one. It should be evident that the algorithm will terminate if there are no add rules, but the same applies if there are finitely many add rules. This follows from the syntactic restriction on add rules, that the objects must be ground terms, so only finitely many new objects can ever be introduced.

It is important to show that all the behaviours predicted by the simulation correspond to possible behaviours of the physical system being modelled. This issue

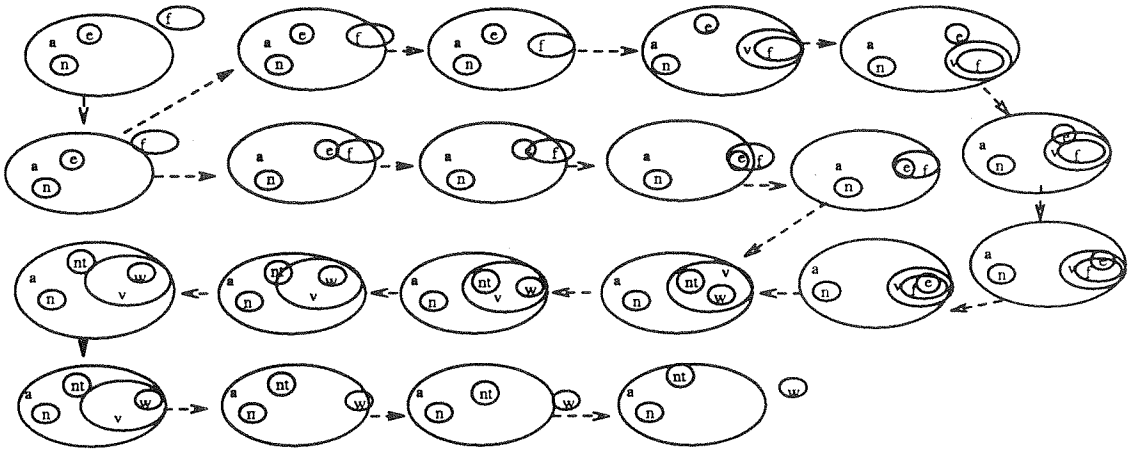

Figure 3: A pictorial representation of two paths generated.

brings to the fore the question whether or not the simulation can be proved to be "sound" and "complete". In our case by "soundness" we need to show that every frontier of the envisionment tree (viewed as a disjunction) generated in the simulation is a provable consequence in the underlying theory, and by "completeness", to show that, given an initial state, every proveable disjunction of conjoined basic atoms in the underlying theory will be expressed in the envisionment. Whereas Kuipers proves the correctness of QSIM relative to ordinary differential equations, our gold standard is the logical formalism presented in Randell (1991). We discuss these issues in the following subsection.

5.1 The Logical Basis of the Program

We now show how the above simulation system can be viewed as a specialized theorem prover. In particular we wish to show that the following is true.

Conjecture $\{AF, \Delta\} \models [S_0 - (S_1 \vee \ldots \vee S_n)]$ and $\forall i (1 \leq i \leq n)\ \{AF, \Delta\} \not\models [S_0 - (S_1 \vee \ldots \vee S_{i-1} \vee S_{i+1} \vee \ldots \vee S_n)]$ iff an envisionment with root S_0 and the frontiers $S_1, \ldots, S_n$ can be produced by the program. Here Δ is a logical representation of the constraints, add and delete rules, and the structure of the envisionment.

The if direction represents a soundness result and the only if direction a completeness result. A straightforward induction on the structure of the tree shows that we can restrict our attention to a root S_0 and immediate descendents $S_1, \ldots, S_n$.

The soundness result is fairly easy to show, but our proof of completeness still relies upon an unproven conjecture. First we need to show how to represent the various structures in the simulation program in our logic.

Formally, a state S_i corresponds to $\{R(a, b, t_i) : R(a, b) \in S_i\}$. The structure of the envisionment is generated by a set of atoms $\{Meets(t, t')$: S' is a sucessor state of S$\}$. Intrastate constraints $\Phi(a_1, \ldots, a_m)$ in the program correspond to $\forall t(\neg \mathrm{NULL}(a_1, t) \wedge \ldots \wedge \neg \mathrm{NULL}(a_m, t) - \Phi(a_1, \ldots, a_m, t))$. Interstate constraints of type 1 can be represented as $\forall t t'[[Meets(t, t') \wedge \Phi(t)] - (R_0(t) - (R_0(t') \vee R_1(t') \vee \ldots R_m(t'))]$. Interstate constraints of type 2 can be replaced by interstate constraints of type 1 because there are only finite number of base relations.

