Relevance Reasoning to Guide Compositional Modeling

Alon Y. Levy and Yumi Iwasaki

Knowledge Systems Laboratory
Stanford University
701 Welch Road, Bldg. C, Palo Alto, CA 94304

Hiroshi Motoda

Advanced Research Laboratory
Hitachi, Ltd.
Hatoyama, Saitama 350-03, Japan

Abstract

The ability to choose an appropriate manner in which to model a given device is crucial in making a compositional modeling [3] approach successful. In compositional modeling, a system is provided with a library of composible pieces of knowledge about the physical world, called model fragments, each representing a conceptually distinct phenomenon such as a physical process or one aspect of a component behavior. Given a specific query about a device, the system chooses among those model fragments to compose a model of the device that is most adequate to answer the query. Selection of appropriate model fragments can be viewed as a special case of a more general problem of reasoning about relevance of knowledge to a given goal. In this paper we pursue this view by applying a general framework for reasoning about relevance to the problem of model fragment selection. We show that heuristics for model selection can be usefully stated as irrelevance claims.

Employing such a framework allows one to state both general and domain-specific heuristics about relevance declaratively, as opposed to building them into the control structure of the system. Given relevance heuristics stated in the language, our relevance reasoning system can immediately make use of them to control the model formulation process, enabling us to experiment easily with different heuristics.

1 Introduction

The ability to choose an appropriate manner in which to model a given device is crucial in making a compositional modeling [3] approach successful in a complex domain. In compositional modeling, a system is provided with a library of composible pieces of knowledge about the physical world, called model fragments. Each model fragment represents a conceptually distinct phenomenon such as a physical process or one aspect of a component behavior. A knowledge base of a large complex domain can contain many model fragments representing alternative ways to model each phenomenon. Choosing the appropriate set of model fragments for a given problem is one of the most difficult tasks in compositional modeling. An appropriate choice of model fragments can lead to a correct answer efficiently, while an inappropriate choice can result in an inefficient, incorrect or no solution at all.

Selection of appropriate model fragments is a special case of a more general problem of reasoning about relevance of knowledge to a given goal. Subramanian & Genesereth [13] and Levy [9] have proposed general frameworks for reasoning about relevance of knowledge. This paper presents an application of Levy's framework to the problem of model fragment selection. The problem addressed is as follows: given a description of a physical system, and a specific question about some aspect of its behavior, how can a program select relevant model fragments to best answer the query. The selection must be sufficient to produce a correct answer to the question with the desired amount of detail. Choosing an appropriate model for a device involves deciding which abstractions of the domain should be made. The framework provides a set of formally defined primitive irrelevance claims, and shows how they serve as justifications for creating certain kinds of abstractions. We show that heuristics for model selection involving abstractions, including the heuristics underlying compositional modeling approaches proposed by Falkenhainer [3] and Nayak [11], can be stated using irrelevance claims.

There are important advantages to employing a general framework for relevance reasoning in model fragment selection. The framework allows one to state both general and domain-specific heuristics about relevance declaratively. In contrast, the model formulation programs developed so far have such heuristics built into their control structure. Given relevance heuristics stated in the language, our relevance reasoning system can immediately make use of them to control the model formulation process, enabling us to experiment easily with different heuristics.

In reasoning about physical system behavior, the works that have had most influences on our own are Qualitative Process Theory (QPT) by Forbus [4] and compositional modeling by Falkenhainer and Forbus [3]. Our representation of physical phenomena in the form of model fragments is based on the representation of processes and individual views in QPT. We will try to use the same terminology as used in [3] in this paper.

1.1 Model Fragments

In our system, knowledge about the physical world is organized into model fragments. Each model fragment represents a class of conceptually distinct physical phenomenon, such as a physical process, an object, or a component function, in terms of the conditions under which it takes place and the constraints and changes it will impose on the state of the world. In this scheme, the behavior of a device is modeled by a collection of model fragment instances, where each instance represents the different aspect of its behavior.

Formally, a model fragment is a predicate whose arguments are the formal parameters of the model fragment. If $a_1, \ldots, a_n$ are bindings for the formal parameters of a model fragment M, then $M(a_1, \ldots, a_n)$ means that the tuple $a_1, \ldots, a_n$ can be considered to be an instance of M. The model fragment will be *activated* only if its conditions are satisfied, and only then its content facts, $\mathcal{B}_M$, are included in the compositional model used to solve the query. Some model fragments describe continuous phenomena while other describe discontinuous phenomena. $\mathcal{B}_M$ of a continuous phenomenon specifies the functional relations among quantities that hold while it is taking place and the influences (increase, decrease) of the phenomenon on quantities. That of a discontinuous phenomenon specifies its consequences as assertions about the new state of the world, which we will call *Action*.

