Task-dependent Qualitative Domain Abstraction ⋆

M. Sachenbacher [a,∗], P. Struss [b]

[a] *MIT CSAIL, 200 Technology Square, Cambridge, MA 02139, USA*

[b] *Technische Universität München, Fakultät für Informatik, Boltzmannstr. 3, 85748 Garching, Germany*

Abstract

Automated problem-solving for engineered devices is based on models that capture the essential aspects of the behavior. In this paper, we deal with the problem of automatically abstracting behavior models such that their level of granularity is as coarse as possible, but still sufficiently detailed to carry out a given behavioral prediction or diagnostic task. A task is described by a behavior model, as composed from a library, a specified granularity of the possible observations, and a specified granularity of the desired results. The goal of task-dependent qualitative domain abstraction is to determine maximal partitions for the variables' domains (termed qualitative values) that retain all the necessary distinctions. We present a formalization of this problem within a relational (constraint-based) framework, and devise solutions to automatically determine qualitative values for a device model. The results enhance the ability to use a behavior model of a device as a common basis to support different tasks along its life cycle.

Key words: Model-based systems, Qualitative reasoning, Domain abstraction

1 Introduction

Model-based systems [11, 25] represent knowledge about the structure and behavior of a physical system in terms of a behavior model, and use it to

⋆ Part of this work appeared in preliminary form in the Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03).

∗

Email addresses: sachenba@mit.edu (M. Sachenbacher), struss@in.tum.de (P. Struss).

support engineering tasks such as behavior prediction, diagnosis, planning and testing. In recent years, model-based systems have increasingly been applied in on-board contexts, as part of embedded systems, and in real-time applications such as monitoring and control of space systems or passenger vehicles [15, 3, 20].

When constructing model-based systems, one of the most difficult parts is modeling the device. A fundamental idea to support and facilitate modeling is to compose models from model fragments, that is, re-usable elements of knowledge about a device that can be organized in a library [6]. This requires that model fragments have to be formulated, as far as possible, in a generic way and independent of their specific application context. However, it also means that information about the task a model will be used for cannot be anticipated in the model fragments.

But a model needs to be suited for the problem-solving task at hand in order to provide an effective and efficient solution to it. Using always the most accurate and most detailed model available can render the respective problem-solving task intractable, or at least unnecessarily complex and resource-consuming. For instance, for the task of diagnosing a device in an on-board environment, it is crucial to have a model that focuses only on those aspects that are essential to the goal of discriminating between its normal and faulty behavior. Any unnecessary details that are not relevant to this task impair its ability to meet the stringent time and space requirements of this application. In general, models straightforwardly composed from a library tend to be either inefficient, because they are overly detailed (that is, too fine-grained), or ineffective, because they are not detailed enough (that is, too coarse-grained) for the task they will be used for.

The approach pursued in this paper is therefore to automatically re-formulate a behavior model, after it has been composed, to a level of abstraction that is adequate for the specified task. We focus on the abstraction of the domains of variables, that is, the problem of deriving meaningful qualitative values. Much of the work in qualitative reasoning about physical systems [25, 7] relies on this type of abstraction. The resolution of a behavior model's domains has a strong effect on the size of the model, the efficiency of reasoning with the model, and the size of the solutions. Within an on-board or real-time setting, the number of qualitative values determines how many of the observations will be qualitatively different, and therefore it influences the frequency at which reasoning has to be initiated at all. But often, qualitative values are defined only ad hoc, for example, by introducing values such as "high", "medium", or "low". Although work has been carried out on finding qualitative values within specific contexts, such as simulation [12, 13], the general problem of characterizing and systematically deriving qualitative values for an arbitrary relational (constraint-based) behavior model is a relatively unexplored area.

Consider, for example, the system depicted in Figure 1. The device is a simplified version of a pedal position sensor used in a passenger car. Its purpose is to deliver information about the position of the accelerator pedal to the electronic control unit (ECU) of the engine management system. The ECU uses this information to calculate the amount of fuel that will be delivered to the car engine.

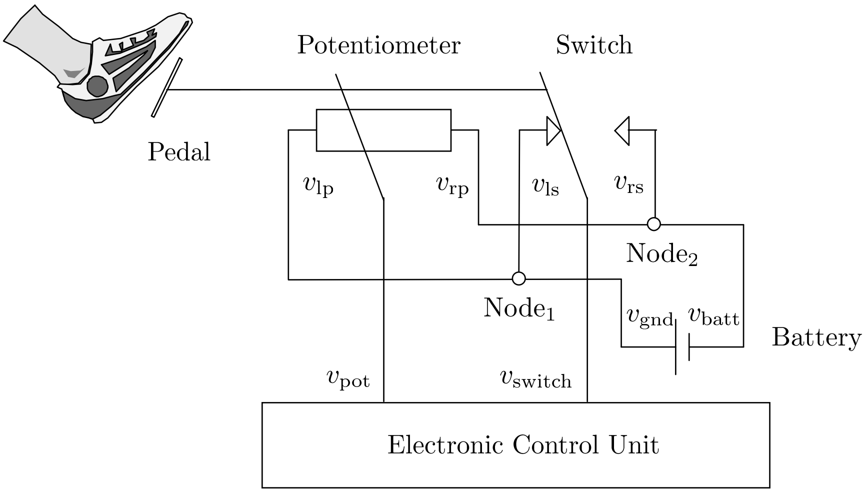

Fig. 1. The Pedal Position Sensor

The pedal position is sensed in two ways, via the potentiometer as an analog signal, v_{pot}, and via the idle switch as a binary signal, v_{switch}. The idle switch changes its state at a particular value $p_{\mathrm{switching}}$ of the mechanically transferred pedal position. The reason for the redundant sensing of the pedal position is that the signals v_{pot} and v_{switch} are cross-checked against each other by the onboard control software of the ECU. This plausibility check is a safety feature of the system, in order to avoid cases where a wrong amount of fuel injected evokes dangerous driving situations.

Assume we want to perform the plausibility check between the electrical signals v_{pot} and v_{switch} automatically by the means of a behavior model of the system. For the potentiometer model fragment, this requires a distinction in the domain of v_{pot} that corresponds to the switching point $p_{\mathrm{switching}}$ of the switch. This is the only distinction in this domain that is required for the purpose at hand.

The problem is that this particular distinction cannot be anticipated in a generic model fragment of the potentiometer component, because it would not make any sense in a different structure. It is only the specific combination of the potentiometer and the switch together with the pursued task that requires this distinction. In contrast, other tasks such as control or design might require more detailed domains that would allow to relate the position of the switch to particular potentiometer voltages.

The problem is important, because it impairs the idea of using a model of the pedal position sensor as a common basis for different tasks. For engineered systems, it is typical that several tasks along the product's life cycle — such as failure effects analysis, on-board diagnostics development, generation of repair manuals or workshop diagnosis — share a significant amount of common knowledge about the behavior of the system under consideration. It would be unacceptable having to *manually* create models from scratch that are tailored to each of these tasks.

1.2 Towards Qualitative Domain Abstraction from First Principles

The example above has confronted us with the problem that simply picking model fragments from a library and composing the model is not enough. It is infeasible, in general, to anticipate the required granularity in the domains of variables. Therefore, the ability to *transform* the domains to the right level of abstraction *after* composing the model's constraints is a highly practical requirement. It means grouping together domain values whose distinction is irrelevant for the task at hand.

The core idea of distinctions between domain values being redundant is captured by the concept of interchangeability, first proposed by Freuder [8]. For a constraint satisfaction problem (CSP) that consists of a set of variables, domains and constraints on these variables, two values val_1, val_2 of a variable v are said to be fully interchangeable, if for any solution where $v = \mathrm{val}_1$, substituting $v = \mathrm{val}_2$ produces another solution, and vice versa. That is, solutions involving val_1 (val_2) are identical to solutions involving val_2 (val_1) except for the values val_1, val_2 themselves. Interchangeable values define equivalence classes on the domains of the variables, and grouping them together corresponds to an abstraction of the CSP that exactly preserves its set of solutions. Freuder and Sabin [8], [9] already observe that interchangeability is related to abstraction and the formation of "semantic groupings" within the domains of variables. However, it is also known that in practice, interchangeability in CSPs does not occur very frequently.

The idea pursued in this paper to leverage the specific context of a model-based problem solving task to identify redundant distinctions. A model-based problem solving task, such as behavioral prediction or diagnosis, is specific in two respects:

(1) the input consists not only of the model, but also of the observations that it is confronted with, such as measurements, hypothetical situations, etc.

Typically, observations are restricted because not all of the variables in the model are observable (like in the example in Section 1.1), or because values cannot be observed beyond a certain granularity.

(2) the output involves not all the feasible assignments of values to the variables, but instead only certain aspects of the solutions are required. Typically, we want to know whether values remain below or exceed a certain threshold (as for the example in Section 1.1), or it is sufficient to determine values for a subset of the variables, such as mode variables, etc.

