Reducing Ambiguity by Learning Assembly Specific Behaviour*

Bert Bredeweg and Cis Schut
Department of Social Science Informatics (S.W.I.)
University of Amsterdam, Roetersstraat 15
1018 WB Amsterdam, The Netherlands
email: bert@swi.psy.uva.nl

Abstract

In this paper we present a technique for automatically generating constraints on parameter derivatives that reduce ambiguity in the behaviour prediction. Starting with a behaviour prediction using an initial library containing general domain knowledge the technique employs feedback about valid and spurious states of behaviour and knowledge about the causal dependencies between the parameters in the model in order to determine the constraints that remove the undesired states of behaviour that result from spurious ambiguity. In addition, the technique points out the assembly of physical objects to which the generated constraints apply.

1 Introduction

A recurring issue in qualitative prediction of behaviour is the construction of non-ambiguous models that only predict valid states of behaviour. In particular, when using a library of model fragments that represent general domain knowledge (such as processes [Forbus, 1984] and device behaviours [de Kleer and Brown, 1984]) the ambiguity introduced by the qualitative calculus, together with the requirement of modelling device behaviour independent from the context in which it operates (the 'no function in structure' principle [de Kleer and Brown, 1984]), makes it difficult to define adequate prediction models for a specific system. In order to remove undesired ambiguity additional constraints must be specified that represent assembly specific behaviour, such as (1) order of magnitudes [Raiman, 1986], and (2) conservation of quantities for the system as a whole. In this paper we present a technique that automatically derives these constraints by analysing valid and spurious states of behaviour and by using a model of the underlying causality. In addition, the technique identifies the physical structure, with its specific mode of behaviour, to which the constraints apply.

2 Automatic Model Construction

Our approach can be thought of as supporting a knowledge engineer who, on the basis of a library containing general domain knowledge, has to develop a specific model that can be used for a behaviour prediction task. Given such a library the knowledge engineer is confronted with two problems: relating the elements from the real-world system that has to be modelled to the canonical entities present in the library, and modelling additional constraints to reduce the ambiguity in the behaviour prediction. In other words the construction of a qualitative model amounts to finding the applicable model fragments and augmenting them with additional constraints to capture the device specific overall behaviour. Typically, the knowledge engineer goes through a debugging/refinement process, depending upon the behaviour prediction that the qualitative prediction engine produces. Each predicted state reflects a valid or a spurious form of behaviour and provides feedback for how the models from the initial library must be modified.

The problem of modelling is complex and cannot be automated all at once. In this paper we concentrate on deriving additional constraints on the derivatives of parameters that are required to remove undesired states of behaviour that result from ambiguity. An assumption therefore is that the initial knowledge in the library is sufficient for predicting at least all possible behaviours, but that it is too general in the sense that it may also predict behaviours that in fact do not occur.

After the knowledge engineer has classified (c.q. modelled) some system from the real-world in terms of the canonical elements present in the initial library (input system), the prediction engine generates a graph of possible behaviours. Although in principle this graph may include *all* behaviours (valid and spurious) that can be derived on the basis of the general knowledge in the library, it is usually necessary to limit the number of states to a subset that can still be understood and used by the knowledge engineer (partial prediction). Next, for each state of behaviour the knowledge engineer determines whether it represents a valid or a spurious state of behaviour by comparing it with the actual behaviour of the system in the real-world. The sets of valid and spurious

"The research described in this paper was partly 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).

states of behaviour are input for the process of refining the knowledge in the library.

The technique presented in this paper automates a part of this refinement cycle by proposing new constraints that will exclude false predictions (without excluding the valid predictions). The technique diagnoses the set of valid and spurious behaviours and determines *which* constraints have to be added to *which* parts of the knowledge in the library in order to remove the undesired ambiguity.

