Significant Distinctions Only:
Context-dependent Automated Qualitative Modeling

Peter Struss[1,2] and Martin Sachenbacher[1,3]

[1]Technical Univ. of Munich
Dept. of Computer Science
Orleansstr. 34
D-81667 Munich
Germany

[2]OCC'M Software GmbH
Gleissentalstr. 22
D-82041 Deisenhofen
Germany

[3]Robert Bosch GmbH
Department FV/SLN
P.O. Box 30 02 40
D-70442 Stuttgart
Germany

{struss, sachenba}@in.tum.de, struss@occm.de
http://www9.in.tum.de/MQM/

Abstract

Qualitative modeling means making the essential distinctions only. Compositional modeling requires to state behavior models of system constituents (e.g components) independently of their context. This creates a problem, because what is essential depends not only on the local model fragment, but also on the context of the model and its usage, i.e. the structure of the entire system and the task to be performed. For instance, in diagnosis the goal of discriminating between different behavior modes determines the distinctions to be made. The paper deals with the problem of deriving the sets of qualitative values of model variables that allow to generate the distinctions required by the goal of model based prediction and the structure of the system. We present a formal definition and analysis of the problem and an algorithm for computing appropriate qualitative values based on propagation of distinctions. An important special case is the computation of local landmarks of variables. Based on the generic solution, we show how models for diagnostic can be derived.

Introduction

The success of model-based systems and the growing interest in their industrial application strongly relies on the availability of models that are both compositional and powerful enough to support solving different problems. Compositionality, i.e. the possibility to combine models taken from a library to form a system model, requires **genericity** and addresses the efficiency of the **model formation process**. Models with a granularity and sufficient inferential power for a particular problem have to **be specific** and address effectiveness and efficiency of the **problem solving process**.

Obviously, there is a tension between these requirements. Qualitative modeling, besides the feature of providing facilities for conceptual modeling, can be seen as pursuing the goal of making significant distinctions only in a model. However, what is significant depends, on the context and the task. Finding a solution to this dilemma is crucial, because otherwise model libraries with re-usable models are not feasible, and this will seriously reduce the practical value of model-based systems.

The paper presents a contribution to analyzing and solving the task of generating a model that reflects the distinctions required by individual model fragments and the context and purpose of the model. We first illustrate the problem by a tiny, but practical example. Section 3 defines the goal formally and precisely. Then we characterize the solution in a theoretical way. In section 5, we present the foundations and an outline of an algorithm that computes the solution. Finally, we attempt to assess the impact of the results achieved so far and point out open questions and future work.

An Introductory Example

The following example is meant to illustrate the problem and, in the following, the solutions presented. Although we simplified it for this purpose, it is drawn from a real application involving diagnosis and fault analysis of vehicles.

The device in Figure 1 shows a pedal position sensor 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. This happens in two ways: via the potentiometer as an analogue signal, v_{pot}, and via the idle switch as a binary one, v_{switch}. The idle switch changes its state at a particular value of the mechanically transferred pedal position (this is a simplification, since the transition happens in some "fuzzy" interval). This system is fairly simply structured (nevertheless, the manufacturer's failure mode and effects analysis (FMEA) for this system covers 6 pages), and its electrical subsystem comprises only standard components. We do not expect any problems finding appropriate behavior models for them in our model library.

Actually, there **can** be a problem, dependent on the type of the model. If we have chosen to use qualitative models, which distinguish only between voltage "gnd", "between" and "batt" as convenient for many tasks, the models of the main components, potentiometer and switch may be the ones shown in Table 1 (the depicted case $i_{out}=0$ applies to the correctly working system of Fig. 1) and Table 2, respectively. They may suffice for some purposes. However, they are of limited use when we want to use the model for instance for diagnosis, FMEA, or to support the development of control unit software. The reason is that these tasks have to exploit the redundancy which purposefully has been implemented in the system: the two signals entering the ECU can be checked for plausibility. The two possible values of v_{switch}, due to the switch state and, hence, the pedal position, correspond to two ranges of v_{pot} separated by a particular voltage value. However, this "landmark" is missing in the domain of voltage, and, hence, the compositional model will be of very limited utility for such tasks. How could the modeler of the potentiometer anticipate this particular context? Actually, he could not, and, moreover, he **should not**, because the voltage landmark would not make any sense in a different structure.

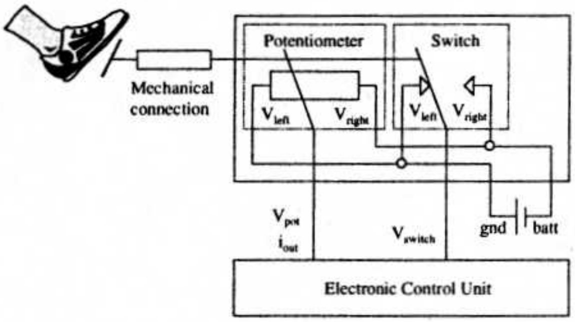

Figure 1: The Pedal Position Sensor

Of course, we could use numerical models in our library which would be able to relate the switching position to a particular division of the potentiometer voltage and v_{pot}. However, this model would be overly detailed for the purpose discussed. What we would like to have, is a composition of component models that make just the right distinctions required by other components and the task the model is used for. Since we cannot expect to find such models in the library right away, the only way out is to **generate** it from a base model which may be in a library. If we cannot automate this goal-directed transformation of models, much of the benefit of model-based systems, namely re-use of model fragments, cannot be achieved, and this will limit the utility of the technology drastically.