This context of a model-based task will be captured as (1) *observable distinctions* that express what inputs to the problem solving process (for example, observations) can occur, and (2) *target distinctions* that express what aspects of the outcome we are after. Observable and target distinctions can be exploited to obtain so-called *induced abstractions* — domain abstractions that often go beyond the level of interchangeability, but are still adequate for the given model-based task. We pursue the approach in the context of general, relational models that are not limited to restricted cases such as linear relationships or monotonic functions.

The paper is organized as follows. Section 2 introduces relational behavior models, task-dependent distinctions and domain abstractions as fundamental concepts of our approach. Section 3 formally defines task-dependent qualitative domain abstraction as the problem of obtaining distinctions that are adequate for a certain task but as coarse as possible, and it characterizes solutions to this problem. Section 5 sketches how domain abstractions are computed and exploited within a model-based reasoning framework. Section 6 describes the application of reformulation based on task-dependent domain abstraction in the context of a real-world example taken from the automotive domain. In section 7, we discuss related work and identify directions for future research.

2 Model-based Problem Solving

A behavior model is a relation (constraint) defined over a set of variables $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ with domains $\mathrm{dom}(v_i)$, $i = 1, 2, \ldots, n$:

$$R \subseteq \mathrm{dom}(\boldsymbol{v}) = \mathrm{dom}(v_1) \times \mathrm{dom}(v_2) \times \ldots \times \mathrm{dom}(v_n).$$

A domain can have an infinite or a finite number of elements, such as the "left" and "right" states of the idle switch in the example considered in Section 1.1. The relation is not limited to a class of algebraic operations or monotonic functions, but can be any subset of the cross-product of the domains. Tables 1

Table 1
Variables and domains for an instance of the pedal position sensor model in Section 1.1. The domain for variables involving voltage has five values. The domain for variables involving position has six values (0% means that the gas pedal is in rest position, and 100% means that the pedal is fully pushed through)

Variable	Domain
s_{switch}	{left, right}
p, $p_{\text{switching}}$	{0%, 20%, 40%, 60%, 80%, 100%}
v_{pot}	{[0V,2V), [2V,4V), [4V,6V), [6V,8V), [8V,10V)}
v_{ls}, v_{rs}, v_{switch}, v_{lp}, v_{rp}, v_{batt}, v_{gnd}	{[0V,2V), [8V,10V)}

Table 2
Relation for an instance of the pedal position sensor model in Section 1.1. It is assumed that the only parameter in the system, $p_{\text{switching}}$, equals 40%. The relation R consists of ten tuples (some of the variables have been omitted)

v_{pot}	p	v_{switch}	s_{switch}	v_{batt}	v_{gnd}	...
[0V,2V)	0%	[0V,2V)	left	[8V,10V)	[0V,2V)	...
[0V,2V)	20%	[0V,2V)	left	[8V,10V)	[0V,2V)	...
[2V,4V)	20%	[0V,2V)	left	[8V,10V)	[0V,2V)	...
[2V,4V)	40%	[0V,2V)	left	[8V,10V)	[0V,2V)	...
[4V,6V)	40%	[0V,2V)	left	[8V,10V)	[0V,2V)	...
[4V,6V)	60%	[8V,10V)	right	[8V,10V)	[0V,2V)	...
[6V,8V)	60%	[8V,10V)	right	[8V,10V)	[0V,2V)	...
[6V,8V)	80%	[8V,10V)	right	[8V,10V)	[0V,2V)	...
[8V,10V)	80%	[8V,10V)	right	[8V,10V)	[0V,2V)	...
[8V,10V)	100%	[8V,10V)	right	[8V,10V)	[0V,2V)	...

and 2 show the variables, domains and the relation for an instance of the pedal position sensor example in Section 1.1.

We apply the operations $\bowtie$ (join), Π (projection) and σ (selection) on relations. Using the model for problem-solving means that external restrictions

$$R_{\text{obs}} \subseteq \text{dom}(\boldsymbol{v})$$

such as observations, further restrict the set of possible states (left part of Figure 2). Given a model R and an external restriction R_{obs}, the basic task of model-based reasoning is then to determine the remaining states (right part of Figure 2):

$$R_{\mathrm{sol}} = R \bowtie R_{\mathrm{obs}}.$$

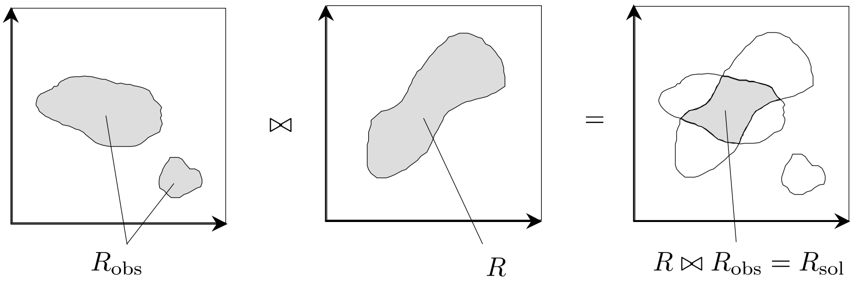

Fig. 2. Relational framework for model-based problem solving. A model relation R, confronted with external restrictions R_{obs}, yields solutions R_{sol}.

For instance, for the model relation in Table 2, observing v_{pot} = [2V,4V) implies that s_{switch} = left and v_{switch} = [0V,2V).

2.1 Task-dependent Distinctions

We now augment the model-based problem solving framework presented above by a means to represent task characteristics. A task is characterized by the granularity of the inputs R_{obs} that can occur or have to be considered, and by the granularity of the outputs R_{sol} that is interesting or useful when solving problems with the model. The former is captured by so-called observable distinctions, while the latter is captured by so-called target distinctions. Both are defined in terms of domain granularity and both influence the appropriate granularity of the behavior model R.

Observable distinctions are a means to express measurement granularity or incomplete observability of variables (observations are only a special case of external restrictions, which could also correspond to specifications given by the user, or hypothetical situations). Observable distinctions identify states that *cannot* be distinguished from each other; they give rise to abstractions of the model because they introduce a "don't know" indeterminism among its states. An observable distinction for a variable is expressed as a partition of its domain:

Definition 1 (Observable Distinction) *An observable distinction for a variable v_i, denoted $\pi_{\mathrm{obs},i}$, is a partition of its domain* $\mathrm{dom}(v_i)$.

A variable v_i is not observable at all if $\pi_{obs,i}$ is equal to the trivial domain partition $\pi_{\mathrm{triv},i} := \{\mathrm{dom}(v_i)\}$. For instance, in on-board diagnosis, only certain variables corresponding to the sensor inputs might be observable.

Example 2 (Observable Distinction for Pedal Position Sensor) *For the pedal position sensor presented in Section 1.1, the electronic control unit senses the output voltages of the potentiometer and the switch component, but cannot measure the other variables. This can be stated as*

$$\begin{aligned}\pi_{\mathrm{obs},v_{\mathrm{pot}}} &= \{\{[0V,2V)\},\{[2V,4V)\},\ldots,\{[8V,10V)\}\},\\ \pi_{\mathrm{obs},v_{\mathrm{switch}}} &= \{\{[0V,2V)\},\{[8V,10V)\}\}.\end{aligned}$$

The other variables receive the trivial partition

$$\pi_{\mathrm{obs},i} = \{\mathrm{dom}(v_i)\}.$$

Target distinctions reflect the granularity of solutions we are after, identifying states that *need not* be distinguished from each other. Target distinctions give rise to abstractions of a model because they introduce a "don't care" indeterminism. Analogously to observable distinctions, target distinctions are expressed as domain partitions:

Definition 3 (Target Distinction) *A target distinction for a variable v_i, denoted $\pi_{\mathrm{targ},i}$, is a partition of its domain* $\mathrm{dom}(v_i)$.

A variable v_i is said to have no target partition, if $\pi_{\mathrm{targ},i}$ is equal to the trivial partition. For instance, we might be interested in the values of certain output variables only, such as the possible behavior modes of the components in the case of a diagnostic task. For on-board diagnosis, it might even not be necessary to know the particular behavior mode of the components, but instead it might be sufficient to distinguish only those classes of behavior modes that require different actions of the control unit.