3 Representational Context

The framework for qualitative prediction of behaviour that we use, is implemented as a domain independent qualitative reasoning shell, called GARP, which can be used by a knowledge engineer for developing prediction models [Bredeweg, 1992]. In contrast to using a pure component oriented approach [de Kleer and Brown, 1984] (modelling the physical world as components connected by conduits) or a pure process oriented approach [Forbus, 1984] (modelling the physical world as physical objects that interact via processes), we claim that it is essential to use both component and process oriented abstractions in a single prediction model.

Similar to these approaches, GARP uses the notion of model fragments for determining the behaviour of some real-world system. Model fragments can be either static, process or agent models. Static model fragments represent general properties of physical objects, process models represent changes that occur between physical objects based on inequalities between interacting parameters and agent models represent changes imposed on the system by external agents. 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: *System elements:* abstractions of entities in the physical world; *Parameters:* relevant quantities; *Parameter values:* the values (intervals and derivatives) of quantities; *Parameter relations:* dependencies between quantities; *Model fragments:* other model fragments that must be true. 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.

Important for the technique discussed in this paper are the different types of parameter relations within the framework. Proportionalities and influences are used in a similar way as defined by Forbus [Forbus, 1984]: for modelling causal dependencies. Additional (non-causal) constraints on derivatives are modelled through inequalities, such as: < , < , = , > , >• Inequalities can be defined for a single derivative or a sum of derivatives with respect to zero, or for sets of derivatives with respect to each other. Inequalities are used to further constrain the ambiguity introduced by causal relations or for modelling constraints on derivatives that lack a clear causal dependency. As argued in Bredeweg [Bredeweg, 1992] both the causal and non-causal dependencies are essential features of a qualitative model.

4 Causes of Ambiguity

This section describes how several types of ambiguities result from the use of influences and proportionalities. The ideas are illustrated with a model of a refrigerator. Figure 1 visualises the important physical objects of this system. In order to reason about the behaviour of these objects, model fragments are required for representing device behaviour and for representing processes between physical objects (see [Bredeweg, 1992] for more details). The throttle valve and the compressor can be modelled as 'agent models' influencing the *amount* of substance in the condensor *(Amount$_c$on)* and in the evaporator *(AmountEvap)-* The throttle valve also influences the *pressure* of the substance in the evaporator *(PressEvap)* directly (expansion of fluid into gas leads to higher pressure). The substances in the condensor and the evaporator are both modelled as a 'closed contained substance' (cf. [Forbus, 1984]). In this model fragment temperature is proportional to pressure, which in turn is proportional to amount and to heat. The world and the cooling area can be modelled in this way too, omitting amount and pressure. A direct proportional relation between temperature and heat is sufficient for modelling the behaviour of these entities. Finally, model fragments are needed that represent processes such as heat-flow, evaporation and condensation. The causal relations (proportionalities and influences) that result from using these model fragments are depicted in figure 2.

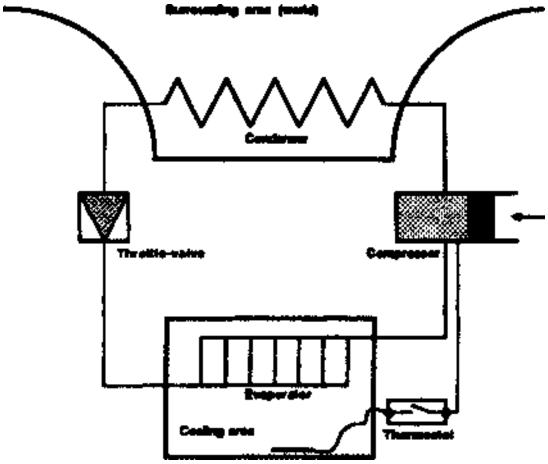

Figure 1: A model of the refrigerator

Presenting the model as sketched above to the prediction engine (GARP) results in 179 derived states of behaviour (total envisionment). The first specification step (finding all sets of model fragments that apply to the input system) produces already 15 states, 14 of them being spurious. The values of the parameter derivatives in these states are shown in table 1.

