On-Line Diagnosis of Dynamic Systems based on Qualitative Models and Dependency-recording Diagnosis Engines

Oskar Dressler

81739 München
Germany
dressler@informatik.tu-muenchen.de

Abstract

Dynamic systems impose an inherent complexity on simulation and diagnosis tasks. In particular, dependency-based diagnosis engines perform extremely bad on models of such systems. Putting these diagnosers *on-line* is a necessity from the application point of view, but a quite ambitious goal in view of previous attempts in this direction.

Understanding dynamic systems in terms used by the qualitative reasoning community and making use of the distinction between *intra*-state and *inter*-state constraints allows us to make a major step towards a satisfying solution.

Inter-state constraints are motivated by the continuity of physical variables and the direction of (the) time (arrow). Previous authors, e.g. [13],[14], and [16], have unconditionally assumed that predictions across time, i.e. inter-state constraints, are necessary for diagnosing dynamic systems. The key insight, however, that enables us to diagnose a large class of dynamic systems efficiently is that inter-state constraints often need *not* be checked for the purpose of consistency-based diagnosis at all! We identify sufficient conditions about models, observability, and faults for this result.

Based on this idea we employed a technique for caching and temporally generalizing inferences to build a new diagnostic engine, **Magellan-MT**. It is capable of diagnosing dynamic systems at multiple times but at the computational cost of static systems. Magellan-MT *requires qualitative models* to fully exploit these computational advantages.

We present empirical results from an application on ballast tank systems that are used on ships and offshore platforms.

1. Introduction

Over the last twenty years the electronic revolution has turned everyday products such as copiers and cars into networks of computers. Various bus systems allow using the same sensor reading at multiple places and help coordinating e.g. motor, brakes, and gearbox. By 1992, a typical high end car already contained 50 microcontrollers, 25 control units and had 500KB memory on board. Despite decreasing costs for "silicon-based" hardware the automotive electrical and electronic systems are estimated to make up more than 20% of the car production value by the year 2000.

1.1 Diagnosis Needs Everywhere

Although the electronics are needed for additional control features for e.g. meeting low emission standards, the system 'car' as a whole has become more vulnerable. A sensor, for example, which was not needed in the pre-electronic era, now provides information that is crucial for the operation of the vehicle. If it breaks, the motor might simply stop. Therefore most controllers contain surprisingly complex diagnosis (and repair) software. To a certain extent a controller recognizes faults in its environment and even takes compensating actions. A missing sensor reading, for instance, may be suspected to be caused by a broken connection. As a repair action the controller may estimate a value based on other available data. For a standard motor controller *hand-coded heuristic* knowledge of this kind makes up 40% of its software.

The apparently inevitable trend towards more complex systems is clearly visible in the automotive industry but by no means limited to it. The need for diagnosis arises everywhere in the technical domains. The extremely large number of variants (not only in the car domain) makes the hitherto common approach to providing the required diagnosis and repair software obsolete. A better treatment of the problem is needed. *A possible new candidate technology is model-based diagnosis.*

1.2 Diagnosis *and* Monitoring

Successfully deploying model-based diagnosis systems for complex technical devices ranging from cars and copiers to chemical plants etc. ultimately

- requires an on-line coupling with the artifacts via sensors and actuators and
- an integration of monitoring and diagnosis phases.

Monitoring means tracking the system and signaling

deviations from expected behavior. **Diagnosis** means identifying faulty parts. An integrated system interleaves monitoring and diagnosis. It e.g. switches back from diagnosis to monitoring mode interpreting the measurements coming from the sensors under the hypotheses that the identified components are broken.

