Automatic Enhancement of Model Parsimony *

Cis Schut
Bert Bredeweg

Department of Social Science Informatics (S.W.I.)
University of Amsterdam, Roetersstraat 15
1018 WB Amsterdam (The Netherlands)
Telephone: +31-20-525 6801, Telefax: +31-20-525 6896
E-mail: {schut,bert}@swi.psy.uva.nl

January 8, 1993

Abstract

This paper presents a technique that supports a knowledge engineer in constructing a qualitative model of a physical device. The technique is specifically aimed at achieving a parsimonious model of device behaviour. First we explain why a model that is composed of general model fragments may still contain irrelevant particles. Then we present a technique that removes these irrelevant model particles from the model fragments. The technique assumes that, based on a set of applicable model fragments, the behaviour of the device can be modelled although the description may contain irrelevant model particles. It takes this behaviour description as input and for manipulations of specific parameters identifies the irrelevant particles. Both knowledge concerning the relevance as well as the irrelevance of model particles is used for this purpose. We further employ the distinction between overly detailed and superfluous model particles. Finally, modification of the model fragments yields a behaviour simulation with better computational properties and more clarity.

*The research presented in this paper was partially supported by the Foundation for Computer Science in the Netherlands (SION) with financial support from the Netherlands Organisation for Scientific Research (NWO), project number 612-322-307. We would like to thank Jaap Kamps for the stimulating discussions on the research presented in this paper.

1 Introduction

Modelling system behaviour for simulation purposes is generally conceptualised as proceeding from a complex base model to a simplified "lumped model", tuned appropriately to the requirements of the task for which it is constructed [11]. When constructing a qualitative model that adequately predicts the behaviour of a physical system [10] this tuning process entails both a modelling and a simulation activity, which are repeatedly performed in an alternating cycle. Typically, it is the selection of a set of applicable pieces of general domain knowledge, ususally represented in a library as canonical model fragments [3; 6], that initiates model construction. Subsequently, this general model is refined in a cyclical process of model simulation and adaptation of the applicable model fragments. Adaptations are based on differences between the required or expected output of the simulation and the actual output. The aim of our research is to support this refinement process by providing a modeller with tools that automate basic steps in this refinement process.

An important reason why a set of general prototypical model fragments may not satisfactorily model device behaviour is that they describe more domain knowledge than is strictly necessary for predicting the behaviour of the device. This paper presents a technique that automatically removes irrelevant detail from a behaviour prediction. In particular, we formulate principles that generate parsimonious model fragments by modifying canonical ones. The technique uses three types of input: (i) an input system (or scenario) that describes the device at hand with its starting conditions, (ii) a set of general model fragments that apply to it, and (iii) a simulation of its qualitative behaviour. Provided with these inputs, the technique identifies superfluous and overly detailed model particles by imposing three criteria on the model: (i) the desired behaviour prediction should be attainable, (ii) it should be physically appropriate, and (iii) it should be cognitively comprehensive.

2 Automatic Model Construction

The technique described in this paper is part of a set of tools that is being developed to support a knowledge engineer in the construction of qualitative models of device behaviour. This modelling process can be viewed as consisting of a number of tasks (see figure 1). An important step

