# Automated Qualitative Domain Abstraction

**Martin Sachenbacher**
MIT Artificial Intelligence Laboratory
Cambridge, MA 02139, USA
sachenba@mit.edu

**Peter Struss**
TU München, Fakultät für Informatik
85748 Garching, Germany
struss@in.tum.de

## Abstract

Automated problem-solving for engineered devices is based on models that capture the essential aspects of their 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 is to determine partitions for the domains of the variables (termed qualitative values) that are both necessary and sufficient for the task at hand. We present a formalization of the problem within a common relational (constraint-based) framework, present results regarding solutions to task-dependent qualitative domain abstraction, and devise methods for automatically determining qualitative values for a device model.

## 1 Introduction

Model-based systems ([Hamscher *et al.*, 1992; Weld and de Kleer, 1990]) represent knowledge about the structure and behavior of a physical system in terms of a behavior model, and use it to support engineering tasks such as behavior prediction, diagnosis, planning and testing. 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 ([Falkenhainer and Forbus, 1991]). 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 may make 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 sets of meaningful qualitative values. Although some work has been carried out on finding qualitative values within a specific context, such as simulation ([Kuipers, 1986]), the general problem of characterizing and systematically deriving qualitative values for an arbitrary relational (constraint-based) behavior model is a relatively unexplored area. However, much of the work in qualitative reasoning about physical systems ([Weld and de Kleer, 1990]) 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. Furthermore, 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.

### 1.1 Example

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.

The pedal position is sensed in two ways, via the potentiometer as an analogue signal, $v_{pot}$, and via the idle switch as a binary signal, $v_{switch}$. The idle switch changes its state at a particular value $pos_{switching}$ of the mechanically transferred

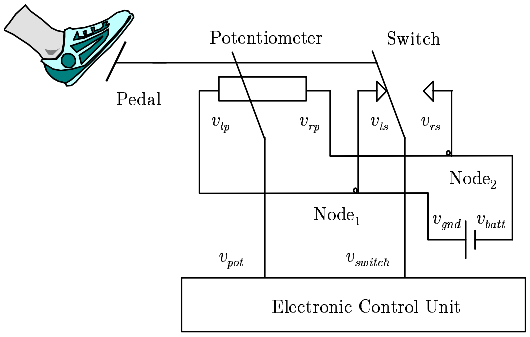

Figure 1: The Pedal Position Sensor

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 on-board 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 $pos_{switching}$ 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 modes and effects analysis (FMEA), 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 inacceptable having to *manually* create models from scratch that are tailored to each of these tasks.

## 2 Task-dependent Distinctions

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 constraints of the model is a highly practical requirement. It means grouping together those 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 ([Freuder, 1991]). For a constraint satisfaction problem 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 $v = val_1$, substituting $v = val_2$ 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 value $val_1$ $(val_2)$ itself.

Interchangeable values define equivalence classes on the domains of the variables, and grouping them together corresponds to an abstraction of the constraint satisfaction problem that exactly preserves the set of its solutions. Freuder and Sabin ([Freuder, 1991], [Freuder and Sabin, 1995]) 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 of domain values does not occur very frequently.

A model-based problem solving task, such as behavioral prediction or diagnosis, can be cast as an instance of a constraint satisfaction problem. However, it is particular 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 device model are observable, or because domain values cannot be observed beyond a certain granularity.

(2) the *output* involves not all feasible assignments of domain values to variables, but only certain aspects of the solutions are required. Typically, we might be interested in knowing whether values remain below or exceed a certain threshold, or what the values of mode variables of components are (for a diagnostic task), etc.

The idea pursued in this paper is to generalize the basic principle of interchangeability, leveraging on this specific context of a model-based problem solving task. The notion of a task is 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. They can be exploited to obtain so-called *induced abstractions* — domain abstractions that go beyond the low, generic level of interchangeability, but are still adequate for the given 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. A relational (or constraint-based) behavior model is a subset

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

that restricts the possible combinations of values for the variables $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, where $DOM(v_i)$ denotes the domain of a variable $v_i$. We use join ($\bowtie$), projection ($\Pi$), and selection ($\sigma$) as operators on relations.