Consider figure 1 taken from an application on ballast

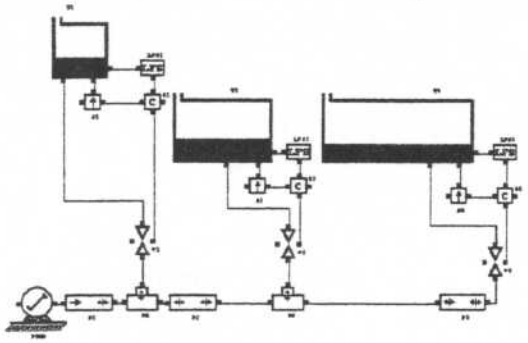

Figure 1 A ballast tank system

tank systems ([9]). A collection of ballast tanks of various sizes is placed at different locations on ships and offshore platforms. Depending on load, wind and sea motion, water is pumped into or out of some of the tanks or the sea. A leaking valve is normally tolerated in this context. After localizing the fault a truly intelligent supervision system would probably mark the valve for inclusion in the routine repair schedule and then continue monitoring, but *know about the leaking valve*.

1.3 The All-Components-Included Conflict Syndrome

Dynamic systems have an inherent complexity for (simulation and) diagnosis tasks. The system state at later times depends on system states occurring before. Predicting the behavior of a dynamic system means making predictions about future values of system variables. This apparently inevitable characteristic makes it difficult for dependency-based diagnosis engines to diagnose dynamic systems well.

The success of diagnosis systems in the GDE tradition rests on the use of dependencies recorded by some truth maintenance system (TMS). Dependencies recorded for every predicted value of the system variables allows tracing back contradictory derivations to their origins. This enables diagnosis engines such as GDE ([8]), GDE$^+$ ([20]), Sherlock ([7], [5]), DDE ([11], [12]), and others to first identify conflicting assumption sets and then to generate diagnoses.

Keeping track of dependencies across different times, however, quickly accumulates large sets of involved components because of *global feedback effects across time*. For this reason the straightforward application to the diagnosis of dynamic systems fails. The assumption sets underlying detected discrepancies are too large, and hence, not very informative. This, for instance, happens with the approach described in [14]. In the extreme they contain almost all components in the device. Then the information in them reduces to 'there is something wrong with the device', or equivalently 'any component(s) could be faulted'. In such cases the various dependency-based diagnosis engines which perform so successfully on static systems will generate huge sets of candidates which represent nothing more than this information.

Where is the way out? Can dependency-based diagnosis systems deal with dynamic systems efficiently?

1.4 Solution

One solution is to provide for extensive observations and to guarantee enough readings so that most of the global feedback loops can be broken. This is the route taken in [3], [2], and certainly appropriate in the case of analog circuits *in the laboratory*. In the field and on board, however, the cost of required sensors would be prohibitive. The general and economically much more interesting question of how to diagnose well

- on-line, *even on-board* (!), and
- under limited observability

is addressed here.

Our approach is based on just one critical insight that questions the relevance of predicting *future* values of system variables for the purpose of consistency-based diagnosis. In the ballast tank application and for a large class of models of dynamic systems *all* relevant faults can be diagnosed *without* considering so-called *interstate constraints* at all. These constraints describe the space of admissible (qualitative) state transitions. For a lot of physical system models the origin of these constraints lies in the continuity of physical variables over time[1] and the direction of (the) time (arrow). A physical system, however, - even a broken one - neither can reverse the flow of time nor can it violate continuity of physical variables. Therefore, checking these constraints is needless from the viewpoint of consistency-based diagnosis! The rest of the paper shows how this idea is put to

1. By a continuous physical variable we more precisely mean a *reasonable* function over time as defined in [15] which essentially is a continuously differentiable function.

work.

Section 2 introduces the distinction between intra-state and inter-state behavior and shows how it is exploited for diagnosing dynamic systems. Section 3 sketches our approach to an integrated diagnosis and monitoring system. In section 4 we present a summary of extended truth maintenance functionalities that allow for temporal indices and temporal generalization ([10]). The next section shows how we use these techniques to extend our diagnosis framework for static systems ([1], [11], [12]) to a more general one which is applicable to dynamic systems. Finally, in section 6 we present empirical results.