The example should trigger some intuitions of how this might be achieved. Starting from a more detailed model, the requirements for certain distinctions arising from some components and/or the particular task can induce significant distinctions in other components via the structure of the device and the individual component models. For instance, the required distinction between "gnd" and "batt" for v_{switch} requires the distinction between the switch states which in turn determines a distinctive position of the pedal, and this induces a landmark in the domain of v_{pot}.

Pos_{pot}	i_{out}	v_{left}	v_{right}	v_{pot}	...
0	0	{gnd, betw, batt}	{gnd, betw, batt}	$= v_{left}$	...
$pos_{p,max}$	0	{gnd, betw, batt}	{gnd, betw, batt}	$= v_{right}$	...
(0, $pos_{p,max}$)	0	gnd	gnd	gnd	...
(0, $pos_{p,max}$)	0	{gnd, betw}	{betw, batt}	betw	...
(0, $pos_{p,max}$)	0	batt	{gnd, betw}	betw	...
(0, $pos_{p,max}$)	0	batt	batt	batt	...
...	...	...	...	...	...

Table 1: Partial potentiometer behavior model from the library

pos_{switch}	state	v_{left}	v_{right}	v_{switch}
[0, $pos_{s,med}$]	left	{gnd, betw, batt}	{gnd, betw, batt}	$= v_{left}$
($pos_{s,med}$, $pos_{s,max}$]	right	{gnd, betw, batt}	{gnd, betw, batt}	$= v_{right}$

Table 2: Switch behavior model from the library ($pos_{s,med}$ is the switching position)

We would like to turn this intuition into an algorithm and, in order to to create its foundations, analyze the task in a rigorous and formal way.

Formalizing the Goal

The first step is to state our goal precisely and formally. We will define it in a fairly general way. In particular, we do not only treat models composed of continuous functions, but relational models, as motivated by the switch example in the previous section. Accordingly, a model of a system, S, to be analyzed is given by a relation

$$R_S \subset DOM(\underline{v}_S) ,$$

where $\underline{v}_S$ is the vector of all parameters and variables (input, output, internal and state variables) in the system.

In this formalism, our problem can be stated as follows: there is

- a base domain $DOM_0(v_i)$ for each variable v_i, (e.g. real numbers, intervals reflecting precision, but also states or a qualitative domain) and a model

$$R_{S,0} \subset DOM_0(\underline{v}_S) = DOM_0(v_1) \times DOM_0(v_2) \times ... \times DOM_0(v_n),$$

- a characterization of the primary distinctions for each v_i required for some external reason (a functional specification, safety limits, diagnostic distinctions etc.) or due to the structure of the local model, expressed in

$DOM_0(v_i)$ (e.g. the switch output voltage being zero or positive). More precisely, for a variable v, such distinctions are specified as partitions $\Pi=\{P_k\}$ of $DOM_0(v)$ [1]. A partition $\{P_k\} \subset P(DOM_0(v))$ (the power set of the domain) is a set of non-empty disjoint subsets that together cover the entire domain:

$\forall P_k : P_k \neq \varnothing$

$P_k \cap P_l \neq \varnothing \Rightarrow P_k = P_l$ and

$\bigcup P_k = DOM_0(v)$.

The intuition behind the partitions is that they define **qualitative values**: exactly values in different partitions P_k have to be distinguished from each other. In our example, the output voltage of the switch has the primary partition $\Pi_{vswitch}=\{\{0\}, (0, \infty)\}$. For many parameters and internal variables, there are no primary distinctions to be made. In this case, the partition is the trivial one:

$\Pi=\{DOM_0(v)\}$.

In the following, it is often convenient to talk about the mapping of values to qualitative values, which we call qualitative domain abstraction.

Definition (Qualitative Domain Abstraction)
A qualitative domain abstraction is a mapping

$\tau\colon DOM_0(v) \rightarrow DOM_\alpha(v) \subset P(DOM_0(v))$
where $\forall\, v_0 \in DOM_0(v)$: $v_0 \in \tau(v_0)$.

Remark: A qualitative domain abstraction is a domain abstraction in the sense of (Struss 92) and induces an abstraction of the system model by

$R_{S,\tau} := \tau(R_{S,0})$.

There is an obvious correspondence between domain abstractions and partitions: a qualitative domain abstraction τ of $DOM_0(v)$ induces a partition

$\Pi_\tau = \tau(DOM_0(v))$,

and vice versa. By π, we denote the qualitative domain abstraction induced by the primary partition:

$\pi := (\pi_1, \pi_2, \dots \pi_n) : DOM_0(\underline{v}_S) \rightarrow \Pi_1 \times \Pi_2 \times \dots \times \Pi_n$.

Variables that do not have any primary distinction associated, are mapped to the trivial partition, i.e. there exists only one „qualitative value" which represents the entire domain. Our view is that all we are ultimately interested in when using the model is optimal information about the primary distinctions, and that other distinctions should be considered if and only if they are necessary to derive conclusions about the primary ones. From the initial fine-grained model $R_{S,0}$, primary distinctions can be determined by applying π.

What does it mean to „use the model"? It means, given information on some parameters or variables (through measurements, design choices, etc.), to determine resulting restrictions on other parameters and variables. If, for instance, measurements MEAS for some variables with the granularity of the respective DOM_0 are given, we can compute the resulting restriction $R_{S,0} \cap$ MEAS. But only the primary distinctions implied by this restriction matter, i.e.

$\pi(R_{S,0} \cap MEAS)$.