Example 4 (Target Distinction for Pedal Position Sensor) *The goal to distinguish the ground voltage and battery voltage levels for the variable v_{switch} in the example in Section 1.1 can be expressed as a target partition that separates the two domain values $[0V,2V)$ and $[8V,10V)$:*

$$\pi_{\mathrm{targ},v_{\mathrm{switch}}} = \{\{[0V,2V)\},\{[8V,10V)\}\}.$$

The other variables receive the trivial partition

$$\pi_{\mathrm{targ},i} = \{\mathrm{dom}(v_i)\}.$$

A domain partition π_i can also be understood as a domain abstraction

$$\tau_i : \mathrm{dom}_1(v_i) \rightarrow \mathrm{dom}_2(v_i) \subseteq 2^{\mathrm{dom}(v_i)}$$

that maps values from a base domain $\mathrm{dom}_1(v_i)$ to a transformed domain $\mathrm{dom}_2(v_i)$ that consists of sets of values from the base domain, such that val $\in \tau_i(\mathrm{val})$. Abstractions can be extended straightforwardly from a single value to a set of values by taking the union of the resulting sets, $\tau_i(\mathrm{val_1}) \cup \ldots \cup \tau_i(\mathrm{val_k})$. Likewise, they can be extended from abstracting a single domain to abstracting a set of domains, $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_n)$. The application of an abstraction $\boldsymbol{\tau}$ to a relation R yields the transformed relation

$$\boldsymbol{\tau}(R) := \{(\tau_1(\mathrm{val}_1), \tau_2(\mathrm{val}_2), \ldots, \tau_n(\mathrm{val}_n)) \mid (\mathrm{val}_1, \mathrm{val}_2, \ldots, \mathrm{val}_n) \in R\}.$$

Figure 3 illustrates the domain abstractions $\boldsymbol{\tau}_{\mathrm{obs}}$ and $\boldsymbol{\tau}_{\mathrm{targ}}$ corresponding to observable and target distinctions, and their application to the relations R_{obs} and R_{sol}, respectively.

The *merge* of two domain abstractions $\tau_{i,1}$ and $\tau_{i,2}$ is the domain abstraction $\tau_{i,3}$ such that $\tau_{i,3}(\mathrm{val}) = \tau_{i,1}(\mathrm{val}) \cap \tau_{i,2}(\mathrm{val})$. A domain abstraction $\tau_{i,1}$ is a *refinement* of $\tau_{i,2}$, if for all val $\in \mathrm{dom}(v_i)$, $\tau_{i,1}(\mathrm{val}) \subseteq \tau_{i,2}(\mathrm{val})$. We apply the notion of refinement and merge equally to mappings and domains. An abstraction $\boldsymbol{\tau}_1$ is a refinement of $\boldsymbol{\tau}_2$, if every $\tau_{i,1}$ is a refinement of $\tau_{i,2}$. Two abstractions $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are *piecewise comparable*, if for each $\tau_{i,1}$ and $\tau_{i,2}$, either $\tau_{i,1}$ is a refinement of $\tau_{i,2}$, or $\tau_{i,2}$ is a refinement of $\tau_{i,1}$.

We extend the definition of the join operation to combine relations abstracted by mappings $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$. The result is only defined if $\boldsymbol{\tau}_1$ is a refinement of $\boldsymbol{\tau}_2$ or $\boldsymbol{\tau}_2$ is a refinement of $\boldsymbol{\tau}_1$. The result is a relation on the level of abstraction of $\boldsymbol{\tau}_1$ in the former case, and a relation on the level of abstraction of $\boldsymbol{\tau}_2$ in the latter case.

3 Qualitative Abstraction Problems

Given this representational apparatus, we can now formally define the problem of task-dependent qualitative abstraction.

Definition 5 (Qualitative Abstraction Problem) *Let R be a relational behavior model,* Obs *a set of external restrictions,* $\boldsymbol{\tau}_{\mathrm{obs}}$ *a domain abstraction*

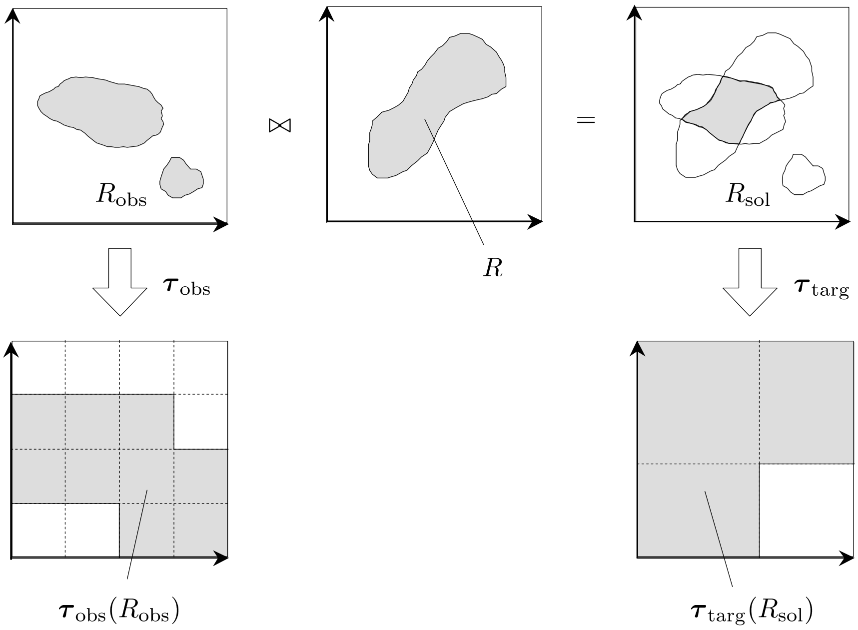

Fig. 3. Framework for model-based problem solving enhanced by task-dependent distinctions. Observable distinctions ($\boldsymbol{\tau}_{\text{obs}}$, indicated by grid on left-hand side) define the granularity of external restrictions (in this case, only four values can be distinguished for each variable). Target distinctions ($\boldsymbol{\tau}_{\text{targ}}$, indicated by grid on right-hand side) define the granularity of solutions (in this case, only two values need to be distinguished for each variable).

defined by observable distinctions, and $\boldsymbol{\tau}_{\text{targ}}$ *a domain abstraction defined by target distinctions. The qualitative abstraction problem consists of finding a so-called induced domain abstraction* $\boldsymbol{\tau}_{\text{ind}}$ *such that*

(1) (Adequacy) For all external restrictions $R_{\text{obs}} \in \text{Obs}$,

$$\boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})) = \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}_{\text{ind}}(\boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}}))).$$

(2) (Maximality) If $\boldsymbol{\tau}_{\text{ind}}$ *is a refinement of* $\boldsymbol{\tau}'_{\text{ind}}$ *and* $\boldsymbol{\tau}'_{\text{ind}}$ *fulfills (1), then* $\boldsymbol{\tau}'_{\text{ind}} = \boldsymbol{\tau}_{\text{ind}}$.

The first condition (adequacy) states that the abstracted model $\boldsymbol{\tau}_{\text{ind}}(R)$ derives a solution on the level of target distinctions, if and only if the original model R derives the same solution on the level of target distinctions. We require this to hold for a set of external restrictions (actual observations, design specifications, etc.) on the level of observable distinctions. This guarantees that for any such external restriction, the abstracted model will yield the same results as the original model. That is, if we apply $\boldsymbol{\tau}_{\text{ind}}$ *before* carrying out our problem-solving task, it won't affect the result because this abstraction incorporates all the distinctions that are necessary for this task. As a consequence, we can substitute the abstracted model $\boldsymbol{\tau}_{\text{ind}}(R)$ for the original model R in problem solving.

The view we take here is that a domain abstraction is considered adequate if it keeps enough distinctions in the domains $\mathrm{dom}(v_i)$ to preserve their original "distinguishing power" with respect to the solutions. Distinctions in the domain of a behavior model should be made only if they are really necessary to derive conclusions about the required solutions.

In general, there may be many adequate domain abstractions. In particular, the identical domain abstraction $\boldsymbol{\tau}_{\mathrm{id}}$ that retains all the distinctions in the domains is an adequate abstraction. However, among all adequate abstractions, we prefer those that are maximal according to the second condition of Definition 5. Maximal abstractions are coarsest in the sense that there exists no other adequate abstraction of which they are a strict refinement (an abstraction that would further aggregate at least two of the qualitative values).

Definition 5 formalizes the problem of finding qualitative values for the domains of variables: a domain abstraction that is both adequate and maximal neither makes any unnecessary distinctions, nor abstracts away any distinctions that are crucial to solve the problem.

A qualitative abstraction problem (QAP) describes a whole class of instances defined by a model relation and set of external restrictions. This is in contrast to interchangeability, which is concerned with possibilities for abstraction within a single problem instance only. Interchangeability requires that the solutions remains the same as for the original model. A QAP relaxes this basic principle and demands that the solutions remain the same only on the level of target distinctions, and only for inputs on the level of observable distinctions.