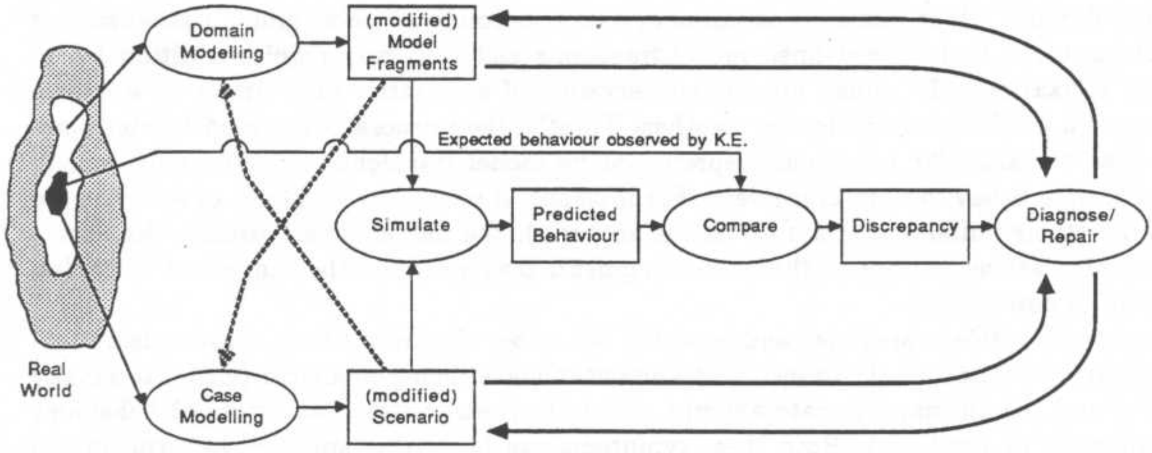

Figure 1: Tasks in Constructing Qualitative Models of Device Behaviour

in model construction is the representation of general domain knowledge in model fragments, possibly stored in some kind of library to ensure that they are reusable in different model building enterprises. Examples of model fragments that may result from this *domain modelling* effort are "individual views" and "processes", as defined by the process-centred approach [6] and "qualitative states of component behaviour", as defined by the component centred approach [3].

A second important step is the construction of the scenario that specifies the important aspects of the device for which the qualitative behaviour model is being developed. The scenario that results from this *case modelling* typically provides information about the physical structure of the device. It may also specify some initial values for parameters that are relevant for the simulation.

Case and domain modelling are not independent. The output of case modelling may influence the domain modelling and vice versa. The identification of certain physical objects may require the construction of model fragments that represent their behaviour(s). Or, the other way around, the presence of certain model fragments may lead to the identification of specific physical objects. In general, it is important that the outputs of the two modeling steps relate to each other, that is, the behaviours modelled by the model fragments must map onto the device specifications given by the scenario.

When a significant amount of domain and case modelling has been carried out, a qualitative simulator can be used to derive the expected behaviour of the device. Next, the knowledge engineer *compares* the output of the qualitative reasoner to the behaviour of the device in the real-world. Often there are discrepancies between the predicted and the expected device behaviour, in particular in the initial phase of the model building process. Therefore a *diagnose/repair* step is required that (i) determines the faults that cause these discrepancies and (ii) modifies the model fragments and/or the scenario in such a way that the discrepancies between the predicted and expected behaviour disappear.

Usually the modelling process is cyclic, which means that the different tasks will be carried out more then once, possibly in (slightly) different orders. Support can be provided for various aspects of this modelling process. The technique presented in this paper is concerned with the diagnose/repair step.

There are a number of causes that can explain possible discrepancies between predicted and expected behaviour. In the case of model fragments, for example, refinement may typically be required for the following reasons (this list is not complete). Firstly, due to idiosyncrasies of devices canonical model fragments may be too general and therefore need augmentation or they may be too detailed which demands omission of detail. Secondly, specific global configurations cannot be captured by locally defined model fragments and thus may require additional constraints or relaxations. Examples are the conservation of a quantity in a circuit or a known dominance of a particular variable over another. Thirdly, the purpose of the model determines the relevance of particular behaviours represented by model fragments. Usually only certain aspects of device behaviour are of interest. For instance, if particular variables of a device are manipulated (their initial values are set in the scenario), the model should predict the effects of these manipulations. Although the model fragments may refer to other unrelated variables, these are of no importance.

Discrepancies between predicted and expected behaviour that result from these causes manifest themselves in terms of: (i) an incorrect number of states being predicted (either too many or too few) and (ii) an inappropriate amount of detail present within each state of behaviour (either too much or too little). Both these symptoms can be further specified in terms of the knowledge representation that is being used (see section 3). Unfortunately, the discrepancies (or symptoms) that result from different causes are not independent. For example, too many states of behaviour may result from too detailed model fragments, but may also be caused by ambiguity that results from lacking knowledge about global behaviours. In addition, multiple faults may cause discrepancies, which makes the problem even worse.