2. Diagnosis of Dynamic Systems

2.1 The Ballast Tank Model

Ballast tank systems like the one in figure 1 can be modelled using variables for the height of water in the tanks, the pressures at their bottoms, the flow and pressures in the pipes, the pressure before and after the pump etc. In [9] the complete models can be found. There are two remarkable facts about this system and its model that are of interest here:

- *reduced observability*: only a small subset of the variables can be measured, e.g. by the pressure sensors at the tank bottoms
- *dynamics*: at any given time the *partial* information given by the measurements can be completed by applying component models locally. The future development of the system, however, is additionally determined by knowledge that is *not* represented in the component models. It consists of
  - knowledge about the direction of (the) time (arrow) and
  - knowledge about the continuity of physical variables. We come back to this point later.

2.2 Global Feedback Across Time

The straightforward application of standard dependency-based diagnosis engines, e.g. Sherlock ([7]) or GDE$^+$ ([20]), to this system model requires that every inference based on the system model is recorded as a justification (expression) in some TMS. When the predictive engine uses the component models to complete the partial description of the system state at time t as given by the sensor readings it produces a constant stream of justifications submitted to the TMS. Likewise it submits justifications when future values of system variables are predicted. It is exactly this latter step that results in global feedback across time. It drags down the performance in terms of accuracy of diagnosis (because conflicts are too large and, hence, the set of diagnoses is indiscriminate) and wastes computational resources.

Long chains of inference originating at some component C and at some time t_i will eventually lead to (new) values of system variables connected to the same component C, but the (new) values hold at time t_{i+1}! The term 'feedback' refers to the fact that component C now has become part of a loop. These loops tend to encompass large parts of the system because predicting a future state (with low ambiguity) out of a current state (even in the case of qualitative simulation) requires an almost completed current state.

Viewed in temporal terms the loop becomes a spiral. A component model, say ok-C, is used at time t_i and then again at t_{i+1} along one and the same inference, i.e. justification, chain.

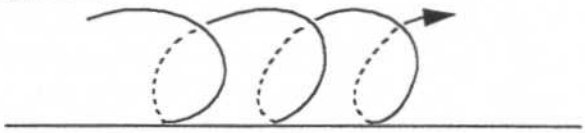

Figure 2 t_i t_{i+1}

There is some confusion in the diagnosis literature surrounding the notion of 'feedback'. The authors of e.g. [14], [3], [2], [13], and [16] are not very clear in that respect. The confusion seems to be caused by *not* distinguishing between the assumptions made at different times. The assumption that C is working correctly at time t_i is different from the one about C working correctly at t_{i+1}. If this distinction is not made, the spiral - in as far as the tracking of underlying assumptions is affected - collapses to a loop (and brings us back to square one, i.e. the all-components-included conflict syndrome).

In the ballast tank application we have found - by making the above mentioned distinction - that *all* of the faults (classified relevant by a shipbuilding company) can be detected and located based on conflicting assumption sets each of which has a single common time index t_i. This means that

- discrepancies between expected and observed behavior show up at one time index (interval or point) and that
- they contain enough information for localizing the faults.

This could be just a lucky accident in our application. The following considerations, however, show that we are

actually dealing with a larger class of dynamic systems or more precisely their models.

2.3 Understanding Dynamic Systems in Qualitative Simulation Terms

At any time a physical system is in some state that can be described qualitatively. Each system variable takes on a qualitative value which is constrained by the qualitative values of other variables. These restrictions are given by *intra-state constraints*. An example is Kirchhoff's current law; the sum of the flows into a node is zero. But not only algebraic equations, also differential equations are allowed.