Example 6 (Multiplication Constraint) *Let* $\boldsymbol{v} = (v_1, v_2, v_3)$. *Let* $\mathrm{dom}(v_i)$ *be equal to the real numbers,* $i = 1, 2, 3$. *Let* R *express the behavior* $v_1 \cdot v_2 = v_3$, *and let* $\mathrm{Obs} = 2^{\mathrm{dom}(\boldsymbol{v})}$. *Assume that only* v_1 *and* v_2 *are observable, and that a target partition is given only for* v_3*:*

$$\pi_{\mathrm{targ},3} = \{\mathrm{val}_1, \mathrm{val}_2, \mathrm{val}_3, \mathrm{val}_4, \mathrm{val}_5\} = \{(-\infty, 0), 0, (0, 1), 1, (1, \infty)\}.$$

Assume, first, that the observable distinction for v_1 *and* v_2 *is the identical partition* π_{id}. *Then the induced abstractions for* v_1 *and* v_2 *are also equal to the identical abstraction:*

$$\pi_{\mathrm{ind},1} = \pi_{\mathrm{ind},2} = \pi_{\mathrm{id}}.$$

To see this, consider an abstraction $\boldsymbol{\tau}$ *that maps two different real numbers*

$a_1 \neq a_2$ onto the same value for v_1. Then choosing the external restriction

$$R_{\text{obs}} = \{(a_1, \frac{1}{a_1}, (-\infty, \infty))\}$$

reveals the loss: for the case $0 < a_1 < a_2$, the original relation R yields the solution val_4 for v_3, whereas the abstraction $\boldsymbol{\tau}(R)$ yields the solution $\text{val}_4 \cup \text{val}_5$ for v_3 (the other cases are similar). Now assume that the observable distinction for v_1 and v_2 is a partition that consists of the integer values and open intervals between them:

$$\pi_{\text{obs},1} = \pi_{\text{obs},2} = \{\dots, -1, (-1, 0), 0, (0, 1), 1, \dots\}$$

As suggested by Figure 4, in this case all values of v_1 greater than 1 can be summarized. It would not pay off to distinguish between them because the values of v_2 are not fine-grained enough to determine, for instance, whether v_3 is less than, equal to, or greater than 1. Therefore, the induced abstractions for v_1 and v_2 are given by

$$\pi_{\text{ind},1} = \pi_{\text{ind},2} = \{(-\infty, -1), -1, (-1, 0), 0, (0, 1), 1, (1, \infty)\}.$$

This example illustrates the influence of the granularity of the external restrictions on the level of abstraction that can be achieved.

3.1 Solutions to Qualitative Abstraction Problems

The definition of a qualitative abstraction problem is quite general. In particular, it includes situations where the external restrictions do not comprise all possible observations at the level of observable distinctions, and situations where it is impossible to discriminate solutions at the level of target distinctions. In the following, we restrict ourselves to qualitative abstraction problems where these cases do not occur.

Definition 7 (Obs-Completeness) *A QAP is obs-complete, if all observations at the level of observable distinctions can occur: $\boldsymbol{\tau}_{\text{obs}}(\text{Obs}) = 2^{\boldsymbol{\tau}_{\text{obs}}(\text{dom}(\boldsymbol{v}))}$.*

Obs-completeness means that all possible observations on the level of observable distinctions have to be considered. Consequently, induced abstractions can be derived without knowing the exact set Obs.

Definition 8 (Sol-Completeness) *A QAP is sol-complete, if all solutions at the level of target distinctions can be distinguished: $\forall\ \pi_{\text{targ},i}$, $\exists\ R_{\text{obs}} \in \text{Obs}$ such that $\Pi_i(\boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}}))) = \pi_{\text{targ},i}$.*

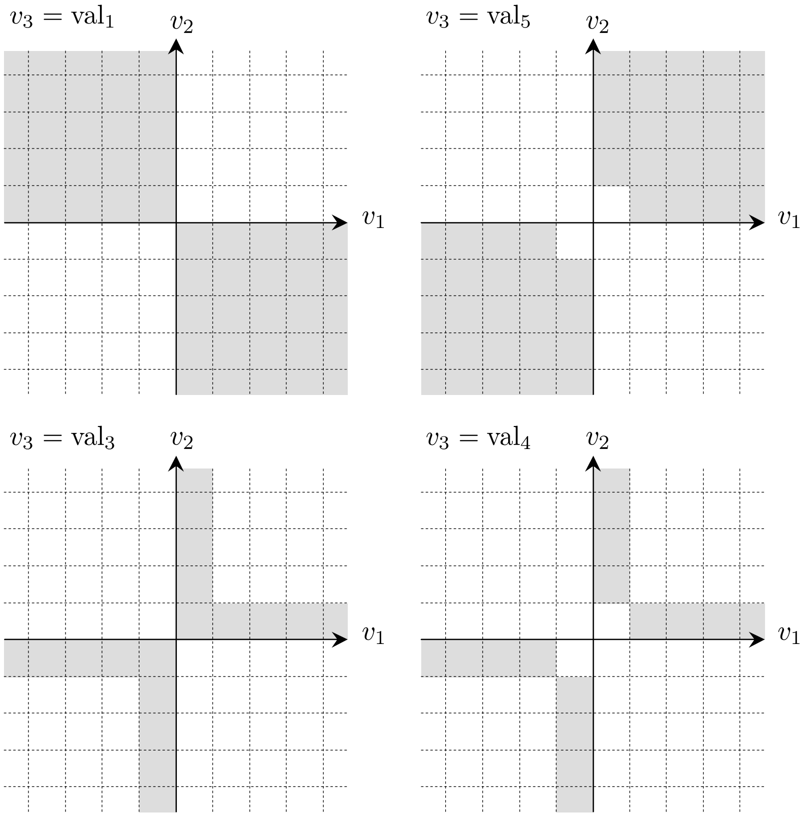

Fig. 4. Projections of the multiplication constraint in Example 6 on the qualitative values of v_3. The grid corresponds to the granularity of observations for v_1 and v_2. The diagram for $v_3 = \mathrm{val}_2$ (not shown) coincides with the axes v_1,v_2.

Sol-completeness means that all possible solutions defined by the target distinctions can indeed be distinguished based on the model and the external restrictions.

The results of our analysis can still be applied to QAPs that are not obs-complete or sol-complete, but then the resulting abstractions are not necessarily maximal (because there might be additional possibilities for abstraction that are not related to the coarseness of observable or target distinctions). In addition, we demand that $\boldsymbol{\tau}_{\mathrm{obs}}$ and $\boldsymbol{\tau}_{\mathrm{targ}}$ are piecewise comparable. Note that this is not actually a restriction, because it can be established for any QAP by possibly introducing additional variables that separate the target and observable distinctions.

Intuitively, if QAP is sol-complete, we have to keep all the target distinctions, because we need them to distinguish the solutions. But we can eliminate the distinctions between observations that would lead to the same set of solutions. If QAP is obs-complete, then the possible observations on the level of $\boldsymbol{\tau}_{\mathrm{obs}}$ are the possible subsets of $\boldsymbol{\tau}_{\mathrm{obs}}(\mathrm{dom}(\boldsymbol{v}))$. For each tuple $t_{\mathrm{obs},j} \in \boldsymbol{\tau}_{\mathrm{obs}}(\mathrm{dom}(\boldsymbol{v}))$, let

$R_{\mathrm{sol},j}$ be the solution it derives on the level of target distinctions:

$$R_{\mathrm{sol},j} := \boldsymbol{\tau}_{\mathrm{targ}}(R \bowtie t_{\mathrm{obs}}).$$

Let $R_{\mathrm{obs},k}$ denote the sets of tuples $t_{\mathrm{obs},j}$ that obtain the same solution:

$$R_{\mathrm{obs},k} := \bigcup_{j: R_{\mathrm{sol},j} = R_{\mathrm{sol},k}} t_{\mathrm{obs},j}.$$

Then the $R_{\mathrm{obs},k}$ form the elements of a partition of $\boldsymbol{\tau}_{\mathrm{obs}}(\mathrm{dom}(\boldsymbol{v}))$:

$$\Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}}) := \bigcup_{k} \{R_{\mathrm{obs},k}\}.$$

The set $\Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}})$ defines a partition over tuples at the level of observable distinctions, aggregating those that yield the same solutions. We illustrate these concepts with a simple example.

Example 9 (Equality) *Let* $\boldsymbol{v} = (v_1, v_2)$. *Let* $\mathrm{dom}(v_i) = \{0, 1, 2\}$ *for* $i = 1, 2$. *Let* R *be given as*

$$R = \{(0, 0), (1, 1), (2, 2)\}.$$

Assume that the only non-trivial observable partition is a partition for v_1*:*

$$\pi_{\mathrm{obs},1} = \{\{0\}, \{1\}, \{2\}\}, \pi_{\mathrm{obs},2} = \{\{0, 1, 2\}\},$$

and that the only non-trivial target partition is a partition for v_2*:*

$$\pi_{\mathrm{targ},1} = \{\{0, 1, 2\}\}, \pi_{\mathrm{targ},2} = \{\{0\}, \{1, 2\}\}.$$

Then the set $\Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}})$ *contains the two elements (see also Figure 5)*

$$R_{\mathrm{obs},1} = \{(\{0\}, \{0, 1, 2\})\}, R_{\mathrm{obs},2} = \{(\{1\}, \{0, 1, 2\}), (\{2\}, \{0, 1, 2\})\}.$$

For the pedal position sensor model (Table 2) with the observable and target distinctions specified in Examples 2 and 4, the partition $\Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}})$ consists of the four elements shown in Figure 6.

The following theorem shows that two domain values can be aggregated if they are not distinguished by a target distinction, and if they are interchangeable with respect to every relation in $\Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}})$.