Consider again the device shown in figure 1. Assume that the domain is

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

for variables involving voltage and

$$\{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$$

for variables involving position, and that the only parameter in the system, $pos_{switching}$, equals 40%. Then the device can be modeled by a relation $R$ that consists of 10 tuples:

| $v_{pot}$ | $position$ | $v_{switch}$ | $\dots$ |
|---|---|---|---|
| [0V,2V) | 0%, 20% | [0V,2V) | $\dots$ |
| [2V,4V) | 20%, 40% | [0V,2V) | $\dots$ |
| [4V,6V) | 40% | [0V,2V) | $\dots$ |
| [4V,6V) | 60% | [8V,10V) | $\dots$ |
| [6V,8V) | 60%, 80% | [8V,10V) | $\dots$ |
| [8V,10V) | 80%, 100% | [8V,10V) | $\dots$ |

Observable distinctions reflect the measurement granularity or incomplete observability of variables. 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 $\pi_{obs,i}$, is a partition of its domain $DOM(v_i)$.*

A variable $v_i$ is not observable at all if $\pi_{obs,i}$ is equal to the trivial domain partition $\pi_{triv,i} := \{DOM(v_i)\}$. For instance, the fact that for the pedal position sensor, the control unit observes the signal $v_{pot}$ and the signal $v_{switch}$ can be expressed by the observable distinction

$$\pi_{obs,v_{pot}} = \{\{[0V,2V)\}, \dots, \{[8V,10V)\}\},$$
$$\pi_{obs,v_{switch}} = \{\{[0V,2V)\}, \dots, \{[8V,10V)\}\},$$

whereas all other variables receive the trivial domain partition.

Target distinctions reflect the granularity of solutions we are after. Analogously to observable distinctions, target distinctions are expressed as domain partitions:

**Definition 2 (Target Distinction)** *A target distinction for a variable $v_i$, denoted $\pi_{targ,i}$, is a partition of its domain $DOM(v_i)$.*

A variable $v_i$ is said to have no target partition, if $\pi_{targ,i}$ is equal to the trivial partition. For instance, the target distinctions for the pedal position sensor are determined by the goal to distinguish between the domain values for the variable $v_{switch}$ (the plausibility check itself is not represented in the model):

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

## 2.1 Domain Abstractions

A domain partition $\pi_i$ can also be understood as a domain abstraction

$$\tau_i : DOM(v_i) \rightarrow DOM'(v_i) \subseteq 2^{DOM(v_i)}$$

that maps elements $val$ from a base domain $DOM(v_i)$ to a transformed domain $DOM'(v_i)$ that consists of sets of values from the base domain such that $val \in \tau_i(val)$. Abstractions can be straightforwardly extended from single values to sets of values, by taking the union of the resulting sets. For two abstractions $\tau_i$ and $\tau_i'$, $\tau_i$ is a *refinement* of $\tau_i'$, if

$$\forall val \in DOM(v_i) : \tau_i(val) \subseteq \tau_i'(val).$$

Two abstractions $\tau_i$, $\tau_i$' are *comparable*, if $\tau_i$ is a refinement of $\tau_i$' or $\tau_i$' is a refinement of $\tau_i$. We apply the notion of refinement and comparability equally to abstractions and domains. A vector of domain abstractions is denoted

$$\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_n).$$

A domain abstraction $\boldsymbol{\tau}$ is a refinement of $\boldsymbol{\tau'}$, if every $\tau_i$ is a refinement of $\tau_i'$. Two abstractions $\boldsymbol{\tau}$, $\boldsymbol{\tau'}$ are *piecewise comparable*, if every $\tau_i$ is comparable with $\tau_i'$. If a join operation combines two relations that are defined on different, but piecewise comparable domains, we define, for convenience, that the result is a relation over the finer domains.

# 3 Qualitative Abstraction Problems

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

**Definition 3 (Qualitative Abstraction Problem)** *Let $R$ be a relational behavior model, $OBS$ a set of external restrictions, $\boldsymbol{\tau}_{obs}$ a domain abstraction defined by observable distinctions, and $\boldsymbol{\tau}_{targ}$ a domain abstraction defined by target distinctions. The qualitative abstraction problem consists of finding an induced domain abstraction $\boldsymbol{\tau}_{ind}$ such that:*

*(1) (Adequacy) For all external restrictions $R_{obs} \in OBS$, $R_{obs} \subseteq DOM(\boldsymbol{v})$,*

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

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

The first condition (adequacy) states that the abstracted model $\boldsymbol{\tau}_{ind}(R)$ 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 all the possible external restrictions (actual observations, design specifications, etc.) on the level of observable distinctions. This guarantees that for any external restriction that can occur, the abstracted model will yield the same results as the original model. That is, if we apply $\boldsymbol{\tau}_{ind}$ *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 $\boldsymbol{\tau}_{ind}(R)$ for the original model $R$ in problem solving.