Inter-state constraints, in addition, make restrictions on the way the system will develop over time, i.e. which transitions to other qualitatively described states are admissible. For example, a physical variable like the pressure at a tank bottom cannot change its value in discrete steps.[1] It must change continuously visiting every real value in between. When the qualitative abstraction of a physical variable is considered, the real number line is divided into a set of qualitative values, i.e. points (so-called landmarks) and open intervals in between ([15]). The physical variable can only change its qualitative value by switching to adjacent values. Suppose the domain of the pressure variable was divided into qualitative values 'low', 'medium', and 'high' and landmarks L_1 and L_2 defining the borders between them. If the pressure is currently 'low', i.e. in the interval $(0, L_1)$, and keeps increasing, then its next qualitative value is the landmark L_1, then the interval (L_1, L_2), i.e. 'medium', etc.

In this way each of the qualitative variables is restricted in its future development. Of course, the future qualitative state of a system is further constrained by physical laws and the system description that were expressed as intra-state constraints.

This is the basis for a variety of (qualitative) simulation algorithms. They alternate between consistency-checking of states (intra-state constraints) and successor-state generation (inter-state constraints). For instance, the basic QSIM algorithm ([15]) takes a partial description of an initial qualitative state and completes it using *intra*-state constraints. It generates potential successor states by *inter*-state constraints which are then checked and completed by *intra*-state constraints again. And so on. In QSIM inter-state constraints are called P- and I- transitions. They encode the knowledge about admissible transitions depending on whether the current state is associated with a time point (P- transition) or a time interval (I- transition).

Mathematically speaking, inter-state constraints use the intermediate value theorem and the mean value theorem. There is nothing more in them than the assumption that physical variables behave *reasonably* (as e.g. defined in [15]), which essentially means that they are continuously differentiable functions over time.

Numerical simulation is based on the same assumption. The generated successor state is, however, unique and the consistency checking of intra-state constraints is built into it's construction.

Naturally, no physical system will violate this assumption. One must, however, be aware that *models* of physical systems can do so!

In the light of this analysis the results obtained in the ballast tank application become less mysterious. A broken system behaves like a broken system *all of the time*, although

- a fault may not be visible all of the time because the conditions under which the system operates do not expose the malfunction all of the time (think of the FDIV bug in Intel's Pentium) and
- a fault may be present but produce discrepancies that lie in the future. An example is a *potentially* overflowing tank. Unless somebody (maybe an automatic controller) stops the pump, the tank *could* overflow.

Under sufficient assumptions, in both cases the fault will be detected (and diagnosed) when the situation occurs, i.e. with intra-state constraints only. Extending diagnosis and monitoring algorithms so that they can warn about and analyze *potentially* occurring malfunctions that lie ahead is an interesting perspective.

2.4 The Scope of the Approach

From the application we can abstract assumptions that are sufficient to diagnose with intra-state constraints only:

- A control system pre-processes sensor information, deals with noisy data, and computes qualitative information, e.g. signs, about derivatives of measurable variables.
- The time between snapshots is small compared to the evolution speed of the physical system.

1. Obviously, we are not dealing with ballast tanks and similar devices on the level of quantum mechanics.

- Sensors provide measurements all the time, not only sometimes; if the measurements are distributed sparsely over time, there is not much that can be done with intra-state constraints alone. It is not required, however, that all variables be measured.

Observations provide a *partial* view of the system state. As in the static case, observability, model granularity, the degree of completeness of the predictive engine, and the class of relevant faults interact. If a coarse model is used, faults may be indistinguishable from normal operation. An incomplete predictor may fail to detect discrepancies. The set of available observations may be insufficient to reveal the manifestation of a fault. These individual deficiencies can be compensated by strengthening other contributors, e.g. a more complete predictor, more observations, or a finer model. Whether or not the selected setup suffices also depends on the set of *relevant* faults. Therefore,

- available observations, model, and predictor must be sufficient to *significantly limit* the set of consistent states with respect to the relevant faults. Note, that it is not required that the system state be uniquely identified within the system's total envisionment.