In order for a model fragment M to be activated, three types of conditions must be satisfied: instantiation conditions, $\mathcal{I}_M$, activation conditions, $\mathcal{O}_M$ and relevance conditions $\mathcal{A}_M$. $\mathcal{I}_M$ are conditions on the formal parameters of the model fragment. They identify the set of objects in the representation and relations between them that must exist in order for there to be an instance of the model fragment. $\mathcal{O}_M$ are conditions about the current scenario that must be satisfied for the model to be applicable, usually conditions on ranges of parameters in the model fragment. $\mathcal{I}_M$ and $\mathcal{O}_M$ only assure that the model fragment correctly describes the behavior of the mechanism modeled. Deciding to include the model fragment in the compositional model also hinges on its relevance to the query and appropriateness in the present problem solving context. For example, we may have several models of a battery, each describing a different aspect of its behavior, such as electrical, thermal and gassing properties, but not all being relevant to the current goal.

We use $\mathcal{A}_M$ to state heuristics for determining when the model fragment is relevant to a goal. They are meta-level statements about the representation of the device and the specific problem solving task. They concern the choice of objects that we need to represent for the specific problem and the distinctions that should be made in the representation in terms of granularity. In this document we concentrate on relevance conditions that state which abstractions are to be made during the search for the solution of the goal.

The key distinction between relevance conditions and the other conditions is that $\mathcal{A}_M$ are meta-level conditions that must hold in order for the model to be useful; i.e., conditions about the representation and about the problem solving task. $\mathcal{I}_M$ and $\mathcal{O}_M$ are base-level conditions.

2 Relevance Reasoning

Often, representations contain too much detail for a specific goal, either in the form of irrelevant facts in the theory of the domain or by containing irrelevant granularity distinctions in the domain. A powerful method to control search in such cases is by providing the problem solver with meta-level control advice about what is irrelevant to a given goal, (e.g., Lenat [7], Subramanian [14], Levy [9]). In simple cases, this advice might be to ignore a certain fact or set of facts, thereby pruning the solutions paths containing it. In other cases, we might advise the problem solver that certain granularity distinctions are irrelevant to the given goal, and therefore the represenation can be abstracted (e.g., for a certain goal it is not necessary to represent the subparts of a certain component, and it will suffice to represent the component by a single object).

Levy [9] describes a framework in which various notions of irrelevance are defined and analysed. The definitions of irrelevance differ along several axes, such as the kind of element being deemed irrelevant (e.g., single fact, object, predicate) and strength of the justification for the irrelevance claim. For example, a fact f can be defined to be *strongly irrelevant* to a goal g if it appears in no proof of g, or, *weakly irrelevant* if there is a proof of g that doesn't use f. Alternatively, we can define f to be irrelevant if it appears in no *minimal* proof of g.[1] In[9] we describe several such definitions and analize their properties. Irrelevance claims can either be automatically deduced by the system by examining the knowledge base (as in [10]) or they can be given to the system by the user, either as knowledge that the user has or as heuristics the user wishes the system to employ. In this paper we focus on stating model selection heuristics as relevance claims.

An irrelevance claim is a statement of the form $Ir(\alpha, g, \Delta)$, where α is the *subject* of irrelevance, g is a problem solving goal and Δ is the knowledge base (which we usually omit). α, as we describe below, can denote either a single fact, a predicate-symbol, object-constant, distinction between predicates, etc. To emphasize the different subjects we use specific predicate names (e.g., $IrObject$, $IrPredicate$, etc.).

In this paper, we are mostly interested in stating what *is* relevant, rather than what is not. We do so by stating $Rel(\alpha, g)$. We make the closed-world assumption on the predicate Rel, i.e., if we cannot conclude $Rel(\alpha, g)$ then we assume $Ir(\alpha, g)$[2]. The following property connects the relevance of a formula to the relevance of terms mentioned in it:

[1] Given some definition of minimum derivation [10].