In general, there may be many domain abstractions that fulfill the adequacy criterion. In particular, the identical domain abstraction $\boldsymbol{\tau}_{id}$ that retains all the distinctions in the model is an adequate abstraction according to the definition. However, among all adequate abstractions, we prefer those that are the "simplest" ones. Simplicity of an adequate abstraction is defined in the second condition of definition 3. The approach taken is to select the abstractions that are coarsest in the sense that there exits no other adequate abstraction of which they would be a strict refinement (an abstraction that would further merge at least two of the qualitative values). A domain abstraction that is both adequate and simple incorporates only distinctions that are both necessary and sufficient according to the target and observable distinctions. It represents a level of abstraction that neither makes any unnecessary distinctions, nor abstracts away any distinctions that are crucial to solve the problem. Definition 3 thus formalizes the problem of finding qualitative values for the domains of variables. Compared to interchangeability, which is concerned with possibilities for abstraction within a single problem instance only, a qualitative abstraction problem (QAP) describes a whole class of instances defined by a model relation, a set of external restrictions, and the task-dependent distinctions. Interchangeability enforces that the set of solutions remains the same as for the original model. A QAP generalizes (or relaxes) this basic principle and demands that the set of solutions remains the same but only on the level of target distinctions, and only for inputs on the level of observable distinctions.

**Definition 4 (Properties of QAP)** *A QAP with a set of external restrictions $OBS$ is said to be obs-complete, if $\{\boldsymbol{\tau}_{obs}(R_{obs}) \mid R_{obs} \in OBS\} = 2^{\boldsymbol{\tau}_{obs}(DOM(\boldsymbol{v}))}$. It is said to be sol-complete, if for all $v_i$: $\tau_{targ,i}(DOM(v_i)) \subseteq \Pi_i(\{\boldsymbol{\tau}_{targ}(R \bowtie \boldsymbol{\tau}_{obs}(R_{obs})) \mid R_{obs} \in OBS\})$.*

Obs-completeness means that all the possible observations on the level of observable distinctions can actually occur or have to be considered during problem-solving (consequently, induced abstractions can be derived without knowing the exact set $OBS$). Sol-completeness means that all the possible solutions defined by the target distinctions can indeed be distinguished based on the model and the external restrictions. In addition, we demand that $\boldsymbol{\tau}_{obs}$ and $\boldsymbol{\tau}_{targ}$ 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, under these conditions, we expect that 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, the $\boldsymbol{\tau}_{obs}(R_{obs})$ are given by the possible subsets of $\boldsymbol{\tau}_{obs}(DOM(\boldsymbol{v}))$. For each tuple $\boldsymbol{val}_{obs,j} \in \boldsymbol{\tau}_{obs}(DOM(\boldsymbol{v}))$, define $R_{sol,obs,j}$ to be the solution it derives on the level of target distinctions:

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

Let $R_{obs,k}$ denote the sets of $\boldsymbol{val}_{obs,j}$ that obtain the same solution, i.e. those for which $R_{sol,obs,j}$ is equal:

$$R_{obs,k} := \bigcup_{j:R_{sol,obs,j}=R_{sol,obs,k}} \boldsymbol{val}_{obs,j}.$$

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

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

Figure 2 shows the resulting partition $\Sigma(R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$ for the pedal position sensor example, given the observable and target distinctions stated in Sec. 2 (all variables except $v_{pot}$ and $v_{switch}$ have no distinction and have been omitted from the figure; values [2V,4V), [4V,6V) and [6V,8V) for $v_{switch}$ do not appear in $R$ and have been omitted from its domain).

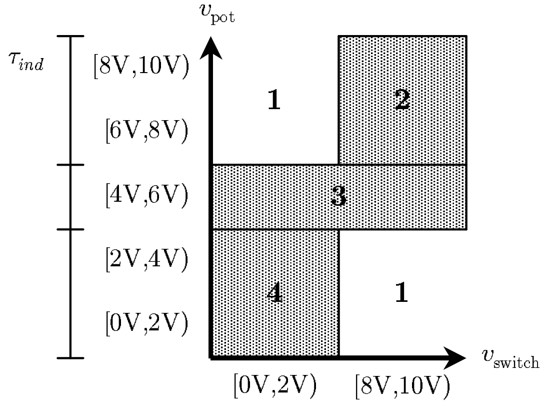

Figure 2: Partition $\Sigma(R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$ for the pedal position sensor example. The partition consists of four elements numbered 1, 2, …, 4. Applying theorem 1 yields the distinctions for $v_{pot}$ shown on the left-hand side of the figure.