Although our aim is to automate the diagnose/repair process as a whole, this is currently not feasible. We therefore have to take some pragmatic choices in order to get started. Our strategy is to develop individual techniques for *each cause.* These techniques should be able to modify the model fragments and/or scenario in such a way that the discrepancies that result from the underlying cause are removed. The techniques operate under the single fault assumption [4]. The knowledge engineer is supposed to point out the fault that has to be tackled. The knowledge engineer also has the ability to overrule the modifications proposed by a technique if required.

In [2] we have described a technique for removing spurious behaviour caused by missing knowledge about global behaviour properties of a device. The technique presented in this paper deals with "enhancement of model parsimony". It assumes that the right set of applicable model fragments has been selected and that a correct, i.e. desired, simulation of device behaviour has been achieved. Although this simulation is correct it may still contain irrelevant aspects: parameters that never cause transitions between states, details in the physical structure of the device that are unnecessarily distinguished, quantity spaces that are too detailed, etc. The technique identifies and removes such irrelevant detail from the qualitative model.

3 Representational Context

This section briefly describes the important aspects of the framework for qualitative prediction of behaviour that we use. This framework is implemented as a domain independent qualitative reasoning shell, called GARP, which can be used by a knowledge engineer for developing prediction models [1]. Similar to the component [3] and the process [6] oriented approach, GARP uses the notion of model fragments for determining the behaviour of some real-world system. All model fragments have associated with them a set of conditions under which they are applicable and a set of consequences that are given once their conditions hold. Conditions and consequences are stated in terms of **model particles**: *system elements:* abstractions of entities in the physical world, such as containers, substances, components, etc.; *parameters:* quantities describing properties of system elements, such as pressure, temperature, voltage, etc.; *parameter values:* the values (intervals and derivatives) of quantities, such as $\{-, 0, +\}$, and others; *parameter relations:* relations (or constraints) between quantities, such as inequalities, proportionalities, influences, etc.; *model fragments:* other model fragments, such as views, processes, component models, etc., that hold or must be true. For example, a heat-flow process that must be present in order for a boiling process to be applicable.

The behaviour of a system during a particular time period is described by the set of applicable model fragments. The behaviour over different time periods is determined by the application of transformation rules between states of behaviour (further details can be found in [1]).

4 Enhancement of Model Parsimony

The two main benefits of omitting irrelevant model particles are the increased efficiency of the simulation (computation of irrelevant parameter values is no longer needed) and the increased clarity in the description of device behaviour (irrelevant model particles no longer need to be interpreted).

This can be illustrated by a problem involving a balance with each arm supporting a container filled with water (see figure 2). The water gradually flows out of the containers through an outlet