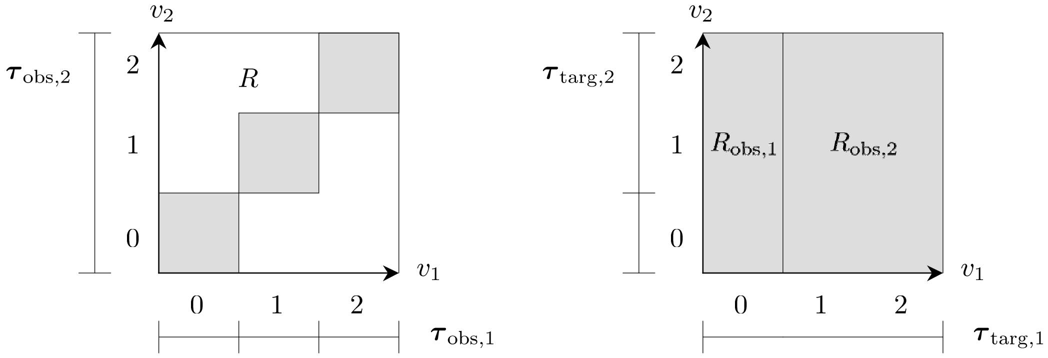

Fig. 5. The model relation R (left) and the two elements of the partition $\Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$ (right) for Example 9.

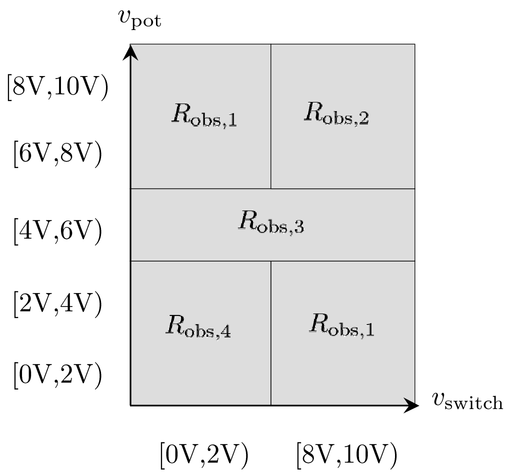

Fig. 6. Partition $\Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$ for the pedal position sensor example. The partition consists of four elements (all variables except v_{pot} and v_{switch} have no observable distinction and have been omitted from the figure).

Theorem 10 (Solution to QAP) *Let QAP be a qualitative abstraction problem that is obs-complete and sol-complete. Let $\tau_{\text{FI},\Lambda,i}$ be the domain abstraction that aggregates the interchangeable values of a relation Λ, that is, two values* $\text{val}_1, \text{val}_2 \in \text{dom}(v_i)$ *are combined if* [1]

$$\Pi_{1,\ldots,i-1,i+1,\ldots,n}(\sigma_{v_i=\text{val}_1}(\Lambda)) = \Pi_{1,\ldots,i-1,i+1,\ldots,n}(\sigma_{v_i=\text{val}_2}(\Lambda)).$$

Then the merge of $\tau_{\text{targ},i}$ and every domain abstraction

$$\tau_{\text{FI},\Lambda,i} \textit{ where } \Lambda \in \Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$$

is an induced abstraction for QAP.

[1] This relational notion of interchangeability is slightly more general than the original definition of Freuder [8] in that it includes also the case where val_1 or val_2 do not occur in the relation Λ.

A proof is given in the appendix. In Example 9, the values 1 and 2 for variable v_1 are interchangeable in both partition elements $R_{\mathrm{obs},1}$ and $R_{\mathrm{obs},2}$. Therefore, the two qualitative values $\{0\}$ and $\{1,2\}$ are derived for variable v_1. Variable v_2 receives the same distinctions because they are equal to its target partition.

For the pedal position sensor example, two pairs of values for v_{pot} are interchangeable in all partition elements $R_{\mathrm{obs},1},\ldots,R_{\mathrm{obs},4}$ (Figure 6). Theorem 10 derives the following three qualitative values for v_{pot}:

$$\{\{[0\mathrm{V},2\mathrm{V}),[2\mathrm{V},4\mathrm{V})\},\{[4\mathrm{V},6\mathrm{V})\},\{[6\mathrm{V},8\mathrm{V}),[8\mathrm{V},10\mathrm{V})\}\}.$$

The first qualitative value {[0V,2V), [2V,4V)} corresponds to situations where v_{switch} equals ground voltage, the third qualitative value {[6V,8V), [8V,10V)} corresponds to situations where v_{switch} equals battery voltage, and the second qualitative value {[4V,6V)} corresponds to situations where the position of the switch and, hence, the voltage of v_{switch}, is ambiguous.

Theorem 10 shows that the basic concept of interchangeability plays a central role in the determination of solutions to a qualitative abstraction problem. In particular, the problem of finding interchangeable values in a relation can be recast as a special case of a QAP, where one distinguishes only empty from non-empty solutions:

Corollary 11 (Interchangeability as QAP) *Let* $QAP = (R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}})$ *be an obs-complete qualitative abstraction problem such that* $\boldsymbol{\tau}_{\mathrm{obs}} = \boldsymbol{\tau}_{\mathrm{id}}$*,* $\boldsymbol{\tau}_{\mathrm{targ}} = \boldsymbol{\tau}_{\mathrm{triv}}$ [2]*. Then* $\tau_{\mathrm{FI},R,i}$ *is an induced abstraction for QAP.*

In general, however, the granularity of induced abstractions is different from the granularity of interchangeable values. Induced abstractions can be either more coarse or more fine-grained than $\boldsymbol{\tau}_{\mathrm{FI},R}$. The former case occurs in Example 9, where interchangeability would have distinguished between all the domain values for both v_1 and v_2. The latter case occurs if target distinctions are specified between domain values that would be interchangeable with respect to the model relation.

Theorem 10 provides only one solution, but a QAP might have multiple solutions in general. The following example illustrates this.

Example 12 (Multiple Solutions) *Let* $\boldsymbol{v} = (v_1, v_2)$*. Let* $\mathrm{dom}(v_1) = \{0,1,2,3\}$, $\mathrm{dom}(v_2) = \{0,1\}$*. Let R be given by*

$$R = \{(1,0),(2,1),(3,0),(3,1)\}.$$

[2] These conditions already imply that the QAP is sol-complete.

Assume that the only non-trival observable partition is a partition for v_1

$$\pi_{\mathrm{obs},1} = \{\{1\},\{2\},\{0,3\}\}, \pi_{\mathrm{obs},2} = \{\{0,1\}\}$$

and that the only non-trival target partition is given by a partition for v_2*:*

$$\pi_{\mathrm{targ},1} = \{\{0,1,2,3\}\}, \pi_{\mathrm{targ},2} = \{\{0\},\{1\}\}.$$

Then there a three different induced abstractions for v_1*:*

$$\begin{aligned}\pi_{\mathrm{ind},1} &= \{\{1\},\{2\},\{0,3\}\},\\ \pi'_{\mathrm{ind},1} &= \{\{1\},\{0,2\},\{3\}\},\\ \pi''_{\mathrm{ind},1} &= \{\{0,1\},\{2\},\{3\}\}.\end{aligned}$$

In Example 12, the domain value 0 of variable v_1 does not occur in the model relation R and can be freely allocated to different partition elements. Among the three possible solutions, Theorem 10 provides the induced abstraction $\pi_{\mathrm{ind},1}$, whose qualitative values are comprised of observable partition elements.

Theorem 10 also constitutes a possible starting point for finding approximations of qualitative values that avoid the cost of computing interchangeable values. One approach is to use only necessary conditions for interchangeability. A necessary condition for two domain values $\mathrm{val}_1, \mathrm{val}_2 \in \mathrm{dom}(v_i)$ to be interchangeable with respect to a relation Λ is that $\mathrm{val}_1, \mathrm{val}_2$ are interchangeable with respect to a projection of Λ on a subset of its variables. In the extreme case, we consider only the projection of Λ on the variable v_i itself:

Proposition 13 (Approximate Solution to QAP) *Let QAP be a qualitative abstraction problem that is obs-complete and sol-complete. Let* $\tau_{\mathrm{app},i}$ *be the merge of* $\tau_{\mathrm{targ},i}$ *and every domain abstraction*

$$\tau_{\mathrm{FI},\Lambda',i} \text{ where } \Lambda' := \Pi_i(\Lambda), \Lambda \in \Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}}).$$

Then there exists an induced abstraction for QAP that is a refinement of $\tau_{\mathrm{app},i}$*.*

Computing the approximation $\boldsymbol{\tau}_{\mathrm{app}}$ is easier than determining $\boldsymbol{\tau}_{\mathrm{ind}}$, because it involves only the projection and intersection of sets, and does not require to determine the interchangeable values in Λ.

The approximation $\boldsymbol{\tau}_{\mathrm{app}}$ corresponds to considering only observations for individual variables, and not simultaneous observations for different variables.

As illustrated by Example 14, it is not adequate because in general, an observation might lead to a different solution only if combined with additional observations for the other variables.