Add and delete rules are translated by the following 2 wffs respectively.

$\forall tt'((\Phi_2(t) \wedge Meets(t,t')) \rightarrow (\Phi_1(t) \wedge \neg \mathrm{NULL}(o_1,t') \wedge \ldots \wedge \neg \mathrm{NULL}(o_n,t'))$
$\forall tt'((\Phi_2(t) \wedge Meets(t,t')) \rightarrow (\mathrm{NULL}(o_1,t') \wedge \ldots \wedge \mathrm{NULL}(o_n,t'))$

We also need axioms to ensure the continued (non) existent of objects unaffected by add or delete rules. The following schemas, paramaterised by 'o' suffice.

$\forall tt'(\neg\Phi(t) \wedge \mathrm{Meets}(t,t') \wedge \mathrm{NULL}(o,t)) \rightarrow \mathrm{NULL}(o,t'))$
$\forall tt'(\neg\Phi'(t) \wedge \mathrm{Meets}(t,t') \wedge \neg\mathrm{NULL}(o,t)) \rightarrow \neg\mathrm{NULL}(o,t'))$

where Φ is the conjunction of all Φ_2 conditions (with suitable renaming to avoid variable clashes) in add rules for o. and Φ' is the conjunction of all Φ_2 conditions in delete rules for o. (If there are no add rules for o then $\Phi(t)$ is taken false. and similarly for $\Phi'(t)$.)

A step by step analysis of the program shows that each step preserves soundness and that there are no extraneous S_i, thus proving the if part of conjucture 1. The only tricky part is the consistency checking step (3). In the program each triple of atoms $R_1(a,b), R_2(b,c), R_3(a,c)$ in a state is checked for consistency using a transitivity table analogous to that of Allen(1983), see Table 1. In Randell et al (1992), we demonstrate its soundness and completeness. This soundness ensures the soundness of step 3.

The central part of the completeness proof is to show that disjunction of next states $(S_1 \vee \ldots \vee S_n)$ is minimal, i.e. none of the S_i are inconsistent. Athough the transitivity table is complete, this does not necessarily imply that simply checking tuples of atoms guarantees global consistency with respect to the theory. We have not yet been able to prove this formally (though we have no counterexample). However, three space is the intended model in the theory. If this is indeeded the case, then the following operations on regions are allowed: cutting, resizing, overlapping. Since there are indefinitely many DC regions in three space, it turns out that the completeness of the consistency checking algorithm can then be proved. Therefore the immediate task is to show there is a model of the theory, which allows infinite number of DC regions and all of the aforementioned operations.

5.2 Complexity

The critical point about the algorithm (and its complexity) is that states are complete, i.e., all relations between all objects are explicitly given in terms of base atoms and there is no disjunctive or indefinite information. This means that all constraints and add/delete rules can be considered individually, one at a time, without worrying about interactions. The complexity of the algorithms is as follows:

Step 3 — the complexity of consistency checking is $O(n^3)$ because there are $n^3 - 3n^2 + 2$ different triples given n objects in a state.

Step 4 — Suppose there are c interstate constraints and each constraint contains at most v variables and there are n objects, then each constraint can be applied at most C_n^v ways. This is polymonial of degree of v. Applying a constraint is linear to the number of connectives in it.

Step 5 — If there are n objects there are $(n^2 - n)/2$ relations. The maximum branching rate in the graph for direct topological transitions is 5 (from equality

if interstate constraints forbid transition to equality and no objects start off equal, then maximum branching rate is 2) so there are at most 5^n successor states (but more likely 2^n which is of course still exponential). This compares to the situation in QSIM. In practice, consistency checking will prune the number of next states dramatically (though they still have to be generated and checked).

Steps 6 and 7 — the complexity of these steps are the same as step 5, i.e. $O(n^r)$.