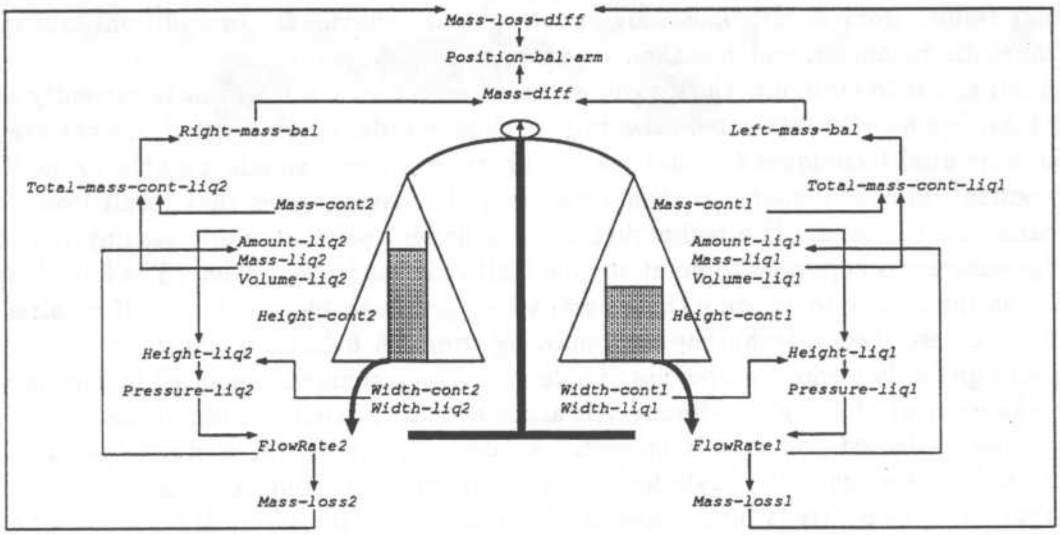

Figure 2: A General Model for the Balance Problems

at the bottom of the containers. The containers have equal masses but their shapes and the amounts of water they contain are different.

The balance problem can be modelled with prototypical model fragments, such as a *contained_liquid, liquid_flow* and a *balance.* This gives rise to the model depicted in figure 2. This model predicts the behaviour of the balance correctly. However, it contains many parameters that are not strictly necessary for deriving the correct behaviour description. In fact, the same behaviour description can be generated with parsimonious model fragments yielding a model as visualised in figure 3.

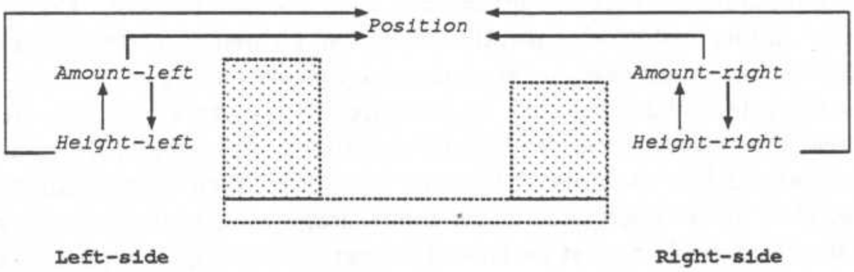

Figure 3: A Parsimonious Model for the Balance Problems

Our goal is to automate the process of generating a set of parsimonious model fragments from a set of general prototypical ones. To do so we must first identify the requirements for such a parsimonious model. We postulate three requirements:

Qualitative attainability A parsimonious model must generate the desired behaviour prediction and therefore needs a minimum set of labels. This minimum set of labels reflects the requirements of the qualitative machinery: i.e. the labels should allow the qualitative inference machinery to derive all states of behaviour.

Physical correctness In order to make sense from a physics viewpoint the labels should be instantiated with physically sensible names. Moreover, the physics viewpoint may require the introduction of additional model particles to render the model physical correctness.

For the qualitative reasoner it may, for example, at some point be sufficient to use a variable that influences itself and thereby generates the required states. Thus the behaviour description is qualitatively attainable, but from a physics viewpoint the model is too abstract and conceptually wrong.

Cognitive comprehensibility On top of the required physical correctness there are requirements with respect to the "cognitive comprehensibility". Particular modelling goals that have a cognitive origin may require further introduction or preservation of model particles. For instance, a modeller may explicitly want the balance model to include the pressure parameter for explanation purposes.

5 Identification of Irrelevant Model Particles

The parsimonious model fragments that are generated by the procedure are applicable to (i) the device as specified in the original scenario, and (ii) manipulations of parameters that were manipulated in the original scenario (for which the correct behaviour description was generated). For a different device structure or if other parameters are manipulated a different set of model fragments might apply, possibly making other model particles relevant or irrelevant.

Irrelevance can exist in two forms: superfluous model particles and particles that are overly detailed. Superfluous model particles refer to instances of modelling primitives that are not used in deriving the behaviour prediction: the behaviour prediction is not influenced by their presence or absence. Overly detailed particles, on the other hand, are irrelevant even though they are used in the prediction. Their irrelevance is caused by the fact that their functionality can be subsumed under the functionality of other particles.