[2] The closed world assumption was chosen for simplicity of exposition. More sophisticated non-monotonic reasoning methods can be employed. However, the closed world assumption has sufficed for our purposes thus far.

$$Wff(w) \wedge Mentions(w, \alpha) \wedge Rel(w, g) \Rightarrow Rel(\alpha, g),^{3} \tag{1}$$

i.e., if w is a relevant well formed-formula, and α (a formula or a term) appears in w, it is also relevant.

Irrelevance claims are justifications for the problem solver to modify the representation (and its search algorithm) such that it won't contain α. Using them, we define the meaning of the relevance conditions, $\mathcal{A}_M$ as follows:

$$\mathcal{A}_{M(x_1,\ldots,x_n)} \equiv Rel(M(x_1, \ldots, x_n), g) \tag{2}$$

$$Rel(M(\vec{x}), g) \Rightarrow Rel(\mathcal{I}_{M(\vec{x})} \wedge \mathcal{O}_{M(\vec{x})} \Rightarrow \mathcal{B}_{M(\vec{x})} \wedge M(\vec{x}), g). \tag{3}$$

It follows that if all $\mathcal{I}_M$, $\mathcal{O}_M$ and $\mathcal{A}_M$ hold, then $M(\vec{x})$ will be relevant as will its behavior constraints, $\mathcal{B}_M$. The problem solver will therefore include the instantiated model fragment in the compositional model. In what follows, we briefly present the different relevance subjects defined in the framework.

In the first set, the subjects of relevance are the basic elements of the representation. $RelFact(f, g)$ means that the fact f might be part of a deduction of the goal g, and therefore should not be ignored in search of a solution. $RelObject(o, g)$, $RelParameter(f, o, g)$ and $RelPredicate(P, g)$ say the same about an object-constant o, term $f(o)$ and predicate-symbol P, respectively. These claims are best understood as negations of their Ir counterparts, i.e., it is not justified to ignore f, o, $f(o)$ or P[4].

The following set of claims denote relevance of more abstract choices in the representation: The claim $RelObjDetail(o, R, g)$ denotes that the representation should contain the set of objects $\mathcal{O} = \{x | \; R(o, x)\}$, as opposed to only containing o. For example, in the case where $R = SubParts$, it states that both o and its subparts should be represented. The following is a simple consequent of the statement:

$$RelObjDetail(o, R, g) \wedge R(o, x) \wedge Rel(o, g) \Rightarrow RelObject(x, g). \tag{4}$$

However, $RelObjDetail$ implies more than the relevance of the finer level objects. Since some of the properties of o are defined by properties of elements of $\mathcal{O}$, these values are constrained by the values of properties of o. For example, the weight of an object is the sum of the weights of its subparts. The statement $ITA(R, F, o, o_1, c)$[5] denotes that the property $F(o)$ depends on some property of $o_1 \in \mathcal{O}$. The fifth argument gives the relation between

[3] Here, α must be either a subexpression, predicate symbol or term.

[4] $RelFact$, $RelObject$, $RelParameter$, and $RelPredicate$ are all specializations of Rel. Therefore, $RelFact(f, g) \Rightarrow Rel(f, g)$, $RelObject(o, g) \Rightarrow Rel(o, g)$, $RelParameter(f, o, g) \Rightarrow Rel(f(o), g)$, and $RelPredicate(P, g) \Rightarrow Rel(P, g)$. We will use the more specific predicates when we want to emphasize the type of the argument of Rel or when it is not clear from the context.

[5] ITA stands for Inherited Through Aggregate

$F(o)$ and a property of o_1. If it is omitted, we assume $F(o) = F(o_1)$. The following is a consequence of the formal definition:

$$ITA(R, F, o, o_1) \wedge RelObjDetail(o, R, g) \wedge Rel(F(o)) \Rightarrow Rel(F(o_1)). \tag{5}$$

A common case of such object aggregation is one where o denotes the set of objects in $\mathcal{O}$. For example, when reasoning about a chemical substance, only the sets of molecules of each type are relevant, not the specific molecules involved. $RelSetElements(S, g)$ ($IrSetElements$) denotes that the individual elements of the set S are (not) relevant to the goal g[6].