$R_{S,0}$ may not be able to determine all required distinctions. But w.r.t. the possible ones, $DOM_0(\underline{v}_S)$ may be overly detailed. We would like to determine the distinctions to be made for each v_i that are both necessary and sufficient in order to express the model in terms of these distinctions only without losing the "distinguishing power" of $DOM_0(\underline{v}_S)$. This means: finding a qualitative domain abstraction for $DOM_0(\underline{v}_S)$

$\tau=(\tau_1, \tau_2, \dots, \tau_n)$
where $\tau_i\colon DOM_0(v_i) \rightarrow DOM_\alpha(v_i) = P(DOM_0(v_i))$

which is maximal in some sense but does not destroy the primary distinctions. For instance, if the pedal position is given, then distinguishing values $[0, pos_{s,med}]$ from those in $(pos_{s,med}\ ,\ pos_{s,max}]$ is necessary and sufficient to derive the primary distinctions for v_{switch} (zero vs. non.zero). In general terms, the requirement means: if there is any external restriction on the system behavior (actual observations, design specification, etc.), applying the qualitative domain abstraction τ before determining the primary distinctions does not change the result, formally: if the external restriction is given by a relation $R_{ext} \subset DOM_0(\underline{v}_S)$, then

$\pi'(\tau(R_{ext}) \cap \tau(R_{S,0})) = \pi(R_{ext} \cap R_{S,0})$. (1)

Here, $\pi'\colon \tau(DOM_0(v_S)) \rightarrow \Pi_1 \times \Pi_2 \times \dots \times \Pi_n$ maps the results of the qualitative domain abstraction τ (i.e. sets) to the primary partitions they are contained in:

$\pi'(\tau(v)) = \pi(v))$ for $v \in V$.

Obviously, this is well-defined only if τ is a refinement of π according to the following definition.

Definition (Refinement and Merge of Partitions and Domain Abstractions)
Let Π_1, Π_2 in $DOM_0(v)$ be two partitions. Π_1 is called a refinement of another one, Π_2, iff

$\forall\, P_1 \in \Pi_1\ \exists\, P_2 \in \Pi_2\ \ P_1 \subseteq P_2$.

It is called a strict refinement, if, additionally,

$\exists\, P_1 \in \Pi_1\ \ \forall\, P_2 \in \Pi_2\ \ P_1 \neq P_2$.

The merge of two partitions Π_1, Π_2 of $DOM_0(v)$ is the partition containing all intersections of their elements:

$merge(\Pi_1,\Pi_2) := \{\, P_1 \cap P_2 \mid P_1 \in \Pi_1 \wedge P_2 \in \Pi_2 \,\} \backslash \{\varnothing\}$.

We apply the same terminology to the qualitative domain abstractions induced by the partitions.

Property (1) guarantees that we can first abstract both the model and the measurements and still are able to detect the same primary distinctions as before:

$\pi'(\tau(R_{S,0}) \cap \tau(MEAS)) = \pi(R_{S,0} \cap MEAS)$.

[1] In the following, when considering an arbitrary variable, we drop the index to improve readibility.

$$\begin{tikzcd}
R_{ext}\ R_{S,0} \arrow[r, "\tau"] \arrow[d, "\cap"] & \tau(R_{ext})\ \tau(R_{S,0}) \arrow[d, "\cap"] \\
R_{ext} \cap R_{S,0} \arrow[d, "\pi"] & \tau(R_{ext}) \cap \tau(R_{S,0}) \arrow[d, "\pi'"] \\
\pi(R_{ext} \cap R_{S,0}) \arrow[r, phantom, "="] & \pi'(\tau(R_{ext}) \cap \tau(R_{S,0}))
\end{tikzcd}$$

Figure 2: Relationship of primary and induced qualitative domain abstractions

Figure 2 illustrates the situation. This analysis justifies the following definition of our target:

Definition (Distinguishing Qualitative Domain Abstraction)

Let $R_{S,0}$ in $DOM_0(\underline{v}_S)$ be the original fine-grained model of a system S and, for each variable v_i, a finite set of primary distinctions be given as a partition $\Pi_i=\{P_{ik}\}$ of $DOM_0(v_i)$.

A qualitative domain abstraction

$$\tau\colon DOM_0(\underline{v}_S) \rightarrow P(DOM_0(\underline{v}_S))$$

is distinguishing w.r.t. $\{\Pi_i\}$ iff it is a refinement of π and

$$\forall R_{ext} \subset DOM_0(\underline{v})$$
$$\pi'(\tau(R_{ext}) \cap \tau(R_{S,0})) = \pi(R_{ext} \cap R_{S,0})\ . \quad (1)$$

A distinguishing domain abstraction τ is **maximal**, if there is no distinguishing qualitative domain abstraction τ' that makes less distinctions w.r.t. one $DOM_0(v_i)$, i.e. there is no τ_i that is a strict refinement of any τ'_i .

An important and common specialization of distinctions and of the task of finding maximal qualitative domain abstractions is obtained if qualitative values are given as intervals of ordered domains of variables. In this case, we can represent qualitative values in a compact way by their boundaries, the „landmarks" as opposed to an extensional representation of sets. For instance, the partition of $DOM(pos_{switch})$ can be represented by the landmark $pos_{s,mod}$.

Definition (Landmark partition)

Let $DOM_0(v)$ be a totally ordered domain for a variable v. For a landmark set

$$L=\{l_k\} \subset DOM_0(v) \text{ with } k<m \Rightarrow l_k<l_m\ ,$$

the induced partition

$$\Pi_L=\{\ \{l_k\}\ \} \cup \{(l_k, l_{k+1})\}$$

is called a landmark partition.

In this case, we can hope for a compact representation of partitions, and, if there are (piecewise) monotonic functional dependencies among variables, also an easier way of computing maximal distinguishing qualitative domain abstractions.

Now that we have defined our goal, i.e. maximal distinguishing domain abstractions, we will characterize the desired solution in a formal way.