Spurious ambiguities are behaviours predicted by a qualitative simulator that do not represent actual behaviour of a device. They occur because particular globed properties of device behaviour are not modelled by individual model fragments and hence do not appear in the aggregate of model fragments that forms the model of

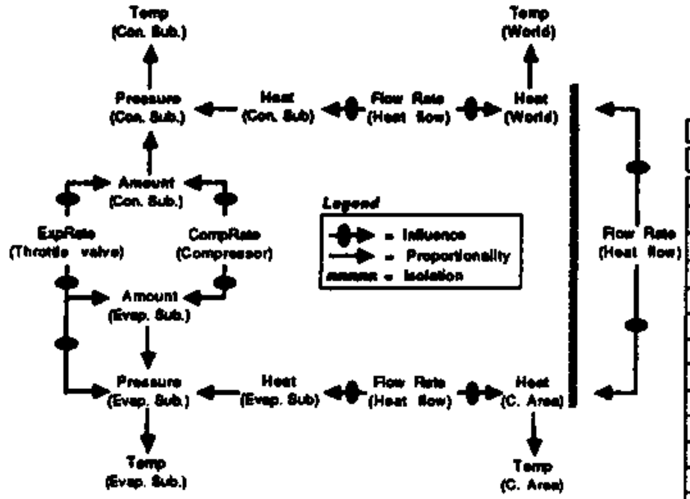

Figure 2: Causal dependencies in the refrigerator

the device. There are two types of these context dependent properties. The first one is the relative magnitude of multiple causal interrelationships originating from different model fragments. In the refrigerator example, the Amountcon is positively influenced by the compressor while the throttle valve influences it negatively. Since there is no knowledge of the relative magnitudes of these influences, the resulting derivative of the amount is ambiguous, i.e. it can either increase (the effect of the compressor being greater), stay constant (the effect of compressor and throttle valve being equal) or decrease (the effect of the throttle valve being greater). This ambiguity is spurious in the sense that it allows the amount, and thus the pressure and temperature, to decrease which allows a heat flow from the world to the condensor. This does not represent actual behaviour of a refrigerator. If the compressor is 'working' the Amountcon might increase or stay constant but never decrease (the reverse goes for the AmountE$_{vap}$). Another spurious ambiguity exists with respect to the Press con and Press Evap. The pressure is proportional to the amount but is also influenced by a heat flow process. If these effects are opposite the derivative of the pressure is ambiguous too.

The second cause for spurious ambiguity concerns missing constraints for conservation of quantities in the system as a whole. In the case of the refrigerator the total amount of substance in the system remains constant. However, if this constraint is missing a qualitative simulator may predict an increase of the Amountcon while the Amount$_{Evap}$ increases too, which is impossible. Constraints for conservation of quantities are essential for disambiguating these faults in the prediction model.

5 Determining Candidate Constraints

Both valid and spurious states of behaviour provide information for generating inequality constraints that remove spurious ambiguity. However, using information from spurious states of behaviour is problematic, because these behaviours may result from multiple, possibly independent, causes. There is, for example, no way to decide whether the first spurious state of behaviour (see table 1) is caused by lack of knowledge about order of magnitude (Amountcon always increases or stays constant, but never decreases) or by a lack of knowledge about conservation of quantities (changes in Amountcon and AmountEvap should equal zero).

Parameter	State														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$Press_{Evap}$	-	-	-	0	+	-	-	-	0	+	-	-	-	0	+
$Amou_{Evap}$	-	0	+	+	+	-	0	+	+	+	-	0	+	+	+
$Heat_{Evap}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$Temp_{Evap}$	-	-	-	0	+	-	-	-	0	+	-	-	-	0	+
$Press_{Con}$	-	-	-	-	-	0	0	0	0	0	+	+	+	+	+
$Amou_{Con}$	-	-	-	-	-	0	0	0	0	0	+	+	+	+	+
$Heat_{Con}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$Temp_{Con}$	-	-	-	-	-	0	0	0	0	0	+	+	+	+	+
$Heat_{Carea}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$Temp_{Carea}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$Heat_{World}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$Temp_{World}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$Erate_{Valve}$	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
$Crate_{Comp}$	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Table 1: First 15 states for the refrigerator (11 is valid)