The claim $IrPredDistinction(P, \mathcal{P}, g)$ where $\mathcal{P}$ is a set of predicates and P is a predicate denotes that the representation should not contain the predicates in the set $\mathcal{P}$, but rather only contain a predicate P which is interpreted as the union of the interpretations of predicates in $\mathcal{P}$. For example, for many reasoning tasks it is not necessary to distinguish between properties such as $RechargeableBattery$ and $nonRechargeableBattery$. Instead, a predicate $Battery$ will suffice. This type of claim is a justification for *predicate abstraction* (Plaisted [12], Tenenberg [15]).

The claim $RelOnlySetRepresentative(S, g)$ denotes that the only properties relevant to the goal are those that are common to all elements of S, therefore it is enough to represent the set S by a representative member that has only these properties. $RelOnlyHomogenousSet$ denotes that the elements of the set S should be represented, but only as a homogeneous set, i.e., properties that distinguish between its elements should be ignored. $IrArgument(P, n, g)$ denotes that the nth argument of the predicate P is irrelevant to g.

Note that in general, using these relevance claims might require us to change the representation. For example, if $RelOnlySetRepresentative(S, g)$ is asserted, we need to add a new object that has all the properties which are common to all elements of S. The precise change of representation required for each relevance-predicate is described in [8]. However, in this document, we assume that the model fragments already contain the abstracted representation; therefore, we are using relevance claims as describers of abstractions rather than abstraction generators.

2.1 Relevance Heuristics

Using the above irrelevance claims, we can express heuristics for model selection, some of which are listed below. A term mentioned in the goal is relevant to it (this is the *query-expansion* heuristic used by Falkenhainer and Forbus [3]):

$$Goal(g) \wedge Mentions(g, o) \Rightarrow Rel(o, g) \tag{6}$$

[6] Another example of this aggregation is in the missionaries and cannibals problem (Amarel [2]), where only the sets of missionaries and cannibals are relevant to the problem and not their specific names.

The following two heuristics enable us to deduce relevance of components from the relevance of others by exploiting the structural hierarchies. They are similar to the ***object-expansion*** heuristic used in [3]. According to their heuristic, if s_1 and s_2 are both descendents of component s in the hierarchy, and their least common ancestor in the hierarchy is t, then any component that is either in between t and s_1 (s_2) or a child of such a component will be considered relevant. Our heuristic is more refined in that it only makes this inference across one level in the hierarchy. Stating the heuristic declaratively enables us to consider other refinements such as delimiting it to specific structural links, or to a certain class of objects. It can also be generalized to be equivalent to their heuristic.

$$StructuralHierarchySlot(P) \wedge Rel(x,g) \wedge RelObjDetail(x,P,g) \wedge P(x,y) \Rightarrow Rel(y,g) \quad (7)$$

$$StructuralHierarchySlot(P) \wedge P(x,y) \wedge Rel(x,g) \wedge Rel(y,g) \Rightarrow RelObjDetail(x,P) \quad (8)$$

An action of a discontinuous model fragment is the fact that is asserted in the subsequent state of the simulation as the consequence of it becoming active. We say that the model fragment *causes* that proposition, i.e.,

$$Action(M,\phi) \Rightarrow Causes(M,\phi). \quad (9)$$

A query of the form $Explain(\phi)$ might be given in a case where the simulation predicts that ϕ will hold, but the model used is not detailed enough.[7] We adopt the following simple axioms to establish a connection between $Explain$ and Rel.

$$Rel(\phi, Explain(\phi)) \quad (10)$$

$$Explain(\phi) \wedge InKB(\psi \equiv \phi) \Rightarrow Explain(\psi) \quad (11)$$

$$Explain(\phi \wedge \psi) \Rightarrow Explain(\phi) \wedge Explain(\psi) \quad (12)$$

$$Model(M) \wedge Causes(M,\phi) \wedge Explain(\phi) \Rightarrow Rel(M,g) \quad (13)$$

When two terms refer to the same object in the domain, the relevance of one of them implies the relevance of the other. We use corefer intuitively to state that two different terms actually refer to the same thing. For example, $PressureOf(g_1)$ refers to the same thing as $PressureIn(c_1)$ when g_1 is the gas contained in the sealed container, c_1. Such coreference statements are given explicitly in our knowledge base.

$$Rel(o_1,g) \wedge Corefer(o_1,o_2) \Rightarrow Rel(o_2,g). \quad (14)$$

Certain unary predicates are identified as *Type* predicates.

$$Type(x,P) \Rightarrow P(x) \quad (15)$$

[7]Sophisticated reasoning about explanation or about coreference is outside the scope of this paper, though there is considerable body of work on these topics in AI and philosophy.