Characterizing Maximal Distinguishing Abstractions

The intuition behind the formal characterization of the desired qualitative domain abstractions is that property (1) can be established if the qualitative domain abstraction τ_j of **each single** $DOM_0(v_j)$ reflects the primary distinctions π_i of **any other** $DOM_0(v_i)$ (of course, including its own). This means, we apply τ to one $DOM_0(v_j)$ at a time only (leaving the other variables at the granularity of DOM_0) which corresponds to the mapping

$$(\ id_1, ..., id_{j-1}, \tau_j, id_{j+1}, ..., id_n)\ ,$$

where id_k is the identical mapping on $DOM_0(v_k)$. Then we determine the primary distinctions by the mapping

$$\pi''_j = (\pi_1, ...\ \pi_{j-1}, \pi'_j, \pi_{j+1}, ..., \pi_n)\ ,$$

and (1) implies

$$\forall\ i,j\quad pr_i(\pi''_j\ (\tau''_j(R_{ext}) \cap \tau''_j(R_{S,0}))$$
$$= pr_i(\pi(R_{ext} \cap R_{S,0})), \qquad (2)$$

where pr_i is the projection to the i-th variable. On the other hand, if (2) holds, then (1) can be proved. This motivates a characterization of distinguishing qualitative domain abstractions starting from the question: *Which distinctions in $DOM_0(v_j)$ are necessary in order to guarantee the determination of the primary distinctions in $DOM_0(v_i)$ under the assumption that all other variables can make distinctions given by DOM_0?*

Given the primary distinctions for some $DOM_0(v_i)$, we have to determine which values in $DOM_0(v_j)$ can be aggregated into one qualitative value. The answer is that we can aggregate two values $v_{j,1}$, $v_{j,2}$ in $DOM_0(v_j)$ if they always lead to the same conclusions for the primary distinctions of $DOM_0(v_i)$, regardless of any additional restriction on other variables. This idea is captured by the following equivalence relation on $DOM_0(v_j)$.

Definition (Induced Partition)

Let an equivalence relation on $DOM_0(v_j)$ be defined by

$$v_{j,1} \approx_j v_{j,2} :\Leftrightarrow \forall\ i,k\quad pr_{-j}(\{v_j = v_{j,1}\} \otimes R_{S,0} \otimes \{v_i = P_{ik}\})$$
$$= pr_{-j}(\ \{\ v_j = v_{j,2}\} \otimes R_{S,0} \otimes \{v_i = P_{ik}\})\ , \quad (3)$$

where pr_{-j} denotes the projection that eliminates the j-th variable, and for two relations R_1, R_2, the join $R_1 \otimes R_2$ is the intersection of the relations after their embedding into the domain of all occurring variables.

The sets of partitions $\Pi_{ind,j}$ for $DOM_0(v_j)$ given by the equivalence classes of the relations $\approx_j$,

$$\Pi_{ind,j} := DOM_0(v_j)\,|_{\approx j}$$

are called the partitions induced by the primary distinctions.

This means: two values are in different equivalence classes if and only if they entail different conclusions for at least one P_{ik} (i.e. refuting it or not), possibly together with additional information on other variables. This leads to the following characterization.

Lemma (Characterization of Maximal Distinctive Domain Aggregations)
The set of induced partitions { $\Pi_{ind,j}$ } defines a maximal distinguishing qualitative domain abstraction τ_{ind} w.r.t. to the primary distinctions.

Example (Pedal Position Sensor Switch)
Considering our introductory example and assuming

$$DOM_0(v_{left})=DOM_0(v_{right})=DOM_0(v_{switch})=\mathbb{R}_0^+$$

and

$$DOM_0(pos_{switch}) = [0,pos_{s,max}] \subset \mathbb{R}$$

for the variables of the switch, we can express the task requirement of distinguishing at v_{switch} between 0 (= "gnd") and positive values as an primary partition for v_{switch}: $\Pi v_{switch} = \{\{0\},\mathbb{R}^+\}$.

For all other variables (including the switch state), we do not assume any primary distinction. When applying (3), we have to consider $R_{switch,0} \otimes \{v_{switch}=0\}$ and $R_{switch,0} \otimes \{v_{switch} \in \mathbb{R}^+\}$ displayed in the tables below:

$R_{switch,0} \otimes \{v_{switch} = 0\}$:

pos_{switch}	state	v_{switch}	v_{left}	v_{right}
$[0, pos_{s,med}]$	left	0	0	$\mathbb{R}_0^+$
$(pos_{s,med}, pos_{s,max}]$	right	0	$\mathbb{R}_0^+$	0

$R_{switch,0} \otimes \{v_{switch} \in \mathbb{R}^+\}$:

pos_{switch}	state	v_{switch}	v_{left}	v_{right}
$[0, pos_{s,med}]$	left	$\mathbb{R}^+$	$\mathbb{R}^+$	$\mathbb{R}_0^+$
$(pos_{s,med}, pos_{s,max}]$	right	$\mathbb{R}^+$	$\mathbb{R}_0^+$	$\mathbb{R}^+$

To compute the induced partition for, say, v_{right}, the projections defined by (3) have to be compared for each value of v_{right}. As can be easily seen, all positive values for v_{right} occur in the same tuples both for $v_{switch} = 0$ and $v_{switch} \in \mathbb{R}^+$ and hence, form one partition, whereas $v_{right} = 0$ occurs with state = right for $v_{switch} = 0$ in contrast to all positive values. Hence, the lemma successfully induces the partition $\Pi v_{right} = \{\{0\}, \mathbb{R}^+\}$ as we would expect. Please note, that this happened only because we looked at the switch in isolation. If we consider the entire pedal position sensor and, hence, the connection to the battery as in Figure 1, v_{right} would be constrained to "batt" and not receive the landmark.