Basically, the procedure for achieving parsimonious model fragments consists of determining which particles are irrelevant and modifying the set of applicable model fragments. Determing the irrelevant model particles is done by an iteration of applying a set of rules specific for each type of model particle. These rules specify the conditions under which a particle of a particular type, say a parameter, is either relevant, overly detailed or superfluous.

There are two complementary directions for determining which particles are irrelevant: identifying particles that are definitely relevant, and thus are not irrelevant, and identifying particles that are irrelevant. The information that can be used for this purpose is threefold: (i) the scenario for which a behaviour simulation is generated, (ii) the model fragments that were used to derive the desired behaviour description, and (iii) the desired behaviour description itself. For instance, the desired behaviour description can be used to detect relevant particles without which the prediction could not be generated.

The order of detecting irrelevant particles is important because irrelevance has the property of non-monotonicity (cf. [9]). Parameters are the key notion in a behaviour description because values and relations depend on them. Therefore, the identification of irrelevant values and relations depends on the identification of irrelevant parameters. In fact, after irrelevant parameters have been removed, their values and relations become superfluous.

System elements and model fragments are conditional for the applicability of model fragments. These fragments introduce parameters, and constraints on their behaviour, to the overall model. A model fragment becomes irrelevant when it does not introduce anything that is relevant for the behaviour description. However, this can only be established after the model particles within the model fragment have been identified as irrelevant. This is why the identification of irrelevant model fragments and system elements should take place after identifying irrelevant parameters, relations, and values.

5.1 Surplus Parametrical Detail

Surplus parametrical detail occurs if the effects of several parameters can be summarised by one parameter. The global procedure in identifying these parameters starts with identifying definitely relevant parameters, then identifies the overly detailed parameters and finally identifies the superfluous parameters. The remaining ones are all assumed relevant to be on the safe side.

5.1.1 Relevant Parameters

There are two sources for determining the definite relevance of parameters: the desired behaviour prediction and the scenario. According to the requirement of qualitative attainability a requirement of the new parsimonious model is that the same states of behaviour are predicted as well as the same state transitions. Therefore, if a state transition in the old model is induced by *one* parameter taking on a new qualitative value or by *one* inequality change of two parameters then it can be concluded that the involved parameter(s) are relevant: leaving out parameters with these properties would inevitably lead to another set of states, which violates the requirement. In other words, these parameters are relevant in order to guarantee qualitative attainability.

The second source for determining the relevance of a parameter is the scenario. In GARP parameters may be specified in the scenario for three reasons: (i) for specifying a parameter of interest that represents a query about the device, (ii) for specifying parameters that are manipulated externally (representing an enforced disturbance from equilibrium), and (iii) for specifying values of parameters that represent particular conditions under which a device operates. The first two render a parameter definitely relevant whereas the third type is not conclusive. The reason for this is as follows. In the case of a queried parameter, it is fair to conclude that this parameter is relevant on cognitive grounds: apparently the knowledge engineer is interested in the behaviour of these parameters. In the second case a parameter is relevant from a physical viewpoint. The parameter is assigned an initial value that triggers the device behaviour. To ascertain that an equilibrium disturbance is enforced it must be verified from the model fragments in the library that the parameter depends on other parameters. If the latter condition is not satisfied the parameter is used to specify an operating condition, which reflects the third case. In this case the parameter might be irrelevant if one of the rules described below applies to it.

5.1.2 Overly detailed Parameters

Once the relevant parameters have been selected the overly detailed parameters are identified. Overly detailed parameters can occur as intermediating variables that merely pass on values or as non-changing parameters that reflect constant conditions under which behaviour takes place. These cases are signaled by two phenomena in the behaviour description: if two or more parameters show identical state transitions or if a parameter is never involved in a state transition. In the first case transitions between states involve sets of parameter value changes that always go together whereas changes concerning individual parameters never occur. In cases like these it is reasonable to assume that the parameters involved are functionally equivalent and thus that the set of parameters can be replaced by a single parameter. However, parameter equivalence cannot be derived from covariance alone, due to lack of knowledge of system behaviour had the manipulated parameters been assigned different values in the scenario. Hence, the model fragments are inspected in order to find evidence for the irrelevance conjecture. For parameters to be equivalent their quantity spaces must correspond since transitions of one parameter have to be mirrored in the other(s). In the balance problem the parameters *height, pressure* and *flow_rate* among others show identical state transitions.