The process is even more complicated when the prediction is partial, which is usually the case (see section 2). Although the reliability of the information stemming from the spurious states becomes greater when more states of behaviour have been predicted, it is likely that not all faults manifest themselves in a certain partial behaviour prediction. In other words, constraints which first seemed to represent a discriminating factor between valid and spurious states may turn out to be inappropriate. That is, there is danger of selecting a constraint that rules out valid states of behaviour that have not been predicted yet. Specifying, for example, that oTempcon > oTempEvap erroneously excludes future states of behaviour in which the two temperatures are equal.

In order to cope with the two problems described above our technique determines constraints for removing spurious ambiguity on information present in valid states of behaviour. For excluding spurious states of behaviour the technique searches within the set of all possible constraints consistent with the known valid states of behaviour. In other words, candidates for removing states that result from spurious ambiguity are based on one or more derivatives in the valid states of behaviour. Examples of such candidates are oPressEvap < 0, oPressEvap < 0 and oAmountcon + oHeatcon > 0. It is important to realise that this set of constraints represents all possible constraints that are consistent with the valid states of behaviour and that therefore the constraints for removing the spurious states of behaviour must necessarily be part of this list.

Although the problems associated with using spurious states of behaviour have been circumvented by this approach, three new problems appear. Firstly, as the number of parameters increases the set of possible constraints soon becomes intractable. Therefore, additional knowledge must be applied to significantly reduce the number of constraints. Secondly, there is a danger of generating constraints that relate parameters which are in fact independent from each other (for example: $Heat_{world}$ + $Heat_{Evap}$ = 0). Thirdly, not all possible constraints actually discriminate between valid and spurious states of behaviour. These issues are discussed in the next section.

6 Discriminating Candidate Constraints

This section describes how the set of candidate constraints can be reduced significantly by focusing on specific types of constraints, by exploiting the notion of causal units and by taking into account the discriminative power of the constraints.

6.1 Three Types of Constraints

Not every combination of parameter derivatives results in a constraint that represents valid knowledge about physics. As discussed above, the ambiguity that must be dealt with is caused by lack of knowledge about orders of magnitude and/or conservation of quantities. This knowledge can be represented by a limited number of constraints. Orders of magnitude between different influences or proportionalities can be represented in two forms. Firstly, the *net-effect* of the interacting parameter relations can be modelled as a specific value of the derivative of the determined parameter. For instance, 6 $Amount_{con}$ > 0 is a constraint, specifying that the derivative of the amount in the condensor should always be greater or equal than zero. Secondly, the *dominance* of one parameter relation over others can be modelled as a ratio of a pair of parameter derivatives. If, for example, the pressure is proportional to both heat and amount this results in ambiguity. Now, if, in the actual device behaviour, changes in the pressure are dominated by the amount, this can be modelled as o*Presscon* = o*Amountcon*- Various degrees of dominance can be represented as different inequalities between derivatives. Conservations of quantities are modelled as the sum of quantities being equal to zero (see next subsection).

In order to determine candidate orders of magnitude, constraints are generated on the basis of the parameter derivatives in the valid states of behaviour. For each individual parameter (net-effect) and for each pair of parameters fdominance") all constraints are generated that **are consistent with the derivatives in all valid states of behaviour. If, for example, there are two valid states of behaviour and parameter X has derivative + in one of them and derivative 0 in the other, then the constraints $\delta X > 0$ & $\delta X \geq 0$ and the constraints $\delta X \leq 0$ & $\delta X = 0$ & $\delta X \geq 0$ are consistent with these values respectively. The constraint $\delta X \geq 0$ is consistent with both derivatives and is therefore a candidate for disambiguating the behaviour prediction. In the case of pairs the procedure is similar. If a pair of parameters (X, Y) has the derivatives $(+,+)$ in one of the valid states and the derivatives $(-,0)$ in the other, then the constraints $\delta X \leq \delta Y$ & $\delta X = \delta Y$ & $\delta X \geq \delta Y$ and the constraints $\delta X \leq \delta Y$ & $\delta X < \delta Y$ are consistent with these values respectively.** The constraint $\delta X \leq \delta Y$ is consistent with both pairs of derivatives and is therefore a candidate for disambiguating the behaviour prediction.