The following heuristic says that if an attribute of an object is of a certain type, then the fact that it is of that type is relevant:[8]

$$Type(f(x), P) \wedge Rel(f(x), g) \Rightarrow Rel(P(f(x)), g) \tag{16}$$

The following heuristic says that if we are trying to explain a certain quantity that is inherited through an aggregate, the decomposition along that aggregate is relevant:

$$Explain(F(o)) \wedge ITA(R, F, o, o_1, c) \wedge Rel(o, g) \Rightarrow RelObjDetail(o, R, g). \tag{17}$$

3 Model Fragment Selection Example

In this section, we present an example in which the relevance heuristics presented in Section 2.1 are used to select appropriate model fragments to be considered by a modeling program. The particular modeling program that we use is Device Modeling Environment (DME) [6], developed at Stanford. Given the topological description of a device and initial conditions, DME formulates a mathematical model and simulates its behavior. DME has a knowledge base of model fragments. DME takes an input description of the initial state, including the topological model of the device, and searches the knowledge base for model fragments that are applicable to the given situation. Equations to describe the behavior of the device are formulated from the set of model fragments thus found. The equations are used to predict the behavior of the device. During prediction, if there are any changes in the set of applicable model fragments, the set of equations is updated accordingly and prediction continues with the new equation model.

The problem domain is a rechargeable, nickel-cadmium battery. The battery is a constant voltage source when the charge level is in its normal range. Otherwise, the voltage generated by the battery increases or decreases as it is charged or discharged. When the battery is over-charged beyond a certain point, a pressure increase in the battery causes the cell to explode. This pressure increase is caused by the hydrogen gas generated in the battery.

In DME's knowledge base, there are a number of model fragments describing different behavioral aspects of a nickel-cadmium battery, such as the electrical, chemical or thermodynamic properties. Table 1 shows some model fragments in the DME knowledge base. Depending on the question posed by the user about the battery, the system must choose an appropriate set of model fragments to consider in formulating a model. We show how this is done through reasoning about relevance of model fragments to the problem in hand.

The following are domain facts needed for the exposition of the problem solving scenarios:

$$ElectricProperty(ChargeLevel) \tag{18}$$

[8]Notice this is one exceptional case in which relevance of an expression is implied by the relevance of a subexpression.

$$GasParts(x, GasIn(x)) \tag{19}$$

$$Type(ChassisOf(x), Container) \tag{20}$$

$$StructuralHierarchySlot(GasParts) \tag{21}$$

$$GasParts(x, ChassisOf(x)) \tag{22}$$

$$Type(GasIn(x), Gas) \tag{23}$$

$$Constituents(x, HydrogenIn(x)) \tag{24}$$

$$Constituents(x, OxygenIn(x)) \tag{25}$$

$$Battery(x) \wedge \neg Sealed(ChassisOf(x)) \equiv Damaged(x) \tag{26}$$

$$Action(SealedContainerRuptureModel(x), \neg Sealed(x)) \tag{27}$$

$$Corefer(pressureIn(ChassisOf(x)), pressureOf(GasIn(x))) \tag{28}$$

$$\begin{gathered} Constituents(x, y) \wedge Gas(x) \Rightarrow \\ ITA(Constituents, MassOf, x, y, MassOf(x) = \\ \textstyle\sum_{y \in Constituents(x,y)} MassOf(y)) \end{gathered} \tag{29}$$

Suppose we are given an instance of an EPS system that includes *Bat001* and a query
$g = ChargeLevel(Bat001, t)$,
where t is the number of cycles (i.e., days) for which the battery is operating. This will lead us to conclude relevance of *BatteryNormalOperatingModel* and *BatteryOverchargeOperating-Model*. Therefore, they will be activated depending on their activation conditions, i.e., the *StoredCharge* of the battery. The proof tree for their relevance is shown in Figure 2.