We point out, that (3) also induces a partition for the state, $\Pi_{state} = \{\{left\},\{right\}\}$. This seems obvious. However, if the switch would be seen as part of a different structure, e.g. connected to the battery as displayed in Figure 3, then we obtain $R_{switch',0} \otimes \{v_{switch} = 0\}$ as

pos_{switch}	state	v_{left}	v_{right}	v_{switch}
$[0, pos_{s,max}]$	{left,right}	0	0	0

and $R_{switch',0} \otimes \mathbb{R}^+ = \emptyset$, and there is no reason to distinguish the two states (and no values of other variables). This illustrates that the induction of partitions can also be applied to discrete variables and may detect that it is useless to distinguish between certain states.

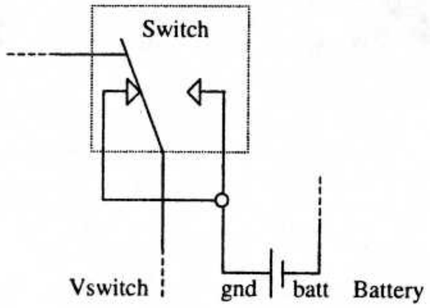

Figure 3: Switch with modified structure

Example (Multiplication Constraint)
As a second example, we consider the constraint c=a*b. Let us assume first that

$$DOM_0(a)= DOM_0(b)= DOM_0(c)= \mathbb{R}$$

and that there is an primary partition on $DOM_0(c)$ given by the landmark set $L_c=\{0, 1\}$ for c and none for a and b. If we try to construct a maximal abstraction τ according to (3), we have to realize that, except for τ=id, there is no abstraction that has the discriminating power of the real numbers. In fact, if a qualitative abstraction τ would map two different real numbers, $a_1<a_2$, onto one partition, then choosing

$$R_{ext}=\{a=a_1\} \otimes \{b=1/a_1\}$$

reveals the loss: over the real numbers, the resulting abstraction π yields

$$\pi(R_{ext} \cap R_{S,0}) = (\mathbb{R}, \mathbb{R}, \{1\}),$$

whereas π' produces

$$\pi'(\tau(R_{ext}) \cap \tau(R_{S,0})) = (\mathbb{R}, \mathbb{R}, [1, \infty)).$$

In other words, we cannot maintain the precision of $\mathbb{R}$ when we give up $\mathbb{R}$. However, if we change the granularity of the domain (e.g. because we cannot measure real numbers, anyway) and choose intervals bounded by integers,

$$DOM_0(a)= DOM_0(b)= DOM_0(c)=\{ (z, z+1) \mid z \in INT \},$$

then the picture changes. As suggested by Figure 4, we can summarize all values greater than 1 for a. Intuitively speaking, it would not pay off to distinguish between them because b's values are not fine-grained enough to exploit the distinction, say, for determining whether or not c is less than, equal to, or greater than 1. Applying (3) induces the landmarks

$$L_a= L_b=\{-1, 0, 1\}.$$

As a lesson to be learnt here, the choice of the initial representation (the „gold standard") can be crucial for the result obtained.

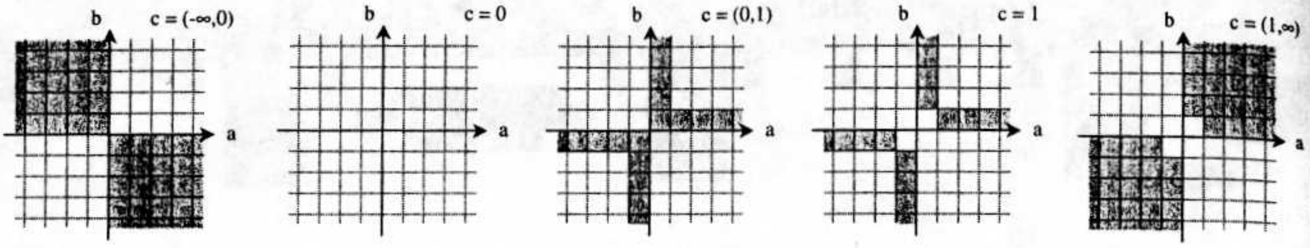

Figure 4: Projections of a*b=c to the (a,b)-plain for the qualitative values of c

What makes the characterization according to (3) not very handy for the actual computation of the qualitative domain abstraction, is

1. that checking (3) involves comparison of any pair of values in the fine-grained representation of DOM_0, and
2. that computation of $\{v_{j,1}\} \otimes R_{S,0} \otimes P_{ik}$ can be costly for complex devices.

Regarding 1, we cannot do much about it without losing distinguishing power, except for the case of landmark partitions and continuous functions. A weaker equivalence relation would be

$$v_{j,1} \approx'_j v_{j,2} :\Leftrightarrow$$
$$\forall\, i,k \quad v_{j,1} \in pr_j(R_{S,0} \otimes P_{ik}) \Leftrightarrow v_{j,2} \in pr_j(R_{S,0} \otimes P_{ik}). \; (3')$$

The resulting qualitative domain abstraction joins the values if and only if the same conclusions for the primary distinctions can be derived from them alone, i.e. without regarding additional information on other variables. (3) is interesting because it does not require mutual comparison of values, but computation of projections and their intersection.