Furthermore, we have assumed that

- the models of the system are similar to the ones used for qualitative simulation.

At first glance this appears to be a minor restriction. After all 'qualitative differential equations' can be obtained more or less directly from the conventional differential equations describing physical systems. However, for the devices we ultimately want to diagnose, discrete event type models and hybrid models of various sorts including models of actions are of interest, too. For example, most of the controllers in these devices can be described as finite state machines. *Wrong* transitions are physically possible in these contexts. This means that at least some faults will only be detected by considering data about the transient.

2.5 Why does it work ?

Observations and intra-state constraints determine a set of admissible states rather than a single one. And this set can be large. After all the problem with dynamic systems is that they have memory and only partial information about their states is available. The intra-state constraints only check whether the current system state occurs in the system's total envisionment.

The on-line coupling provides further restrictions. Initially, the physical system is in a known state. Assume that the system in a fixed condition, ok or faulty in some particular way. Its further development is guided by the inter-state constraints.

Only a very limited number of adjacent states (allowed by the inter-state constraints) exist. No physical system that falls under our modelling assumption can arbitrarily "jump" to another state. Therefore, the system state as seen by the monitoring- and diagnosis-system provides a narrow focus. Within this focus the intra-state constraints - altoough they appear to be very weak - suffice to identify the relevant faults.

Please, note that in the above argumentation we have adopted the non-intermittency assumption ([17]). The physical system, however, may not be in a fixed state but may "flip" between different fault or ok modes. This would result in different sets of diagnoses at different times. In section 5 we show how the non-intermittency assumption can be exploited in our framework.

3. Integration on Monitoring and Diagnosis

3.1 Monitoring as Diagnosis without Discrepancies

We use qualitative models for both, diagnosis and monitoring:

- For consistency-based diagnosis engines they prove to be especially useful; more detailed models become obsolete, when a qualitative abstraction of them has been refuted ([19]). There often is no need to explore further details.
- For monitoring only significant deviations from normal operation are of interest. Using a qualitative model for a component's normal mode, one can capture the complete set of good behaviors instead of just a single one. Even more important, faulty behavior which depends on unknown parameters such as the size of a leak, the position of a hole, or the deviation from a given frequency can be described qualitatively as *significant* deviations from good behavior.

When no discrepancies between observed and expected behavior are detected, the empty diagnosis is computed meaning that every component is working as expected. With this in mind, we can view monitoring as *'diagnosis without discrepancies'*; the monitoring engine is nothing but a diagnosis engine that operates in 'idle mode'.

The purpose of models for monitoring and diagnosis, however, is substantially different. While the former are

only needed to *detect* malfunctions, the latter must have enough detail for *localizing* malfunctioning components. Diagnosis systems like DP ([19]) and Magellan ([1]), however, can use multiple models of different granularity during one and the same diagnosis session. Their diagnosis process starts with coarser models that are suitable for monitoring, too.

3.2 Speed-up by Caching Inferences is possible

For on-line coupling prediction, i.e. the application of the system model, is necessary at the rate of incoming data. *Speed is of prime interest.* Ideally, consistency should be checked at the sampling rate of the sensors.

Suppose we are monitoring a running motor. Incoming real values are first mapped to their qualitative abstractions, and then fed into the prediction machinery. This we can afford to do every, say few seconds, depending on the cost for running the model. Using qualitative models, nothing changes most of the time in qualitative terms, since different real values are mapped to the same qualitative value. Even if a monitored variable changes its qualitative value this most often has very limited effects on the other system variables. Therefore, we end up making more or less the *same* predictions every few seconds while we would like a higher sampling rate and do new predictions only.

Likewise the generation of diagnoses, now an integrated part of the monitoring phase, is carried out over and over again although a conflicting assumption set identified at one time is often re-discovered at other times. The set of diagnoses at a given time, however, only depends on the detectable discrepancies at that time. Most of them may have been detected previously.