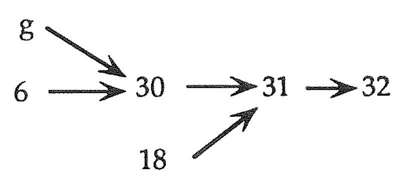

Figure 1: Proof tree for Axiom 32

The numbers refer to the axioms presented so far and those following:

$$Rel(ChargeLevel(Bat001, t)) \tag{30}$$

$$Rel(BatteryChargingDischargingModel(Bat001)) \tag{31}$$

$$Rel(BatteryNormalOperatingModel(Bat001)) \wedge$$
$$Rel(BatteryOverchargeOperatingModel(Bat001)) \quad (32)$$

Now, suppose the battery is overcharged and becomes damaged. In this case, we might ask the system to explain why the battery is damaged by posing the goal
$g_1 = Explain(Damaged(Bat001))$.
This leads us to conclude relevance of *BatteryGassingHydrogenModel, PressureAspectFluidContainerModel, GasPressureIdealModel,* and *SealedContainerRuptureModel* through the proof tree shown in Figure 3. The intermediate axioms are as follows:

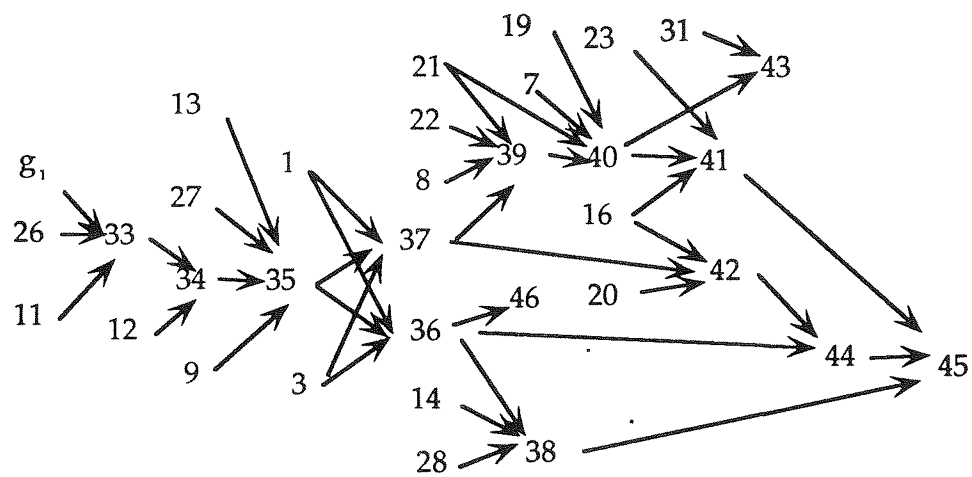

Figure 2: Proof tree for Axioms 43 through 46

$$Explain(Battery(Bat001) \wedge \neg Sealed(ChassisOf(Bat001))) \quad (33)$$
$$Explain(\neg Sealed(ChassisOf(Bat001))) \quad (34)$$
$$Rel(SealedContainerRuptureModel(ChassisOf(Bat001))) \quad (35)$$
$$Rel(PressureIn(ChassisOf(Bat001))) \quad (36)$$
$$Rel(ChassisOf(Bat001)) \quad (37)$$
$$Rel(PressureOf(GasIn(Bat001))) \quad (38)$$
$$RelObjDetail(Bat001, GasParts). \quad (39)$$
$$Rel(GasIn(Bat001)) \quad (40)$$
$$Rel(Gas(GasIn(Bat001))) \quad (41)$$
$$Rel(Container(ChassisOf(Bat001))) \quad (42)$$

$$Rel(BatteryGassingHydrogenModel(Bat001)) \tag{43}$$

$$Rel(PressureAspectFluidContainerModel(ChassisOf(Bat001))) \tag{44}$$

$$Rel(GasPressureIdealModel(GasIn(Bat001))) \tag{45}$$

$$Rel(SealedContainerRuptureModel(ChassisOf(Bat001))) \tag{46}$$

4 Discussion

We have presented a framework in which a problem solver can reason about the choice of relevant parts of the knowledge base that are to be used to solve a given query. We demonstrated the use of this general framework for the task of model fragments selection. The language we have presented enables us to express naturally and declaratively the heuristics about model selection in compositional modeling.

There are related works on model formulation [3, 11, 1] and relevance reasoning [13]. Space limitation allows us to discuss only one of them. Falkenhainer and Forbus' procedure for selecting model fragment consists of four steps: (1) query analysis, (2) object expansion, (3) candidate completion, and (4) candidate evaluation and selection. Step (1) identifies from the query the set of relevant objects and terms. Step (2) uses part-of hierarchy of objects to include all the components of relevant objects. Step (3) generates all the internally consistent and complete sets of model fragments. Step (4) chooses one among the sets based on their simplicity and estimated cost.