6 Related Work

For a detailed discussion of the ontology and formalism used in the simulation see Randell (1991) . We have already discussed the relationship between this simulation program and Kuiper's QSIM above. The volume (Weld and de Kleer 1990) contains several papers on qualitative spatial simulation. Forbus (1980) reports on a simulator called FROB, Gardin and Meltzer (1989) describe an analogical spatial simulator, but all these use very different ontologies to our work. Freksa (1990), Hernandez (1990) and Mukerjee and Joe (1990) present qualitative ontologies of space based on Allen's temporal logic but does not consider simulations. Kaufman (1991) presents a logic of space based on tolerance spaces and uses it to analyze (for example) why a string can pull but not push.

7 Work in Progress

In section 2 we mentioned how further dyadic relations describing bodies that are either inside, partially inside or outside each other can be added. This set could be exploited in the amoeba simulation to give a richer and more realistic model where the food can be made to pass from being ouside the animal to being inside the animal, and then options would be available once the food has been engulfed to whether the food is modelled as forming a part of the animal or not. Originally we simply specialised the DR relation to cover relations describing bodies being inside, partially inside and outside others, together with their inverses, but this ignored some useful distinctions that could be drawn between different cases of bodies being inside another. In this case we separate out the case where one body is topologically inside another, and where one body is inside another but not topologically inside this we call being geometrically inside. A pictorial representation of these are given in Figure 4 below. The important point of one body being topologogically inside another is that one has to 'cut' through the surrounding body in order to reach and make contact with the contained body; in the geometrical variant, this is not the case.

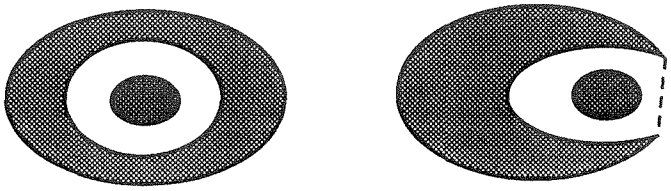

Figure 4: The distinction between being topologically and geometrically inside.

The definitions for these new variants are given below based on the predicate

Connected(x)[4]. and the function outside(x).

$$\text{Separated}(x,y) \equiv_{def} \neg\text{C}(\text{cl}(x),y) \wedge \neg\text{C}(x,\text{cl}(y))$$
$$\text{Connected}(x) \equiv_{def} \neg yz[\text{EQUAL}(\text{sum}(y,z),x) \wedge \text{Separated}(y,z)]$$
$$\text{outside}(x) =_{def} \iota y[\forall z[\text{C}(z,y) \rightarrow \exists w[\text{Outside}(w,x) \wedge \text{C}(z,w)]]]$$
$$\text{Top-Inside}(x,y) \equiv_{def} \text{Inside}(x,y) \wedge$$
$$\forall z[[\text{Connected}(z) \wedge \text{C}(z,x) \wedge C(z,\text{outside}(y)] \rightarrow \text{O}(z,y)]$$
$$\text{Geo-Inside}(x,y) \equiv_{def} \text{Inside}(x,y) \wedge \neg\text{Top-Inside}(x,y)$$

It is relatively easy to see how a path in the simulation using this extended set of relations would satisfy the English description of phagocytosis and exocytosis. First the food would be outside the animal, then it would pass to be partially inside, to being geometrically inside and then topologically inside as the vacuole containing the food is formed. Exocytosis would simply see a reversal of this sequence except that waste material replaces the food in the example. With the food ingested, and broken into nutrient and waste material (as before), then we return to the set of basic relations as the nutrient is allowed to overlap the amobal protoplasm and become part of it.

It is also possible to specialise the relation of being geometrically inside too – in this case setting up definitions to distinguish between the following pictorial representations – Figure 5 below:

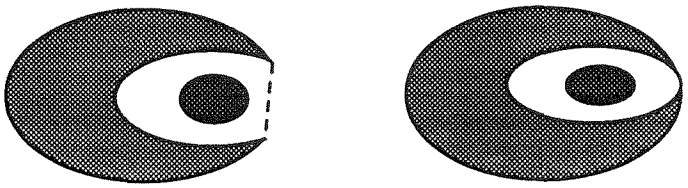

Figure 5: Two variants of being geometrically inside.