6.2 Exploiting Causal Units

The causality in the behaviour description can be used to further reduce the number of candidate constraints. The idea is that dependencies between parameters can be factored into clusters that influence each other, but are independent of other parameters. This corresponds to the idea of 'factoring' (see [Genesereth and Nilsson, 1987]). We shall refer to these clusters as *causal units.* A causal unit starts with a parameter that is being influenced (by an influence relation), traverses via the proportionally related parameters, and ends with a parameter that has no causal effect on any other parameter by means of a proportionality. A causal unit is essentially a graph that may have recursive loops, more than one starting point and more than one terminal node (notice that a causal unit can consist of one or more causal paths). Two examples of causal units in our refrigerator model are: *Heatcon* → $Press_{con}$ → *Tempcon* (see also figure 2) and $Amount_{con}$ → $Press_{con}$ → $Temp_{con}$

Instead of generating all possible constraints that represent orders of magnitude, the notion of causal units can be used to focus on those parameters that are actually capable of causing ambiguity. Constraints on single parameter derivatives are generated for (1) parameters that are start points of causal units if they are multiply influenced and for (2) parameters that occur in more than one causal unit and are dependent on different parameters in the respective causal units. In these cases they might be causing spurious ambiguity. In the refrigerator example, instances of the above cases are $Amount_{con}$ and $Press_{con}$- Constraints for pairs of derivatives are also generated for parameters that occur in more than one causal unit *and* are dependent on different parameters in the respective causal units. These parameters are multiply dependent on *different* parameters. Therefore they are parameters that could cause spurious ambiguity. Each causing parameter, together with the dependent parameter, forms a candidate pair for representing dominance. In the refrigerator *Amountcon* & $Press_{con}$ and *Heatcon* & *Presscon* are two examples of such pairs.

Constraints for conservation of quantities are generated for parameters that: (1) belong to different causal units, (2) model the same type of quantity, and (3) are influenced by the same influence. Usually an influence (c.q. a flow rate) consists of two parts, one that decreases a quantity and one that increases a quantity. Both are required before a conservation constraint may be defined. In the refrigerator example a conservation constraint is generated between the parameters *Amountcon* and $Amount_{Evap}$.

6.3 Two Heuristics

The set of candidate constraints representing orders of magnitude can be restricted by applying two heuristics. The first one concerns the position of the parameters in the causal units:

- Prefer constraints on parameters 'early' in the causal units.

The rationale behind this is that the disambiguation should start at the beginning of the causal dependencies, i.e. to remove ambiguity at the source. For example, a constraint on *Amount$_{con}$* should be preferred over a constraint on *Press$_{con}$*- The second heuristic concerns the restrictiveness of the constraints. If a parameter (or a pair of parameters) has both the stronger and the weaker constraint then:

- Prefer weaker constraints over stronger constraints.

Weaker constraints allow more values than stronger constraints. For example, " > " is a weaker constraint than either ">" or "=". This rule prohibits that on the basis of incomplete information a too restrictive constraint is selected.

Both heuristics can be applied without any danger of losing important knowledge. If constraints on early parameters and/or weak constraints fail to exclude spurious states of behaviour, constraints on later parameters and/or stronger constraints will be found when the behaviour prediction is extended (see section 8).

6.4 Discriminative Power of Constraints

The third source of information for cutting down the set of candidate constraints concerns the number of spurious states of behaviour that each constraint eliminates.