The heuristics (7) and (8) in Section 2.1 demonstrated that the strategy behind query analysis and object expansion can be formalized as relevance heuristics. Our relevance framework enables one to formulate any such heuristics and make use of them immediately in problem solving. Falkenhainer and Forbus' concept of an assumption class, which is a set of assumptions about how an object is to be modeled such that the assumptions are mutually exclusive but one of them must be made, can be formalized in our framework simply by the exclusive-or of the relevance of the elements of an assumption class. Where their system produces several consistent models, our framework would yield a disjunction of sets of relevance claims, each representing (or entailing) a consistent model.

This document has focussed only on relevance claims that result in abstractions. Abstractions have traditionally been divided into two kinds, truth preserving (TD abstractions [5]) and completeness preserving (TI). The irrelevance claims we presented here only account for the former kind. The reason is that an irrelevance claim can only justify ignoring some of the knowledge available and therefore, (when dealing with a monotonic representation language), should not enable us to draw new conclusions that wouldn't follow in the original theory. However, our framework also allows us to state other meta-level claims about the problem solving scenario, such as approximation-claims concerning the accuracy of the desired solution or the time period of the simulation. These in turn enable us to state other types of assumptions made by model fragments.

The fact that we can explicitly state the representational choices being made allows us to state heuristic rules that connect between choices. For example sometimes, a decision to consider a finer level of granularity in one subtree of a structural hierarchy implies that we should do the same in a sibling subtree. This heuristic can be formalized as follows.

$$R(c_1, c_2) \wedge Rel(R, g) \wedge Rel(c_1, g) \wedge Rel(c_2, g) \wedge RelObjDetail(c_1, P, g) \wedge P(c_1, x_1) \wedge$$
$$ITA(P, R, c_1, x_1, constraint_1) \wedge P(c_2, x_2) \wedge ITA(P, R, c_2, x_2, constraint_2) \wedge$$
$$DefinedBy(R, R') \Rightarrow RelObjDetail(c_2, P, g)$$

i.e., if a relation R' on the parts of c_1 explains the relation R on c_1, and R is relevant and the distinction between c_1 and its subparts is relevant, then the distinction between c_2 and its subparts should be relevant, too. In our example, this would mean that we should explore the constituents of the liquid in the battery when we decide to explore the constituents of the gas. The simple object expansion heuristic of Falkenhainer and Forbus [3] does not capture that dependency, but our language allows us to state it. Identifying these primitive relevance claims also helps us in guiding the search for additional heuristics. Stating those heuristics declaratively, as opposed to wiring them into a model formulation procedure, allows the user to inspect them and modify them easily (for example, by adding qualifications as needed). Finally, since the framework allows the abstractions made in a model fragment to be explicitly stated, the problem solver is able to reason with them and to choose abstractions tailored for the specific task at hand, as opposed to being constrained by predefined abstraction hierarchies.

The most important source of relevance heuristics has been ourselves. We try to articulate the heuristics we seem to be using when formulating models. We consider their utility and generalize/specialize them to make them more useful or accurate. We have also gleaned some heuristics from other works on model formulation in qualitative physics. Learning heuristics automatically from problem solving experience is another possibility we plan to investigate. Since relevance claims provide crisp criteria for when abstractions should be done, this problem is now better formulated.

5 Acknowledgements

The authors would to thank Rich Fikes, Pat Hayes and Pandu Nayak for many enlightening discussions about the contents of this paper. The first author would like to thank the Hitachi Advanced Research Laboratory for their support in pursuing this work.