As for the second computational problem, we have several observations: the computation of the $R_{S,0} \otimes P_{ik}$ can be done once for the checks on all $DOM_0(v_j)$. More importantly, we should consider **how** this computation can be achieved. Unless the system is highly connected, the interaction between v_j and the distinctions in $DOM_0(v_i)$ is not direct, but mediated by a sequence of other variables. This leads to the idea of exploiting the structure of the system and to propagate distinctions through it to compute the possible qualitative domain abstractions in different parts of the system. In other words, while we usually compute distinctions from the model by propagating given external restrictions, we would like to start from the necessary primary distinctions and propagate them "backward" in order to determine the granularity that provides optimal results when propagated "forward".

Foundations of the Propagation Algorithm

In this section, we provide the formal foundations for the iterative computation of the $\Pi_{ind,j}$ by propagation of distinctions. There are two fundamental ideas that have to be formalized and proved: For each variable v_j,

1. we can compute the final partition $\Pi_{ind,j}$ by computing the partitions induced by the primary partitions Π_i of the other variables v_i individually and merging them.
2. Rather than carrying out the checks involved in definitions (3) or (3') globally, i.e. w.r.t. the entire $R_{S,0} \otimes P_{ik}$, the algorithm propagates the qualitative values P_{ik} to generate induced partitions in variables adjacent to v_j and then compute the equivalence classes of v_j w.r.t. these neighbor partitions only.

The first step is straightforward, since it means checking the condition for the equivalence for a growing set of indices (i,k) which can only lead to further splitting of equivalence classes and never to joining them. Formally:

Lemma: (Iterative computation of induced partitions)
Let IK = {(i,k)} be the entire index set of the primary partitions $\Pi_i = \{P_{ik}\}$ and $IK'' \subset IK' \subseteq IK$. Then $\Pi_{ind,j}(IK')$ is a refinement of $\Pi_{ind,j}(IK'')$ and more specifically,

$$\Pi_{ind,j}(IK') = \text{merge } \Pi_{ind,j}(IK''), \Pi_{ind,j}(IK' \setminus IK''))$$

Here, $\Pi_{ind,j}(IK')$ is obtained by applying (3) or (3'), respectively, quantifying over $(i,k) \in IK'$.

To formalize the second idea, we select two different variables v_i, v_j and analyze how to compute the partitions of v_j induced by v_i (Figure 5a). v_j can only receive induced partitions via constraints it is directly involved in. Hence, we split the entire relation $R_{S,0}$ for each variable v_j into two relations $R_{S,0} = R_{j,0} \otimes R_{-j,0}$, where $R_{j,0}$ comprises all constraints that involve v_j directly (i.e. restrict it) and $R_{-j,0}$ the other ones. In terms of component-oriented modeling this means:

- For internal variables and parameters v_j, the respective $R_{j,0}$ is basically the component model relation (possibly without constraints that do not mention v_j)
- For the interface (terminal) variables shared by the components, $R_{j,0}$ is constituted by the constraints in the two connected components that refer to v_j.

This replaces $R_{S,0} \otimes P_{ik}$ by $R_{j,0} \otimes R_{-j,0} \otimes P_{ik}$ in (3). In Figure 5b, the ellipse separates $R_{j,0}$ (in its interior) from $R_{-j,0}$. Since v_j does not directly interact with $R_{-j,0}$, it reflects the partitions P_{ik} by reflecting the partitions induced by P_{ik} on the "neighbor variables" of v_j, i.e. all variables occurring in $R_{j,0}$ except for v_j. This means the induced distinctions for v_j are obtained in two steps: First, for all neighbor variables v_m of v_j, the partitions P_{ml} induced on them by v_i are computed. Then the partitions on v_j are computed within the ellipse, i.e. by confining (3) to the neighborhood of v_j,

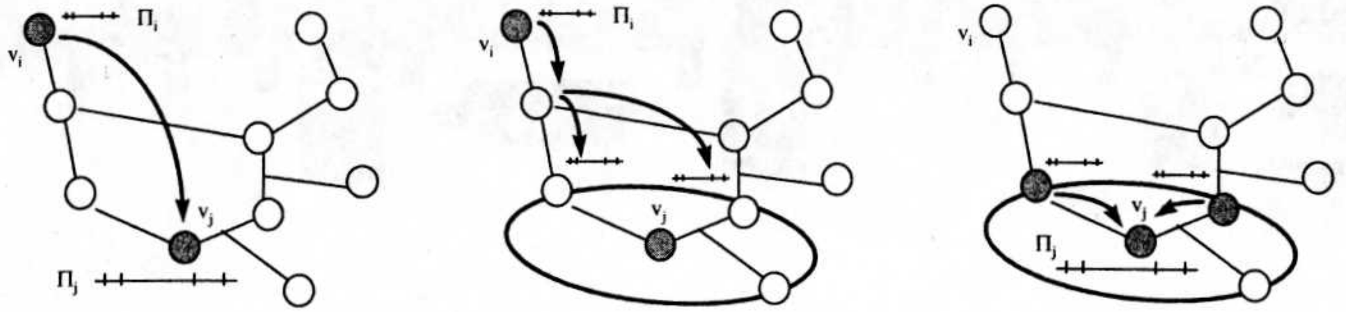

Figure 5 a, b, c: Localizing the computation of partitions

i.e. to $R_{j,0} \otimes P_{ml}$ (Figure 5c). The following lemma states that this correctly determines the partitions for v_j.

Lemma (Propagation of Partitions)
Let $IK' \subset IK$, where for $(i,k) \in IK'$, v_i does not occur in $R_{j,0}$, i.e. is not a neighboring variable of v_j. Let further ML be an index set, such that for all $(m,l) \in ML$, v_m is a neighbor variable of v_j and ML is given by the partitions induced by P_{ik}:

$$\{P_{ml} \mid (m,l) \in ML\} = \Pi_{ind,m}(R_{-j,0}, IK').$$

Then

$$\Pi_{ind,j}(R_{S,0}, IK') = \Pi_{ind,j}(R_{j,0}, ML).$$

Here $\Pi_{ind,j}(R_x, \cdot\ )$ means that (3) are carried out for R_x replacing $R_{S,0}$.