In order to make this formal distinction we first set up a stronger case of a connected or one-piece region to that assumed above. The important part of the following definition is the $\text{P}(\text{conv}(\text{sum}(v,w)),x)$ literal in the consequent of the definiens. This condition ensures that the connection between any two parts of a region whose sum equals that region, is not point or edge connected. That is to say it ensures a 'channel' region exists connecting any two connected parts. This notion of being connected mirrors and simplifies our previous definition of a quasi-manifold – in this case we use the concept of a convex body rather than use topological and Boolean concepts in the earlier definition – see Randell and Cohn (1989). Details of the axiomatised primitive convex-hull function can be found in Randell et al(1992).

$$\text{Connected}'(x) \equiv_{def} \text{Connected}(x) \wedge \forall yz[[\text{sum}(y,z) = x \rightarrow \text{C}(y,z)] \rightarrow$$
$$\exists vw[\text{P}(v,y) \wedge \text{P}(w,z) \wedge \text{P}(\text{conv}(\text{sum}(v,w)),x)]]$$

[4]It should be noted here while we have used the original theory based on Clarke's calculus to set up these definitions, we have developed a new calculus which obviates the need for the topological distinctions drawn between open, semi-open and closed regions. In the new theory, the definition for a connected (one piece) region and the strong concept of being connected using the convex-hull property reappears, but differs from these appearing here – see Randell et al (1992a, 1992b) for further details.

Now we give the formal distinction between the two cases of being geometrical inside. In the first case a 'channel' region exists connecting the outside of the surrounding body with the contained body, in the second case the surrounding body has closed forming (in this case) a point connection. In both cases we can see how in contrast with the notion of being topologically inside, it is possible to construct a line segment that connects with both the surrounding body and the contained body without cutting through the surrounding body. Definitions distinguishing between the two cases are as follows, where the "Open" and "Closed" variants respectively refer to the first and second cases described above.

$$\begin{aligned}
\text{Geo-Inside-Open}(x,y) \equiv_{def} & \text{ Geo-Inside}(x,y) \wedge \\
& \text{Connected}'(\text{sum}(\text{inside}(y), \text{outside}(y))) \\
\text{Geo-Inside-Closed}(x,y) \equiv_{def} & \text{ Geo-Inside}(x,y) \wedge \\
& \text{Connected}(\text{sum}(\text{inside}(y), \text{outside}(y))) \wedge \\
& \neg\text{Connected}'(\text{sum}(\text{inside}(y), \text{outside}(y)))
\end{aligned}$$

The rewrite option mentioned above would require additional relations to be defined. The idea would be to allow the user to choose between whether one body is inside or part of another, where the relations TPP, NTPP and their inverses apply in the part/whole case of this. For example, in order for a configuration satisfying the NTPP relation to be redescribed so that both parts in the configuration are discrete, we require a specialisation of the topologically inside definition to be defined. In this case we would need to ensure that apart from the body that maps to the nontangential part, no other region exists disjoint with this region that could also be viewed as being inside the enclosing region. An idential rationale also applies to the case where a configuration satisfies the TPP relation.

Any simulation using these new definitions in the modelling language would of neccessity require an extended set of envisioning axioms and a very large transitivity table to be constructed. Simply using the unexpanded set of inside, partially inside and outside definitions (and their inverses) alone, generates a set of 23 base relations (529 cells) - see Randell, Cohn and Cui (1992). Further refining of the inside relations to include the distinction between being topologically inside and geometrically inside, increases this set to 31 (961 cells) - this does not exhaust the distinctions mentioned above. Towards this end we have recently constructed a transitivity table via a program which reasons about a bitmap representation of space for the set of 23 base relations, but the resulting transitivity table has not yet been verified with respect to the modeling theory. It should be remembered that each entry in the table corresponds with a theorem in the formal theory, emphasising the difficulty of the task - again see Randell, Cohn and Cui (1992).

At present, the modelling primitives simply capture qualitative information relating pairs of regions. These could be extended to include metric information, capturing for example notions of relative size and distances between objects. The language for expressing constraints would then need to be extended to allow relations of such quantities to be expressed. A standard library of constraints would relate (for example) the distance between two objects and their relationship (eg. if the distance is non-zero then they must be DC). The possibility of introducing a metric extension to the theory is outlined and discussed in Randell (1991). The derivatives of quantities could be introduced and reasoned about in the usual QR manner. We already have definitions that allow one to define increasing, decreasing and constant magnitudes over time - see Randell et al (1992a); these could be

used in the simulation as the basis for reasoning about changes in distances between objects, or the degree of overlap or inclusion between bodies or regions. Further envisaged extensions to the theory that would include a subtheory of motion to the modelling language, for at present motion is represented implicitly by specified topological transitions between sets of objects. Other useful extensions would include explicit information about causality and processes; the latter including teleological accounts of a physical system's behaviour. Another desirable extension to the program is to handle ambiguities produced by alternate temporal orderings of topological transformation where these are not important.