Model Fragment	Instantiation Condition	Relevance Condition	Description*
BCDM**	RechargeableBattery(b) Λ ~Damaged(b)	Rel(p(b)) Λ ElectricProperty(p)	Charging and discharging behavior of a battery. Stored-charge of the battery changes depending on the current through the plus terminal.
Battery-normal-operating-model	BCDM(b)	Rel(BCDM(b))	Voltage is constant when Stored-charge is between 6.0 and 30.0 amp-hours.
Battery-overcharge-operating-model	BCDM(b)	Rel(BCDM(b))	Voltage increases with Stored-charge when Stored-charge is over 30.0 amp-hours.
Battery-damaged-during-overcharge-model	BCDM(b)	Rel(BCDM(b))	The battery is damaged if Stored-charge reaches 34.0 amp-hours.
Aging-model	BCDM(b)	Rel(BCDM(b)) Λ TPOG(g, t) Λ t > T0	The battery capacity decreases if the maximum depth of discharge is less than 20% over a long period of time.
Battery-gassing-hydrogen-model	BCDM(b)	Rel(BCDM(b)) Λ Rel(GasIn(b))	The hydrogen in the battery increases when Stored-charge keeps increasing over a threshold.
PASFCM***	Container(c) Λ Sealed(c)	Rel(PressureIn(c)) Λ Rel(Container(c))	The model of the pressure aspect of a sealed fluid container. Active when the amount of fluid is non-zero.
Gas-pressure-ideal-model	Gas(GasIn(c)) Λ PASFCM(c)	Rel(Gas(GasIn(c)) Λ Rel(PASFCM(c)) Λ {Rel(Pressure-of(GasIn(c))) v Rel(Temperature-of(GasIn(c))) v Rel(Mass-of(GasIn(c))) v Rel(Gas-constant-of(GasIn(c)))}	The model of deal gas in a sealed container. The ideal gas law holds.
Sealed-container-rupture-model	Container(c) Λ Sealed(c)	Rel(Pressure-in(c)) v Rel(Sealed(c)) Λ Container(c)	A container ruptures and becomes un-sealed when the pressure reaches a threshold.
Hydrogen-production-by-overcharge-model	Battery-overcharge-operating-model(b)	Rel(HydrogenIn((GasIn(b))) v Rel(HydroxilIonIn(b)) v Rel(WaterIn(b))	When an over-charged battery continues to be charged, the water decreases, the hydrogen increases, and the hydroxil ion increases in the battery.

* For lack of space, the activation condition and the behavior of each model is briefly described.

** Battery-charging-discharging-model

*** Pressure-aspect-fluid-container-model

Table 1: Model Fragment Examples

References

[1] Sanjaya Addanki, R. Cremonini, and J. Penberthy. Reasoning about assumptions in graphs of models. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.

[2] Saul Amarel. On representations of problems of reasoning about actions. In Bonnie L. Webber and Nils J. Nilsson, editors, *Readings in Artificial Intelligence.* Morgan Kaufmann, Los Altos, CA, 1981.

[3] Brian Falkenhainer and Ken Forbus. Compositional modeling: Finding the right model for the job. In *Artificial Intelligence*, Vol. 51, pp. 95-143, 1991.

[4] Ken Forbus. Qualitative process theory. In *Artificial Intelligence*, Vol. 24, 1984.

[5] Fausto Giunchiglia and Toby Walsh. A theory of abstraction. To appear in the Journal of Artificial Intelligence, 1992.

[6] Yumi Iwasaki and Chee Meng Low. Device modeling environment: An integrated model-formulation and simulation environment for continuous and discrete phenomena. In *Proceedings of Conference on Intelligent Systems Engineering*, 1992.

[7] Douglas B. Lenat, Randall Davis, Jon Doyle, and Michael R. Genesereth. Reasoning about reasoning. In Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat, editors, *Building Expert Systems.* Addison Wesley, Reading Mass., 1983.

[8] Alon Y. Levy. Creating abstractions using relevance claims. In preparation, 1992.

[9] Alon Y. Levy. Irrelevance in problem solving. Proceedings of the Workshop on Change of Representation, Asilomar, California., 1992.

[10] Alon Y. Levy and Yehoshua Sagiv. Constraints and redundancy in datalog. In *The Proceedings of the Eleventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, San Diego, CA.*, 1992.

[11] P. Pandurang Nayak, Leo Joskowicz, and Sanjaya Addanki. Automated model selection using context-dependent behaviors. In *Proceedings of the Fifth International Workshop on Qualitative Reasoning about Physical Systems*, May 1991.

[12] D. Plaisted. Theorem proving with abstraction. In *Artificial Intelligence*, pages Vol. 16, pp. 47–108, 1981.

[13] D. Subramanian and M.R. Genesereth. The relevance of irrelevance. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Los Altos, CA, 1987. Morgan Kaufmann.

[14] Devika Subramanian. A theory of justified reformulations. In *Ph.D thesis, Dept. of Computer Science, Stanford University*. Stanford, CA, 1989.

[15] Josh D. Tenenberg. Preserving consistency across abstraction mappings. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1011–1014, Los Altos, CA, 1987. Morgan Kaufmann.