After identifying sets of equivalent parameters, the next step is to find/choose the parameter that subsumes the others. Here the set of already identified relevant parameters can provide guidance. If among the set of equivalent parameters there is at least one relevant parameter then the whole set is irrelevant because the relevant parameter already models the behaviour of the equivalent ones. If there is no relevant parameter in the set of equivalent ones then a random choice suffices. However, physical and/or cognitive grounds may be used by the knowledge engineer to choose the subsuming parameter.

The second situation in which a parameter is overly detailed is when it never occurs in a state transition and can never occur in one (it represents an operating condition for device behaviour). If a parameter is not involved in qualitative changes, it does not contribute to the behaviour prediction process. To establish this it should be verified from the model fragments that the parameter does not depend on any other parameter and that at least one other parameter depends on it (if not it would be superfluous). For instance in the balance problem, the width of the liquid column is equal to the width of the container. The latter is constant and does not depend on any other parameter, so the width of the container can be subsumed. Another example is the mass of the container, which remains constant, and can be subsumed under the total mass of the contained liquid, which is the sum of the masses of the container and the liquid.

5.1.3 Superfluous Parameters

With the above rules all relevant parameters and all overly detailed parameters can be identified. At this point the superfluous parameters, the ones not used in the prediction, should be identified. Parameters are superfluous if they do not show changing behaviour, i.e. do not appear in any of the causes that induce a state transition, and do not appear in the set of relevant parameters. In other words, they should not have any relations with other parameters. An example of this is the height of the container.

5.2 Other Irrelevant Model Particles

Relations and Values Relations involving superfluous parameters become irrelevant because they have no reference any longer. In relations between a relevant parameter and an overly detailed parameter, the latter can be replaced by the parameter that subsumes its functionality. Values referring to irrelevant parameters also become irrelevant.

System Elements To achieve parsimony in a model it should only address elements in the device that need necessarily be distinguished. For instance, in the balance problem it is not necessary to reason explicitly about the containers on the balance arms that contain the water. None of their properties changes so one can as well forget about them and only reason about the changes in the liquid.

Too much detail in system elements can only occur if they are used for attaching parameters that turn out to be irrelevant. Since all irrelevant parameters have already been identified the problem is reduced to determining which elements are superfluous. Again, as in the case of parameters, the definite relevance of elements can be established by checking all state descriptions: elements not appearing in all states are definitely relevant: leaving out such an element would inevitably lead to another set of states. Elements that have associated relevant parameters are also relevant.

Irrelevant elements, on the other hand, can be identified from the set of applicable model fragments as follows. For system elements to be superfluous two conditions should hold: (i) they should not have any associated relevant parameters, (ii) their removal should not make the conditions of a model fragment empty. An exception to the last condition is the case where the model fragment specifies no consequences and it is not conditional for any other model fragment.

Model Fragments The detection of irrelevant model fragments is a very complex matter. Until now we have not worked this out satisfactorily. The complexity is due to the many possible interdependencies between model fragments, some having conditional other model fragments and so on. However, in some situations we can decide to take some model fragments together or to render them superfluous. For instance, if in all states two or more model fragments are applicable to one or a group of elements they might be joined to form one composite model fragment. This can be done if another condition holds: none of these model fragments is seperately applicable to another element or seperately conditional to another model fragment. A model fragment is superfluous if all its givens are denoted irrelevant (it adds nothing to the behaviour description) and it is not conditional for any other model fragment. These are two cases for which the irrelevance of model fragments can be established relatively easy. We are still analysing how to deal with more complex cases where, for instance, conditions may be moved from two model fragments to another model fragment.