In the implementation presented, constraints and objects have to be individually specified. However, this can and has been generalised in the current program to allow for generic constraints and typed objects in the program's description language, relating individuals of particular types.

References

[1] J. F. Allen:[1983] "Maintaining knowledge about temporal intervals." *CACM*. Vol. 26, pp. 832–843, 1983

[2] J. F. Allen, P. J. Hayes:[1985] "A Common Sense Theory of Time." in *Proc. IJCAI*, Morgan Kaufmann, Los Altos., 1985

[3] B. L. Clarke:[1981] "A Calculus of Individuals Based on Connection." *Notre Dame Journal of Formal Logic*, Vol. 2, No. 3., 1981

[4] B. L. Clarke:[1985] "Individuals and Points," *Notre Dame Journal of Formal Logic* Vol. 26, No. 1, 1985

[5] A. G. Cohn:[1987] " A More Expressive Formulation of Many Sorted Logic." *J. Autom. Reasoning.*, Vol. 3(2), pp. 113–200, 1987

[6] Z. Cui, A. G. Cohn and D. A. Randell:[1992] " Qualitative Simulation Based On A Logical Formalism Of Space And Time." *Proceedings of AAAI92. AAAI Press, 1992*

[7] K. Forbus:[1980] "Spatial and Qualitative Aspects of Reasoning about Motion." in Proc. AAAI-80, Morgan Kaufmann, San Mateo, Ca.

[8] C. Freksa:[1990] "Qualitative Spatial Reasoning". Proceedings of Workshop RAUM, University of Koblenz, 1990.

[9] F. Gardin andB. Meltzer: [1989] *Analogical Representations of Naive Physics. Artificial Intelligence*, **38** 139–159

[10] C. L. Hamblin:[1971] " Instants and Intervals." *Studium Generale.*, Vol. 24, pp. 127–134, 1971

[11] D. Hernandez:[1990] "Using Comparative Relations to represent Spatial Knowledge", Workshop RAUM, University of Koblenz.

[12] S. G. Kaufman:[1991] "A Formal Theory of Spatial Reasoning." in Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning.

[13] B. Kuipers:[1986] " Qualitative Simulation," *Artificial Intelligence.*, Vol. 29, pp. 298–338, 1986

[14] A. Mukerjee and G. Joe:[1990] "A Qualitative Model For Space,", Proceedings of AAAI90, 1990.

[15] D. Randell, A. G. Cohn and Z. Cui:[1992a] *Naive Topology: modeling the force pump.* in Recent Advances in Qualitative Reasoning, ed B Faltings and P Struss, MIT Press, in press, 1992.

[16] D. Randell and A. G. Cohn:[1989] "Modelling Topological and Metrical Properties in Physical Processes," in *Principles of Knowledge Representation and Reasoning,* ed. R. J. Brachman, H. Levesque and R. Reiter. Morgan Kaufmann, Los Altos. 1989

[17] D. A. Randell:[1991] *Analysing the Familiar: Reasoning about space and time in the everyday world* PhD Thesis, University of Warwick, UK 1991

[18] D. A. Randell, A. G. Cohn and Z. Cui:[1992b] "Computing Transitivity Tables: a Challenge for Automated Theorem Provers," *Proceedings of CADE11. Springer Verlag, 1992*

[19] D. A. Randell, Z. Cui and A. G. Cohn:[1992a] "An Interval Logic for Space based on 'Connection'." *in Proceedings of ECAI92*

[20] D. A. Randell, Z. Cui and A. G. Cohn:[1992b] "A Spatial Logic based on Regions and Connection," *to appear in Proceedings of KR92*

[21] D. Weld and J. de Kleer:[1990] "Readings in Qualitative Reasoning about Physical Systems," Morgan Kaufmann, San Mateo, 1990