This provides the foundation for an algorithm (or, rather, a family of algorithms) that computes induced qualitative values from the primary distinctions by propagation through the structure of the model, as illustrated by Figure 5 a, b, c. It can be sketched as follows (Figure 6):

```
Algorithm ComputeInducedPartitions(variable list)
1. For all variables $v_i$ :
     current partition($v_i$) $\leftarrow$ primary partition($v_i$)
2. ComputeNeighborPartitions(variable list)
3. For all variables $v_i$ :
     first partition($v_i$) $\leftarrow$ current partition($v_i$)
     current partition($v_i$) $\leftarrow$ primary partition($v_i$)
4. ComputeNeighborPartitions (reverse(variable list))
5. For all variables $v_i$ :
     final partition($v_i$) $\leftarrow$ merge(first partition($v_i$), current
     partition($v_i$))
```

Figure 6: Algorithm for the computation of induced partitions

The procedure ComputeNeighborPartitions gets the first variable on the variable list and its current partition, computes the induced partitions for all neighboring variables in the rest of the list, then merges them with the current partitions and recursively calls itself on the rest of the variable list.

After the first call of the procedure, every variable in the variable list has received induced partitions from all predecessors in the list, and in the second run from all successors.

For our example of the pedal position sensor, the propagation algorithm would induce the landmark $pos_{s,med}$ for the switch which maps to $pos_{ped,med}$ for the pedal and $pos_{pot,med}$ for the potentiometer, finally generating a landmark $v_{pot,med}$ corresponding to this position, as needed for the task described.

Application: Models for Diagnosis and Significant Deviations

In many cases, for instance in diagnosis and FMEA, what makes a distinction in a model of a component is not determined by some absolute values of other variables, but by the fact whether or not it enforces a significant **deviation** on them, regardless of what their specific value is. The view is here that the function of the overall device imposes a certain tolerance on the output of this device, and its components are not considered faulty unless their behavior causes a disturbance of the output beyond the given tolerance. If we succeed to compute the tolerances of the parameters of the component models starting from the given functional specification, we can automatically generate fault models that reflect the particular device and its context, which we cannot expect by definition from generic models.

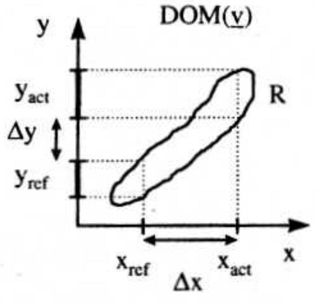

Figure 7: A relational model can impose constraints on the deviations of variables

The idea underlying deviation models is to describe deviations of variables which are consistent with a certain behavior model (Figure 7). For this purpose, we define

$$\underline{\Delta} : \mathrm{DOM}_0(v) \times \mathrm{DOM}_0(v) \rightarrow \mathrm{DOM}_0(v)$$

$$\underline{\Delta}((x_1, x_2, ..., x_n), (y_1, y_2, ..., y_n)) :=$$
$$(\Delta_1(x_1,y_1), \Delta_2(x_2,y_2), .., \Delta_k(x_n,y_n)) = (x_1\text{-}y_1, x_2\text{-}y_2, .., x_n\text{-}y_n)$$

The deviation model of R, denoted R_Δ, describes how deviations of variables propagate through a component:

$$R_\Delta \subseteq \mathrm{DOM}_0(\underline{v}) \times \mathrm{DOM}_0(\underline{v}) \times \mathrm{DOM}_0(\underline{v})$$

$$R_\Delta := \{(\underline{v}, \underline{v}', \underline{\Delta}(\underline{v}, \underline{v}') \mid \underline{v}, \underline{v}' \in R \}$$

The projection of R_Δ on the Δv_i, $pr_{\Delta vi}(R_\Delta)$, can be viewed as a "pure" deviation model which relates deviations of variables independent of their actual and reference value. This is meaningful only in some cases, for example, if the relation describes a monotonic function. In general, at least information about the actual value will be necessary. The analysis of significant distinctions developed above can be applied to such deviation models: we can specify what is considered to be a significant deviation of some relevant variables by landmarks of the respective Δ-variable. They will induce partitions for other respective Δ-variables, but can also propagate to the domains of the variables themselves.

The induced landmarks in the model can be used to properly model the (context-dependent) limits of correct behavior and to define fault models. They also indicate some requirements on the precision of observations for diagnostic purposes.

Basically, the technique determines the **model granularity** (and observation granularity) given particular **functional distinctions**. The result will mainly serve fault detection. Fault localization and identification might be possible, but is not guaranteed.

For the latter, the goal is to determine model and observation granularity given particular **behavior mode distinctions.**

The question is here: *Given a set of behavior modes for each component (one correct mode and several faulty ones), what distinctions have to be made in the models (and the observations) to help discriminate the modes?*

One way to address this problem is to represent the component models as follows: For component c, a **mode variable** m_c is introduced whose values represent the various, mutually exclusive behavior modes $m_{c,i}$ of the component. Each mode has an associated model relation, $R_{c,i}$. This establishes the component model as

$$\wedge_i\ (\mathrm{mode}_c = m_{c,i} \Rightarrow \mathrm{model}(R_{c,i})),$$

or, in relational form,

$$R_c = \cup_i\ \{m_{c,i}\} \times R_{c,i}.$$

The system model relation is then the join of all these component models. For this, we can now formulate the task in our formalism: for each component, we define the primary partition of its mode variable to be the total partition:

$$\Pi\mathrm{mode}_{c,i} = \{\{m_{c,i}\}\},$$

and compute the induced qualitative values. Basically, this means treating the behavior mode like a state variable whose values have to be completely distinguished form each other.