Example 14 (Xor-Gate) *Let* $\boldsymbol{v} = (v_1, v_2, v_3)$. *Let* $\mathrm{dom}(v_i) = \{0, 1\}$ *for* $i = 1, 2, 3$. *Let* R *be given as*

$$R = \{(0,0,0), (0,1,1), (1,0,1), (1,1,0)\}.$$

Assume that a non-trivial observable distinction is given only for variables v_1 *and* v_2 *and that a non-trivial target distinction is given only for* v_3*:*

$$\pi_{\mathrm{obs},1} = \{\{0\}, \{1\}\}, \pi_{\mathrm{obs},2} = \{\{0\}, \{1\}\}, \pi_{\mathrm{targ},3} = \{\{0\}, \{1\}\}.$$

Then $\Sigma(R, \boldsymbol{\tau}_{\mathrm{obs}}, \boldsymbol{\tau}_{\mathrm{targ}})$ *contains the two elements (see Figure 7)*

$$\begin{aligned} R_{\mathrm{obs},1} &= \{(\{0\}, \{0\}, \{0,1\}), (\{1\}, \{1\}, \{0,1\})\}, \\ R_{\mathrm{obs},2} &= \{(\{0\}, \{1\}, \{0,1\}), (\{1\}, \{0\}, \{0,1\})\}. \end{aligned}$$

The approximation yields only the trivial partition as a lower bound for the granularity of v_1 *and* v_2*. However, the induced abstraction corresponds to the granularity of the base domain:*

$$\pi_{\mathrm{ind},1} = \{\{0\}, \{1\}\}, \pi_{\mathrm{ind},2} = \{\{0\}, \{1\}\}.$$

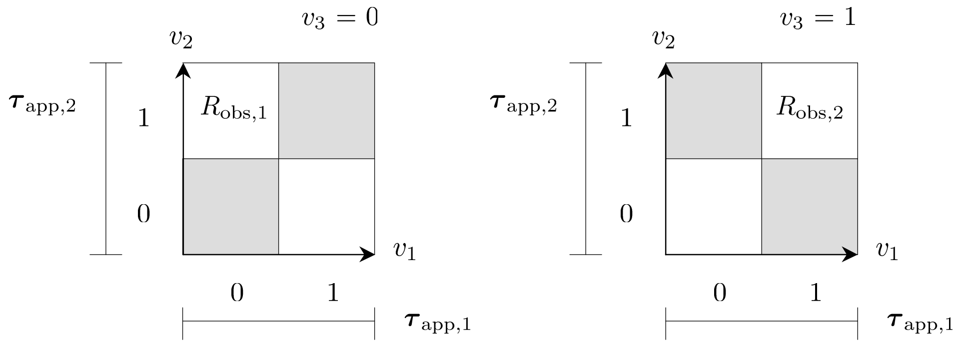

Fig. 7. Projection of $R_{\mathrm{obs},1}$ (left) and $R_{\mathrm{obs},2}$ (right) on variable v_1 yields no distinction for Example 14.

Proposition 13 yields a lower bound on the granularity of the induced abstractions derived by Theorem 10, which is often sufficient in practical cases. Both in Example 9 and in the pedal position sensor example (Figure 6), the distinctions derived by $\boldsymbol{\tau}_{\mathrm{app}}$ are identical to those derived by Theorem 10.

4 A Prototypic System for Task-dependent Domain Abstraction

The computation of induced abstractions for a QAP involves, based on the results above, the subproblems of constructing the model relation R, computing the partition $\Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$, and determining interchangeable values within the elements of this partition.

Our prototypic system AQUA (Automated Qualitative Abstraction) [19] determines the relation R through structural decomposition of the constraint network defined by the model fragments it is composed of. Structural decomposition [10] transforms a constraint network into an equivalent acyclic (tree-structured) instance. AQUA then iterates over the partition elements of the observable and target distinctions and labels the tuples of the relation R that are consistent with the respective partition elements. Since directional arc consistency is sufficient for establishing consistency in a tree-structured network, this step can be performed efficiently by local constraint propagation. This step yields the partition $\Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$.

Interchangeable values in the partition elements of $\Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$ can then be found using the basic algorithm described in [8]. Alternatively, the partition elements can be projected on the individual variables to obtain the approximate solution. AQUA also performs further optimizations in that it can automatically remove redundant values (domain values that do not appear in any constraint) and eliminate variables that have no distinction at all. The decomposition step is independent of the particular task in terms of observable and target distinctions; hence, the resulting tree can be re-used for different combinations of observable or target distinctions.

AQUA builds on components of an existing model-based reasoning framework called Raz'r that consists of a development system for composing a device model from of a library of model fragments, and a runtime system for performing behavioral prediction and diagnosis based on actual measurements for the device. In addition to the basic Raz'r components, AQUA includes a module that computes induced abstractions as described above, an abstractor module that applies domain abstractions to a real-valued or finite behavior model, and a signal transformation module that generates qualitative observations by applying domain abstractions to (time-varying) measurements.

Using AQUA to automate qualitative domain abstraction, several tasks can be supported in the context of building model-based systems that are often carried out manually or solved on an ad hoc-basis. The common theoretical basis is to find suitable domains for the variables in a model. However, in different contexts this basic task can have different interpretations, depending on what the terms variable and domain refer to, including magnitudes, modes of components, and deviations from reference behaviors.

5 Application: On-board Diagnosis of a Passenger Vehicle

In a project involving European car manufacturers and suppliers [21], task-dependent qualitative domain abstraction was used to build a prototype of a model-based system capable of diagnosing emission-related failures of turbo-charged diesel engines, a problem of significant importance regarding environmental impact and compliance with legal requirements.

Fig. 8. View of the Demonstrator Car

This system had to make use of the sensor signals available on-board in the car, transform them to a qualitative level and exploit them for detecting and localizing faults based on a model of the turbo control system of the diesel engine (Figure 9). It was installed on a demonstrator vehicle with a number of built-in faults (see Figure 8).

As part of the project, numerical (real-valued) models were developed for the relevant components, including a characteristic map describing the complex behavior of the engine. The resulting model was composed of 16 fragments and had 146 variables (see [19] for details).

The particular interest of the involved car manufacturers concerned failures of the system that lead to increased carbon emissions due to an excessive quantity of fuel injected or an insufficient airflow to the engine. The fuel combustion process is largely determined by λ, the stoichiometric ratio between air and

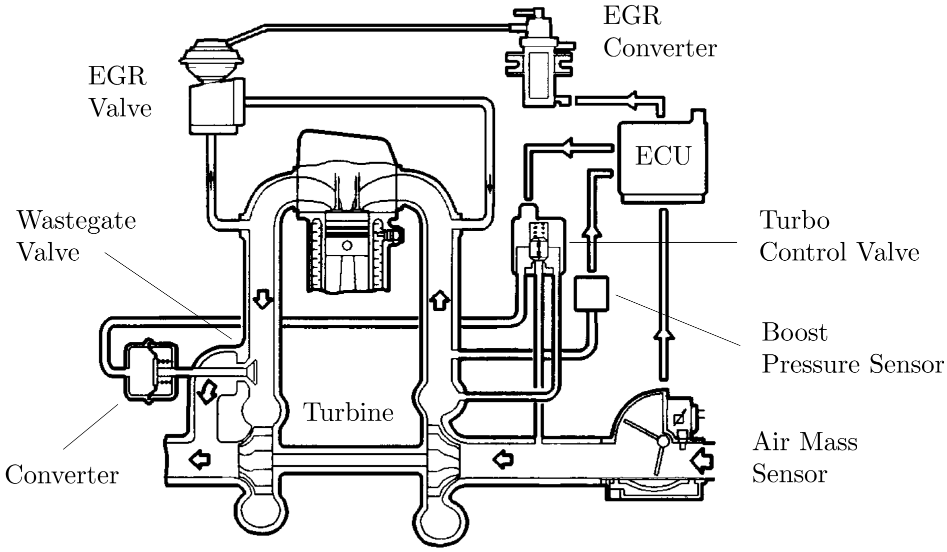

Fig. 9. Turbo Control and exhaust gas re-circulation subsystem of the vehicle. Through the EGR valve and the wastegate valve, the ECU controls the air supply to the engine by determining the amount of exhaust gas that will be fed back to the inlet and that drives the turbocharger turbine, respectively.

fuel:

$$\lambda = \frac{\text{actual air quantity}}{\text{theoretical air requirement}}.$$

A fuel-lean mixture ($\lambda > 1$) contains more air, while a fuel-rich mixture ($\lambda < 1$) contains less air. Hence, in our framework, the modeling goal to distinguish normal situations from situations where combustion is incomplete (and therefore increased carbon emissions occur) could be expressed as a target partition for λ, stating whether it is above or below a certain critical value λ_{crit}:

$$\pi_{\text{targ},\lambda} = \{(-\infty, \lambda_{\text{crit}}), [\lambda_{\text{crit}}, \infty)\}.$$

The fact that only certain variables in the system are measured could be expressed as an observable distinction for the variables in the system. It associates the identical domain mapping with variables that correspond to signals of the control unit, and the trivial domain mapping with all the other variables.