Non-discriminative constraints can be disregarded because they do not exclude any spurious states of behaviour.

Independent constraints exclude states of behaviour that are not excluded by any other constraint. Therefore they are necessary for the disambiguation of the model and must be added to the library knowledge.

Partially overlapping constraints exclude some states of behaviour uniquely, that is, these states are not excluded by any other constraint. In addition they exclude some states of behaviour that are also excluded by other constraints. Because they exclude some states uniquely these constraints are also necessary for the disambiguation of the model and must be added to the library knowledge.

Overlapping constraints exclude only states that are also excluded by other constraints. A reasonable heuristic in this case seems to be to prefer the constraint that excludes most states of spurious behaviour.

If this heuristic concerning overlapping constraints does not resolve the problem then an additional heuristic is to prefer complex overlapping constraints over simple ones, i.e. conservation constraints over derivative pair constraints over single derivative constraints. However, in the case of a partial behaviour prediction the only safe way is to have the knowledge engineer decide upon overlapping constraints. In the case of a full behaviour prediction overlapping constraints introduce no problem because all valid states of behaviour are facilitated by the constraints and they can thus simply be added to the model. In the refrigerator example, the technique as described above comes up with the constraints as shown in table 2.

Type	Constraints	Eliminated states
C	$\delta Amou_{Con} + \delta Amou_{Evap} = 0$	1 2 6 8 9 10 12 13 14 15
D	$\delta Press_{Evap} \geq \delta Amou_{Evap}$	2 3 4 7 8 9 12 13 14
N-E	$\delta Amou_{Con} \geq 0$	1 2 3 4 5
	$\delta Amou_{Evap} \leq 0$	3 4 5 8 9 10 13 14 15

Table 2: Remaining constraints with elimited states

7 Allocating Constraints to Fragments

Once the appropriate constraints have been determined, they should be allocated to model fragments in the library. Two options exist: they are added to an already existing model fragment, or they are added to a newly defined model fragment. The method for allocating them is similar in both cases.

The first step consists of determining all dependencies that the parameters in the proposed constraint have. Each of these relations contributes to the ambiguity that will be removed by the constraint. For example, a constraint between *Amount$_{con}$* and *Press$_{con}$* resolves spurious ambiguity that results from (1) the influence of the compression rate, (2) the influence of the expansion rate, (3) the proportional relation with the heat and consequently (4) the influence of the flow rate (from the Heat flow), (see also figure 2). Again the notion of causal units is important, in the sense that the set of contributing relations is a subset of the causal units to which the parameters belong. In particular, this subset starts with the influencing relations, moves on via proportionalities up to the place where the parameters themselves are located. The relations located higher in the causal unit do not contribute to the ambiguity.

The second step is to find the set of model fragments that introduced the relations that cause the ambiguity. They are conditional for the model fragment to which the new constraint should be added. In the example mentioned above these model fragments are: (1) the heat flow process, (2) the active compressor, (3) the active throttle valve, and (4) the closed contained substance (=the condensor). For allocating the new constraint either a model fragment exists for which these model fragments are conditional or a new aggregate model fragment has to be created. In the latter case, which is more likely, the model fragments that introduce the ambiguous dependencies are conditional to the new model fragment. In the above example this new model fragment could be referred to as the 'condensing assembly' of the refrigerator.

8 Extending the Behaviour Prediction

Further behaviour prediction can be useful or even necessary in some cases. Instead of immediately adding all proposed constraints to the library, the behaviour prediction can be enlarged to create more discriminative power for grounding the constraints. In our example we might have added one transformation step creating 38 states of behaviour.

However, even if candidate constraints have been added to the library it may still be necessary to further specify the knowledge in the library. There are in fact three reasons why this may be the case. Firstly, weaker constraints may have to be replaced by stronger ones and/or constraints on early parameters by constraints on later parameters. Secondly, parameters appearing to be correct, may start behaving in ways that are incorrect and/or undesired. Finding these parameters requires further specification. Finally, it may be the case that some model fragment introduces parameters that were not present in the behaviour prediction yet. Spurious ambiguity introduced by their dependencies should also be removed.