5.3 Updating the Model Fragments

When all irrelevant particles have been identified control is switched to the knowledge engineer who makes the final decision concerning the actual removal of the particles. Based on these decisions the model fragments are modified. Currently we are in the process of defining general procedures for executing model fragment modifications. These procedures are not specific for achieving model parsimony but are developed for model refinement in general (see section 2) and are therefore not further elaborated upon.

6 Related Work and Conclusions

In Subramanian & Genesereth [9] a formal treatment of irrelevance is presented. They distinguish three types of irrelevance: weak, strong and computational. Our notion of overly detailed model particles corresponds to computational irrelevance: the model without the irrelevant particles has better computational characteristics than the model containing those model particles. Our notion of superfluousness corresponds to strong irrelevance: a superfluous particle is not necessary for deriving the state sequence and the state sequence is not necessary for the truth of the particle. Although Subramanian and Genesereth put irrelevance reasoning on firmer ground, they observe that the acquisition of irrelevance facts in a specific domain remains an open question. In this paper we have identified under which conditions model particles are irrelevant for the task of qualitative prediction of behaviour. In addition, we combine irrelevance reasoning with reasoning about definitely relevant facts.

The concept of (ir)relevance has been applied in qualitative reasoning research before, mainly in the compositional modelling approach [5]. In this approach relevance heuristics in the form of different types of assumptions are used in selecting the right set of applicable model fragments. In a similar vein Nayak et al. [8] describe a procedure for generating a minimal device model. They use context dependent behaviours to describe different ways of component functioning. Structural and behavioural context as well as expected behaviour (the function of the device) are subsequently employed to select the applicable model fragments. Then this model is refined by applying simplification operators that substitute model fragments or remove them altogether.

The paper by Levy et al. [7] is another example: they describe heuristics for model fragment selection that are explicitly stated as irrelevance claims.

Compared with these approaches to (ir)relevance reasoning in building qualitative models we take the construction of an adequate model one step further since we do not focus on the (ir)relevance of entire model fragments but aim at removing irrelevant particles from applicable model fragments. We showed in which situations irrelevant model particles can still occur in applicable model fragments and described how these particles can be identified. The applicable model fragments, the generated behaviour description and the scenario are used to modify the model fragments. This approach allows us to refrain from defining large libraries of minimally differing model fragments. Therefore, we believe that our approach constitutes an important complement to the problem of constructing adequate qualitative models.

Further work needs to be done in the identification of parsimonious parameter relations in a model. Here the problem of interacting effects of different relations complicates matters substantially. Currently we continue to formulate guidelines for determining when model fragments can be subsumed and we are in the process of implementing the technique. Finally, we are defining general procedures for updating model fragments.

References

[1] B. Bredeweg. *Expertise in qualitative prediction of behaviour.* PhD thesis, University of Amsterdam, Amsterdam, March 1992.

[2] B. Bredeweg and Cis Schut. Reducing ambiguity by learning assembly behaviour. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, August 1993. (Forthcoming).

[3] J. de Kleer and J. S. Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24:7–83, 1984.

[4] J. deKleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

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

[6] K. D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[7] A. Y. Levy, Y. Iwasaki, and H. Motoda. Relevance reasoning to guide compositional modelling. In *Proceedings of the 6^{th} International Workshop on Qualitative Reasoning about Physical Systems*, pages 7–21, Edinburgh, Scotland, August 1992. Heriot-Watt University.

[8] P. P. Nayak, L. Joskowicz, and S. Addanki. Automated model selection using context-dependent behaviors. In *Proceedings of the 10^{th} AAAI*, pages 710–716, San Jose, California, July 1992.

[9] D. Subramanian and M. R. Genesereth. The relevance of irrelevance. In *Proc. of the 10^{th} IJCAI*, pages 416–422, Milan, Italy, 1987.

[10] D. S. Weld and J. de Kleer. *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann, 1990.

[11] B. P. Zeigler. *Theory of modelling and simulation.* Wiley-Interscience, New York, London, 1976.