**Theorem 1 (Solution to QAP)** *Let QAP be a qualitative abstraction problem that is obs-complete and sol-complete. Let $\tau_{FI,\Lambda,i}$ be the domain abstraction that aggregates the interchangeable values of a relation $\Lambda$, that is, two values $val_1, val_2 \in DOM(v_i)$ are combined if*

$$\Pi_{\boldsymbol{v}\setminus\{v_i\}}(\sigma_{v_i=val_1}(\Lambda)) = \Pi_{\boldsymbol{v}\setminus\{v_i\}}(\sigma_{v_i=val_2}(\Lambda)).$$

*Then the simplest domain abstraction that is a refinement of $\tau_{targ,i}$ and every domain abstraction*

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

*is an induced abstraction for QAP.*

For the pedal position sensor example, Theorem 1 derives three qualitative values for $v_{pot}$ (see Fig. 2):

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

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 1 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 1 (Interchangeability as QAP)** *Let $QAP = (R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$ be an obs-complete qualitative abstraction problem such that $\boldsymbol{\tau}_{obs} = \boldsymbol{\tau}_{id}$, $\boldsymbol{\tau}_{targ} = \boldsymbol{\tau}_{triv}$. Then $\tau_{FI,R,i}$ is an induced abstraction for QAP.*

In general, however, the granularity of induced abstractions is different from the granularity of interchangeable values. $\boldsymbol{\tau}_{ind}$ can be either more coarse or more fine-grained than $\boldsymbol{\tau}_{FI,R}$. The latter case occurs if target distinctions are specified between domain values that are interchangeable with respect to the model relation.

Theorem 1 constitutes also a starting point for finding useful approximations of qualitative values. One approach is to use only necessary conditions for interchangeability. In order for two domain values to be aggregated by $\boldsymbol{\tau}_{FI,\Lambda}$, it is necessary that they appear in the same elements of $\Sigma(R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$:

**Proposition 1 (Approximate Solution to QAP)** *Let QAP be a qualitative abstraction problem that is obs-complete and sol-complete. Let $\tau_{app,i}$ be the simplest domain abstraction that is a refinement of $\tau_{targ,i}$ and every domain abstraction*

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

*Then the induced abstractions are a refinement of $\tau_{app,i}$.*

The approximation $\boldsymbol{\tau}_{app}$ considers only the projection of each $\Lambda$ on the individual variables. This yields a granularity that is simpler than or equal to the induced abstractions. For the pedal position sensor example, the distinctions derived by $\tau_{app,i}$ are identical to those derived by Theorem 1. Obtaining $\tau_{app,i}$ is easier than determining $\tau_{ind,i}$, because it involves only the projection and intersection of sets and does not require to determine the interchangeable values in $\Lambda$. However, because it considers only restrictions for individual variables, the approximation $\boldsymbol{\tau}_{app}$ is not adequate. In general, a restriction might lead to a different solution only if it is combined with additional restrictions for the other variables.

## 4 AQUA: 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 $\Sigma(R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$, and determining interchangeable values within the elements of this partition.

Our prototypic system AQUA (Automated Qualitative Abstraction) ([Sachenbacher, 2001]) determines the relation $R$ through structural decomposition of the constraint network that is defined by the set of model fragments it is composed of. The basic principle of structural decomposition ([Gottlob *et al.*, 2000]) is to transform 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 ([Dechter and Pearl, 1988]), this step can be performed efficiently by local constraint propagation. This step yields the partition $\Sigma(R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$. Interchangeable values in the partition elements of $\Sigma(R, \boldsymbol{\tau}_{obs}, \boldsymbol{\tau}_{targ})$ can then be found using the basic algorithm described in [Freuder, 1991]. 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 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. Using AQUA, 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 ([Sachenbacher, 2001]).