9 Conclusions

In this paper we have presented a technique for automated generation of the constraints that are needed for disambiguation of a behaviour prediction. Based on general domain knowledge a behaviour prediction is generated that contains valid states as well as spurious states. The technique removes the spurious states using feedback about the validity of states and knowledge about the causal dependencies in the model. This is achieved by generating additional constraints on parameter derivatives and by adding these to the assembly of physical objects to which they apply.

In the machine learning field the work of Mozetic is most similar to our technique (e.g. [Mozetic, 1987]). By representing models in a logical language the refinement problem becomes similar to refinement of logic programs. To our knowledge there has been no previous work on refining knowledge that is represented as qualitative constraints. DeJong's work ([DeJong, 1989]) focuses on explanation-based learning in the context of plausible reasoning rather than refinement.

The refinement algorithm presented in this paper is similar to general incremental learning techniques. It is special in the representation and inference engine. An important addition is the learning bias that is introduced by factoring into causal units and the search for specific types of constraints.

In the literature on automated modelling two approaches can be distinguished. One approach derives a qualitative model of a system from data on its actual behaviour (cf. [Richards *et al.,* 1992; Coiera, 1989; Bratko *et ai,* 1992]). In contrast with these data driven techniques, the more commonly adopted approach, as introduced in [Forbus, 1984; de Kleer and Brown, 1984], entails the selection of the appropriate set of readily available model fragments for a particular system. Our technique uses both approaches: generally applicable model fragments are reusable and a model can be tuned to actual device specific behaviour.

The scope of our technique is still limited in that it only deals with derivatives. E.g., spurious behaviours indicated by incorrect parameter values cannot be dealt with (although it seems likely that parts of the technique can be used for this purpose as well). Also the technique requires that the knowledge present in the initial library is correct and sufficient, i.e. it should at least predict all valid states of behaviour.

However, these problems are not caused by limitations of the technique. Instead, they refer to different aspects of the modelling process that must be dealt with in order to further automate the process of qualitative model construction. The technique for generating the disambiguating constraints for parameter derivatives presents an important step in this direction.

Acknowledgement

We would like to thank Gautam Biswas, Ivan Bratko, Adam Farquhar, Ken Forbus, and Lyle Ungar for their insightful comments on an earlier version of this paper. Our thanks are also due to Kees van den Heerik and Maarten van Someren who contributed to the ideas presented in this paper.

References

[Bratko *et ai,* 1992] I. Bratko, S. Muggleton, and A. Varsek. Learning qualitative models of dynamic systems. In S. Muggleton, editor, *Inductive Logic Programming,* pages 437-452. Academic Press, London, 1992.

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

[Coiera, 1989] E. Coiera. Generating qualitative models from example behaviors. Technical Report DCS 8901, Department of Computer Science, University of New South Wales, May 1989.

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

[DeJong, 1989] G. DeJong. Explanation-based learning with plausible inferencing. In K. Morik, editor, *Proceedings European Working Session on Learning EWSL-89,* pages 1-10. Pitman, 1989.

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

[Genesereth and Nilsson, 1987] M. R. Genesereth and N. J. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann, Los Altos, California, 1987.

[Mozetic, 1987] I. Mozetic. The role of abstractions in learning qualitative models. In *Proceedings of the 4th International Workshop on Machine Learning,* pages 242-255, Irvine, 1987. University of California.

[Raiman, 1986] O. Raiman. Order of magnitude reasoning. In *Proceedings of the AAAI,* pages 100-104, San Mateo, California, 1986. Morgan Kaufmann.

[Richards *et a*/., 1992] B. L. Richards, I. Kraan, and B. J. Kuipers. Automatic abduction of qualitative models. In *Proc. of AAAI-92,* pages 723-728, San Jose, California, July 1992.