Abstraction of the model was performed in two steps. In a first step, the real-valued model fragments were turned (using AQUA's model abstractor) to a finite system description by applying initial domain mappings from the real numbers to discrete domains (these initial mappings could be chosen arbitrarily, but had to be sufficiently fine-grained to ensure sol-completeness). In a second step, task-dependent domain abstraction was used to further reduce these discrete values to qualitative values, eliminating any of the initial dis-

Table 3
Typical example showing the performance of the initial model and the model obtained by task-dependent domain abstraction during a time frame of 9.75 seconds

	Quantitative (no abstraction)	Initial abstraction	Qualitative abstraction
Number of Observations	1053	28	12
Runtime for Diagnosis [3]	–	2.79 sec	1.84 sec

tinctions that were irrelevant to the task of diagnosing incomplete combustion.

The two steps took AQUA roughly 3 minutes. Both models were then confronted with actual measurements taken from the car. Table 3 shows the results for a continuously running measurement that involved a leakage fault in the pipe between the turbine and the engine inlet. The duration of the shown time frame is approximately 10 seconds, and 1053 quantitative observation vectors occur during this time frame.

The initial model abstraction reduced this number to 28 different vectors. Compared to the initial abstraction, qualitative domain abstraction achieved a further significant reduction. In the example time frame, only 12 of the 28 initial discrete vectors are found to be qualitatively different, hence the frequency of observations is reduced from approximately 2.8 observations per second to 1.2 observations per second. The qualitative model yields the same diagnostic result about one third faster than the initial model.

Qualitative domain abstraction reduces both the complexity of reasoning with the model, and the number of time points at which this reasoning has to be initiated. In the context of the project, these two effects were instrumental to design a diagnostic system that could meet real-time requirements and was actually capable of keeping up with the rate of the measurements in the on-board context.

6 Discussion

While several pieces of work have addressed the problem of automatically deriving appropriate models [26, 16, 17, 28, 14, 18], the work presented here is distinctive in that it focuses specifically on the granularity (resolution) of the domain values. This limited scope enables a less knowledge-based, more "mathematical" view of automated modeling. It allows for representing the

[3] Apart from the runtime required for diagnosis, there are further processing steps (for instance, signal processing) that are similar for both models and not shown in the table.

space of candidate models implicitly and concisely as the space of possible domain partitions, and it allows for taking on the view of transforming (reformulating) models by means of well-defined operators τ_i. In comparison, other approaches to automated modeling either require to explicitly enumerate the space of candidate models [1, 22], or at least to pre-define a set of fragments with different levels of detail to choose from [16, 17, 14, 18] (note that there might be an infinite number of candidate models in the case of domain abstraction). Another difference is that our general relational representation subsumes both infinite and finite constraints, and is not limited to specific types of constraints or special-purpose reasoning methods (for instance, [16, 17, 28] employ variants of order-of-magnitude reasoning). It leaves reasoning with the model to any problem solver that can handle the relational operations join, projection and selection. Together with the narrow focus on domain abstraction, the relation-based formalization allows for capturing the conditions for a solution in a single, concise formula (Definition 5) and, more importantly, allows for determining the solutions analytically and in closed form (Theorem 10). In contrast, [16, 17, 14, 18] all devise search procedures that start from an initial model and backtrack until they find a solution.

Williams [27] defines a so-called hybrid algebra for automated abstraction of a behavior model, with the goal of preserving information about the sign of the result as far as possible. The approach thus captures the idea of obtaining, given a base model, optimal information with respect to a targeted granularity (in this case, the signs of the variables). However, the specific domain abstraction to signs is hard-wired into the transformation rules of the algebra, and the constraints are restricted to operators such as addition or multiplication. QSIM [12, 13], a system for performing qualitative simulation of device behavior over time, incorporates methods for refining the domains of variables by deriving new distinctions ("landmarks") during the simulation process. The base domain is given as the real numbers, and new landmarks are introduced whenever the derivative of a variable reaches zero. Except for signs, only information on the ordinal relationship and knowledge about values that must be assumed at the same time ("corresponding values") is provided. Therefore, the mapping of qualitative values to their base domain is only partially known, and the derived distinctions can in general not be exploited to simplify the constraints. Extensions of QSIM that deal with semi-quantitative reasoning [2] allow to further constrain the landmark values to numeric intervals. However, these methods are specific to the context of simulation, and the constraints are limited to a set of algebraic relationships and monotonic functions.

The notion of observable and target distinctions generalizes notions of task-dependent characteristics that have been previously exploited in constraint-based and model-based reasoning. Chung [4] presents a compilation method for diagnostic models that eliminates the non-observable variables. This corresponds to a special case of observable distinction that is equal to $\tau_{\mathrm{id},i}$ for the observables and equal to $\tau_{\text{triv},i}$ for all other variables. In constraint optimization, it is common to distinguish between decision variables that appear in the solutions and non-decision variables that do not appear in the solutions. This can be viewed as a special case of target distinctions that is equal to $\tau_{\text{id},i}$ for the decision variables and equal to $\tau_{\text{triv},i}$ for the non-decision variables. Torasso and Torta [24] recently presented an approach for merging together behavior modes that are indistinguishable, based on a notion of observation granularity that is equivalent to observable distinctions. However, the method does not incorporate a notion of target distinctions.

The theory of task-dependent domain abstraction is applicable both to finite and infinite models. However, our current implementation (AQUA) is based on a finite-domain representation and can derive induced abstractions only for the finite case. As outlined in Section 5, it is possible to *approximate* induced distinctions for real-valued models by first applying an initial discretization, and then generating induced abstractions for this finite representation. To mitigate the problem of finding "good" initial discretizations, [19] develops a method for iterative refinement of qualitative values. [23] investigates cases of real-valued functions for which exact induced distinctions can be obtained. For the special case of real-valued monotonic functions and target distinctions that can be expressed as landmarks, deriving induced abstractions becomes similar to the problem of finding corresponding values for landmarks [13]. Note that observability of a real-value variable is most naturally specified as a certain range encompassing a measured value, reflecting the accuracy of the measurement. Since this does not correspond to a partition of the domain values of the variable [4], extending task-dependent domain abstraction to infinite domains might also have ramifications on the notion of task-dependency.

7 Conclusion

The increasing complexity of engineered devices has lead to an increased demand for computer-supported behavior prediction, diagnosis, and testing. Given the maturity and scale of model-based systems, the question of how to re-use behavior models is of growing interest. It has been shown that a model composed from a library cannot be expected to have a level of granularity suitable for different tasks right away. Instead, the ability to re-formulate the model after composing it is a crucial requirement. We identified, within a common relational framework, fundamental properties of re-formulation that is based on abstraction of domain values. The degree of domain abstraction

[4] In some situations, this case could be handled by re-formulating the behavior model in terms of variables expressing deviations (see [26, 5]) and stating observable distinctions for these deviations.

that can be achieved is strongly dependent on the characteristics of the task in terms of available inputs and required outputs. Observable distinctions and target distinctions are a means to capture these aspects, and they can be exploited to derive qualitative values as distinctions that are both adequate and as coarse as possible. Task-dependent qualitative domain abstraction is a contribution to further bridging the gap between quantitative and qualitative modeling, as it allows for expressing knowledge about component behavior without committing early to a specific abstraction level of the domains. It can help to make model-based system more efficient and more effective due to automating steps that are currently done by hand.

Acknowledgements

The authors thank the two anonymous referees for useful comments and suggestions, and Claes Carlén, Oskar Dressler, Ulrich Heller, Andreas Malik, Jakob Mauss, Björn Svensson and Daniele Theseider-Dupré for fruitful collaboration. This research was funded in part by the European Union through contract BE95/2128 and by Deutsche Forschungsgemeinschaft through grant SA1000/1-1.

A Appendix: Proof of Theorem 10

We have to show that $\boldsymbol{\tau}_{\text{ind}}$, defined as the merge of $\boldsymbol{\tau}_{\text{targ}}$ and all abstractions $\boldsymbol{\tau}_{\text{FI},\Lambda}$, is both adequate and maximal. First, we show that $\boldsymbol{\tau}_{\text{ind}}$ is adequate.

Let R_{obs} be an external restriction. We need to show $\boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})) = \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}_{\text{ind}}(\boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})))$. The direction "$\subseteq$" is obvious. To show the direction "$\supseteq$", consider the relation $\Gamma \supseteq \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})$ that contains, for every tuple in $\boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})$, its partition element Λ. Then because Λ combines only tuples that yield the same solution, $\boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})) = \boldsymbol{\tau}_{\text{targ}}(R \bowtie \Gamma)$. Let $\boldsymbol{\tau}_{\text{FI},\Sigma}$ be the merge of every domain abstraction $\boldsymbol{\tau}_{\text{FI},\Lambda}$. Then because $\boldsymbol{\tau}_{\text{FI},\Lambda}$ aggregates only interchangeable values, and because $\boldsymbol{\tau}_{\text{ind}}$ is a refinement of $\boldsymbol{\tau}_{\text{FI},\Sigma}$, $R \bowtie \Gamma = R \bowtie \boldsymbol{\tau}_{\text{FI},\Sigma}(\Gamma) = R \bowtie \boldsymbol{\tau}_{\text{ind}}(\Gamma)$. Because $\boldsymbol{\tau}_{\text{ind}}$ is a refinement of $\boldsymbol{\tau}_{\text{targ}}$, $\boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{ind}}(\Gamma)) = \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R \bowtie \boldsymbol{\tau}_{\text{ind}}(\Gamma)) = \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}_{\text{ind}}(\Gamma))$. From $\Gamma \supseteq \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})$, it follows that $\boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}_{\text{ind}}(\Gamma)) \supseteq \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}_{\text{ind}}(\boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})))$.