Summary and Discussion

The work presented here is an attempt to make progress in understanding and solving the crucial question in qualitative reasoning: *How to generate models that make the essential distinctions only?*

Pursuing the goal of automated generation of such models aims at coping with the tension between the genericity and compositionality of models and their specificity and power for a particular context and task. We feel that ignoring this problem could lead to a failure or, at least, limitations in the application of model-based systems, because re-usability of model fragments and model libraries would be low.

What have we achieved by now? There are two outcomes that both should be appreciated: on the one hand we started to provide a formal, theoretical foundation for formulating and analyzing the goal, the problems and solutions in a rigorous way. This is important, because we have to be aware that a huge part of the problem space is not tractable in theoretical and/or practical sense. We better analyze and characterize what we can expect to solve in practice. After all, this work connects to the analysis in (Struss 88), (Struss 90) where we showed that trying to make all significant distinctions in qualitative models can lead to infinite domains (like the rational numbers). Here, we do attempt to create models that make all significant distinctions, and one goal of future work is to identify conditions under which this can work. We believe that the chosen representation, relational models, is general enough as a basis.

The second result is the design of an algorithm (or, rather, a family of algorithms) that actually computes the significant distinctions making use of the particular system structure. Of course, its applicability and feasibility is threatened by the same problems as its theoretical foundation. There are different dimensions of these problems and potential solutions:

- The computation of the equivalence classes according to (3) and (3') may simply lead to infinite distinctions or be practically impossible. As we illustrated with an example, the granularity of the base model plays an important role here. If a type of constraint is known to provide no qualitative values if only one variable requires distinctions, the respective method could just skip the computation or replace it by a weaker one which, for instance, combines landmarks of several variables to generate others.
- Intuitively, the special case where the relations represent functional relationships among variables seems more restricted. However, there are several caveats: dependent on the granularity for the domain DOM_0, the abstraction of a real-valued function may no longer be a function, and a landmark does not necessarily induce a single value elsewhere. On the

other hand, multi-variate real-valued functions may fail to induce a finite set of landmarks, as the multiplication example showed. What will work smoothly are simple (piecewise) monotonic functions where landmarks map to landmarks and intervals to intervals.

- Right now, the algorithm propagates primary distinctions everywhere. There is simply no criterion where they might ultimately be needed. We were not very specific about where primary distinctions come from and what they mean. Also, the definition of distinguishing domain abstractions, criterion (3), and the algorithm contain or exploit no restriction on the nature or structure of the external restrictions, R_{ext}. When actually solving a particular task, R_{ext} will not be arbitrary (e.g. representing measurements of a certain subset of the variables), and only particular computational paths in the model might actually be relevant. If it is possible to anticipate them, there might be a way to control the "backward" propagation of distinctions appropriately. For instance, if the model is used for prediction, say support for FMEA, then the primary distinctions for variables that are functionally relevant have to be reflected by induced qualitative values of internal variables, but not vice versa, since the prediction always progresses from presumed faults to the output variables.

Future work will have to both push the analysis of conditions that prevent or guarantee the generation of a finite set of qualitative values and the development of appropriate specializations of the algorithm and the exploitation of general or task-dependent control heuristics. Other extensions to the theory and its application concern temporal aspects. We consider our theory to provide a basis for a proper treatment of time scale abstraction (Iwasaki 92) and, in particular, hybrid modeling. The reason is that whether or not certain changes can be either ignored or treated as discontinuous changes reflects what the current task requires in terms of significant distinctions in magnitude of variables, their derivatives and durations. To handle this as an instance of our framework, time has to included as a variable and integration rules must become part of the model relations.

Finally, there exists a more abstract and weaker version of our target: rather than computing qualitative values as sets of values of DOM_0, we might want to introduce distinctions only as (partially) ordered landmarks and then propagate ordering information through the model. While this eliminates some of the computational problems, the result may simply be too weak and ambiguous.

In summary, we are convinced that more research should be dedicated to this important practical problem since it is essential for bringing compositional qualitative modeling to real applications.

Acknowledgments

Many thanks to the members of the MQM group at the Technical University of Munich for discussions and collaboration. The reviewers' comments were very helpful in our attempt to improve the final version, although we certainly did not satisfy all their requests. This work was supported in part by the Commission of the European Union (#BE 95/2128) and by the German Ministry of Education and Research (#01 IN 509 41).

References

Iwasaki, Y. 1992. Reasoning with Multiple Abstraction Models. In Faltings, B. and Struss, P. eds. 1992. *Recent Advances in Qualitative Physics*. Cambridge, Mass.: MIT Press.

Struss, P. 1988. Mathematical Aspects of Qualitative Reasoning. *International Journal of Artificial Intelligence in Engineering* Vol. 3 Nr. 3, Computational Mechanics Publications, pp. 172-173.

Struss, P. 1990. Problems of Interval-Based Qualitative Reasoning. In Weld, D. and de Kleer, J. eds. 1990. *Qualitative reasoning about physical systems*. Morgan Kaufmann Publishers, pp. 288-305.

Struss, P. 1992. What's in SD? Towards a Theory of Modeling for Diagnosis, In Hamscher, W., Console, L., and de Kleer, J. eds. 1992. *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers, pp. 419-449.