In a larger, real-world application taken from the automotive domain [Sachenbacher *et al.*, 2000], AQUA was used to obtain abstractions of behavior models of a turbo control subsystem that involved several hundred (initially, real-valued) variables. The qualitative level of abstraction of the derived model was instrumental to meet real-time requirements, and allowed for keeping up with the rate of the measurements in an on-board diagnostic context. In this context, task-dependent abstraction proved useful as a precompilation method that guaranteed adequacy of the results.

## 5 Discussion

While several pieces of work have addressed the problem of automatically deriving appropriate models ([Nayak, 1994; Levy *et al.*, 1997; Rickel and Porter, 1997]), the work presented here is distinctive in that it focuses specifically on the granularity (resolution) of the domain values, and uses relations and partitions as a common and concise representational framework. This has several important implications.

The narrowed focus allows us to represent the space of possible candidate models implicitly and compactly as the space of possible domain partitions. In contrast, [Nayak, 1994; Rickel and Porter, 1997] define this space as the possible combinations of model fragments. Candidate models are then defined by different choices (selections) of possible fragments. Because in our framework, the difference between different candidate models is well-defined (by means of operators $\tau_i$), we can in contrast take on the view of transforming (re-formulating) models instead of selecting them. The general relational representation subsumes both infinite and finite constraints, and is not limited to specific types of constraints and special-purpose reasoning methods. It allows us to capture the conditions for a solution in a single, concise formula (Definition 3) and, more importantly, allows to determine the solutions analytically and in closed form (Theorem 1). In contrast, [Nayak, 1994; Levy *et al.*, 1997; Rickel and Porter, 1997] all devise search procedures that start from an initial model and backtrack until they find a solution. In general, it can be said that our limited scope allows us to assume a less knowledge-based, more mathematical view of automated modeling.

QSIM ([Kuipers, 1986]), 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. However, except for signs, the mapping of qualitative values to their base domain is only partially known, since only information on the ordinal relationship and knowledge about values that must be assumed simultaneously ("corresponding values") is provided. Therefore, the derived distinctions can in general not be used to simplify the model. Extensions of QSIM that deal with semi-quantitative reasoning [Berleant and Kuipers, 1997]) allow to further constrain the landmark values to numeric intervals, but are specific to the context of simulation, and the constraints are limited to algebraic relationships and monotonic functions. For the specific task of diagnosis, Torasso and Torta [Torasso and Torta, 2002] recently presented an approach for merging together behavior modes that are indistinguishable, given the granularity of the observations. However, the method does not incorporate a notion of target distinctions.

While the theory of task-dependent domain abstraction is applicable both to finite and infinite domains, an important area for future work is to efficiently derive induced abstractions for real-valued base models. [Sachenbacher, 2001] outlines a method for iterative refinement of qualitative values given a real-valued base model. [Struss, 2002] investigates cases where distinctions can be obtained for monotonic regions of real-valued functions.

## 6 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 applications, 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. Observable distinctions and target distinctions allow to capture important aspects of a task. They are the starting point for deriving qualitative values that are both adequate for the task and as simple as possible. Our analysis reveals that the degree of domain abstraction that can be achieved is strongly dependent on the characteristics of the task. Task-dependent qualitative domain abstraction is a contribution to further bridging the gap between quantitative and qualitative modeling, as it allows to express knowledge about component behavior without being committed early to a specific abstraction level of the domains. It can help to make model-based system more efficient and more cost-effective due to automating steps that are currently done by hand.

## References

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

[Dechter and Pearl, 1988] Rina Dechter and Judea Pearl. Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence*, 34(1):1–38, 1988.

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

[Freuder and Sabin, 1995] E. C. Freuder and D. Sabin. Interchangeability supports abstraction and reformulation for constraint satisfaction. In *Proceedings of SARA'95*, 1995.

[Freuder, 1991] E. C. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proc. of AAAI-91*, pages 227–233, Anaheim, CA, 1991.

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

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

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

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

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

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

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

[Sachenbacher, 2001] M. Sachenbacher. *Automated Qualitative Abstraction and its Application to Automotive Systems*. PhD thesis, TU Munich, 2001.

[Struss, 2002] P. Struss. Automated abstraction of numerical simulation models: Theory and practical experience. *Proc. QR-02, Spain*, 2002.

[Torasso and Torta, 2002] P. Torasso and G. Torta. Merging indiscriminable diagnoses: An approach based on automatic domains abstraction. *Proc. DX-02, Austria*, 2002.

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