Second, we show that $\boldsymbol{\tau}_{\text{ind}}$ is maximal. We need to show that for any abstraction $\tau'_{\text{ind},i}$ that further combines partition elements of $\tau_{\text{ind},i}$, at least one external restriction exists that yields a solution R_{sol} for the original model and a different solution $R'_{\text{sol}} \neq R_{\text{sol}}$ for the abstracted model. Because the observable and target distinctions are piecewise comparable, there are only two possibilities how a simpler abstraction $\tau'_{\text{ind},i}$ can be formed:

(1) If $\tau_{\text{targ},i}$ is a refinement of $\tau_{\text{obs},i}$, then $\tau_{\text{ind},i} = \tau_{\text{targ},i}$, and $\tau'_{\text{ind},i}$ must combine at least two partition elements of $\tau_{\text{targ},i}$.

(2) If $\tau_{\text{obs},i}$ is a refinement of $\tau_{\text{targ},i}$, then if $\tau'_{\text{ind},i}$ does not combine partition elements of $\tau_{\text{targ},i}$, it must combine at least two partition elements of $\tau_{\text{FI},\Lambda,i}$.

In the case where $\tau'_{\text{ind},i}$ combines at least two partition elements p_1, p_2 of $\tau_{\text{targ},i}$, because of sol-completeness, there exists at least one external restriction R_{obs} that yields a solution $R_{\text{sol}} := \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}_{\text{ind}}(\boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})))$ such that $\Pi_i(R_{\text{sol}}) = p_1$. Then for $R'_{\text{sol}} := \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}\text{'}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}\text{'}_{\text{ind}}(\boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})))$, it holds that $\Pi_i(R'_{\text{sol}}) = p_1 \cup p_2$, and therefore $R_{\text{sol}} \neq R'_{\text{sol}}$.

In the case where $\tau'_{\text{ind},i}$ combines at least two partition elements p_1, p_2 of $\tau_{\text{FI},\Lambda,i}$, because p_1, p_2 are distinguished in $\tau_{\text{FI},\Lambda,i}$, there exists at least one partition element Λ of $\Sigma(R, \boldsymbol{\tau}_{\text{obs}}, \boldsymbol{\tau}_{\text{targ}})$ for which p_1, p_2 are not interchangeable, that is, where $\Pi_{1,\ldots,i-1,i+1,\ldots,n}(\sigma_{v_i \in p_1}(\Lambda)) \neq \Pi_{1,\ldots,i-1,i+1,\ldots,n}(\sigma_{v_i \in p_2}(\Lambda))$. Let R_{obs} be defined as

$$R_{\text{obs}} := p_2 \times \Pi_{1,\ldots,i-1,i+1,\ldots,n}(\sigma_{v_i \in p_1}(\Lambda)) \cup p_1 \times \Pi_{1,\ldots,i-1,i+1,\ldots,n}(\sigma_{v_i \in p_2}(\Lambda)).$$

R_{obs} comprises exactly the tuples that are missing to make p_1, p_2 interchangeable with respect to Λ. Because of obs-completeness, R_{obs} and Λ can occur as observations, and they yield two solutions $R_{\text{sol}} := \boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{obs}}(R_{\text{obs}})) = \boldsymbol{\tau}_{\text{targ}}(R \bowtie R_{\text{obs}})$, $R_{\text{sol}} := \boldsymbol{\tau}_{\text{targ}}(R \bowtie \boldsymbol{\tau}_{\text{obs}}(\Lambda)) = \boldsymbol{\tau}_{\text{targ}}(R \bowtie \Lambda)$. Because Λ comprises, by definition, all tuples that yield the same solution $R_{\text{sol},\Lambda}$, it holds that $R_{\text{sol}} \neq R_{\text{sol},\Lambda}$. Then for $R'_{\text{sol}} := \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}\text{'}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}\text{'}_{\text{ind}}(R_{\text{obs}}))$, $R'_{\text{sol},\Lambda} := \boldsymbol{\tau}_{\text{targ}}(\boldsymbol{\tau}\text{'}_{\text{ind}}(R) \bowtie \boldsymbol{\tau}\text{'}_{\text{ind}}(\Lambda))$ it holds that $R'_{\text{sol}} = R'_{\text{sol},\Lambda} = R_{\text{sol}} \cup R_{\text{sol},\Lambda}$, and because $R_{\text{sol}} \neq R_{\text{sol},\Lambda}$, it follows that either $R_{\text{sol}} \neq R'_{\text{sol}}$ or $R_{\text{sol},\Lambda} \neq R'_{\text{sol},\Lambda}$.

References

[1] S. Addanki, R. Cremonini, and J. S. Penberthy. Graphs of models. *Artificial Intelligence*, 51(1–3):145–177, 1991.

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

[3] F. Cascio, L. Console, M. Guagliumi, M. Osella, A. Panati, S. Sottano, and D. Theseider Dupré. Generating on-board diagnostics of dynamic automotive systems based on qualitative models. *AI Communications, Special Issue on Model–Based Reasoning*, 1999.

[4] S. Chung, J. Van Eepoel, and B.C. Williams. Improving model-based mode estimation through offline compilation. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, St-Hubert, Canda, 2001.

[5] Hidde de Jong and Frank van Raalte. Comparative envisionment construction: A technique for the comparative analysis of dynamical systems. *Artificial Intelligence*, 115(2), 1999.

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

[7] Boi Faltings and Peter Struss, editors. *Recent Advances in Qualitative Physics.* The MIT Press, Cambridge, Massachusetts, 1992.

[8] E. C. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proceedings of Ninth National Conference on Artificial Intelligence*, pages 227–233, Anaheim, CA, 1991.

[9] E. C. Freuder and D. Sabin. Interchangeability supports abstraction and reformulation for constraint satisfaction. In *Proceedings of Symposium on Abstraction, Reformulation and Approximation (SARA'95)*, 1995.

[10] Georg Gottlob, Nicola Leone, and Francesco Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124(2):243–282, 2000.

[11] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-based Diagnosis.* Morgan Kaufman Publishers, San Mateo, CA, 1992.

[12] B. J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–338, 1986.

[13] Benjamin J. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* MIT Press, Cambridge, MA, 1994.

[14] A. Levy, Y. Iwasaki, and R. Fikes. Automated model selection for simulation based on relevance reasoning. *Artificial Intelligence*, 96:351–394, 1997.

[15] N. Muscettola, P. Nayak, B. Pell, and B.C. Williams. Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103:5–48, 1998.

[16] P. Pandurang Nayak. Causal approximations. *Artificial Intelligence*, 70:277–334, 1994.

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

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

[19] M. Sachenbacher. *Automated Qualitative Abstraction and its Application to Automotive Systems.* PhD thesis, Technische Universität München, Department of Computer Science, 2001.

[20] M. Sachenbacher, P. Struss, and C. Carlén. A prototype for model-based on-board diagnosis of automotive systems. *AI Communications*, 13(2):83–97, 2000.

[21] M. Sachenbacher, P. Struss, and R. Weber. Advances in design and implementation of OBD functions for diesel injection systems based on a qualitative approach to diagnosis. In *Society of Automotive Engineers (SAE) World Congress*, Detroit, USA, 2000.

[22] P. Struss. What's in SD? Towards a theory of modeling for diagnosis. In L. Console W. Hamscher and J. de Kleer, editors, *Readings in Model–based Diagnosis*, pages 419–450, San Mateo, CA, 1992. Morgan Kaufmann.

[23] P. Struss. Automated abstraction of numerical simulation models: Theory and practical experience. In *Proceedings of the International Workshop on Qualitative Reasoning*, Sitges, Spain, 2002.

[24] G. Torta and P. Torasso. Automatic abstraction in component-based diagnosis driven by system observability. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[25] D. Weld and J. de Kleer, editors. *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann, San Mateo, CA, 1990.

[26] D. S. Weld. Reasoning about model accuracy. *Artificial Intelligence*, 56, 1992.

[27] B. C. Williams. A theory of interactions: Unifying qualitative and quantitative algebraic reasoning. *Artificial Intelligence*, 51:39–94, 1991.

[28] K. Yip. Model simplification by asymptotic order of magnitude reasoning. *Artificial Intelligence*, 80, 1996.