For both, prediction and diagnosis candidate generation, caching of inferences pays off.

4. Our Kind of Caching: Prediction Sharing Across Time and Contexts

With respect to time the system description SD, observations OBS, and mode assumptions Π fall into two different classes. The system description is temporally generic in the sense that it describes behavior independent of the specific time at which it takes place. Observations, working hypotheses and mode assumptions, however, may change their truth value over time. Therefore, instead of re-doing inferences based on the system description, we can factor out the relevant computations (and do them only once!) by *caching* inferences made for a specific time and *generalizing* them to other times.

We start from statements like proposition α holding at time t_i, $\alpha@t_i$, which we call temporally indexed statements.

Definition: The **temporal extent of** α, TE (α), denotes the set $\{t_i \mid \alpha \text{ holds at } t_i\}$.

Delays for consequences are not of interest for diagnosis based on intra-state constraints only. It follows immediately that the derivation of α can be generalized from the single time index t_i to sets of time indices.

Definition: A set GS of non-universally holding formulas is called **ground support for** ϕ iff there exists SD' $\subseteq$ SD such that SD' $\cup$ GS $\models \phi$.

Lemma: If GS is a ground support for ϕ then $\bigcap_{\alpha \in GS} TE(\alpha) \subseteq TE(\phi)$

This means we can generalize a derivation of ϕ at a specific time t_i to the intersection of temporal extents of ϕ's support. Whenever all the propositions in GS hold at some $t_i \neq t_j$ we know *without re-deriving* ϕ that it holds at t_j, too.

Definition: For those propositions α that may occur in the ground support of derived formulae we introduce symbols TE_α to represent TE(α). These propositions α are exactly the propositions for which temporally indexed statements are available, i.e. observations and mode assumptions. Symbols like TE_α are called **temporal base symbols**. Using these symbols each atom is labelled with a *unique* symbolic representation of its temporal extent (**temporal label**) like e.g.

$\mathbf{TL}(\phi) = \{\{TE_{\alpha_{11}}, ..., TE_{\alpha_{1n}}\}, ..., \{TE_{\alpha_{m1}}, ..., TE_{\alpha_{mk}}\}\}$.

The similarity to "logical" labels in the ATMS ([4]) is not accidental. In [10] we show how the ATMS both, serves as a vehicle for computing temporal labels, and allows integrating logical and temporal labels.

There are two principal entry points for statements with mixed, temporal and logical, context information.

- On a basic level we can capture logical context information by using justifications like

  $A \wedge B \wedge t_{17} \rightarrow EXT - TE_{\alpha_i}$

where A and B are logical assumptions and t_{17} is an assumption related to a specific time. Consequently, the logical label as defined in [4] is

$LL(EXT - TE_{\alpha_i}) = \{\{t_{17}, A, B\}, ...\}$.

The symbols $EXT - TE_{\alpha_i}$ are created for each TE_{α_i} to denote the extensional description of times and logical

contexts where the individual TE_{α_i} hold.

- On the level of recorded problem solver inferences logical assumptions, say A and B, are added to the antecedents: $A \wedge B \wedge \alpha_1 \wedge ... \wedge \alpha_n \rightarrow \beta$

The temporal labels then are *relative* to the logical context.

Besides the symbols TE_{α_i} ("normal") logical assumptions may appear in the labels, and $\mathbf{TL}(\phi, \Theta)$ now is called temporal label of ϕ under logical assumptions Θ.

The theorem below from [10] relates labels as computed by the ATMS to temporal labels under assumptions.

Theorem: Let *TBS* be the set of temporal base symbols,
TBS-ASSM $\subseteq$ *ASSM* be the subset of ATMS assumptions corresponding to temporal base symbols,
Ψ: *TBS-ASSM* $\rightarrow$ *TBS* be the bijective mapping that associates assumptions with the symbols they were created for and Θ be a set of logical assumptions.
Then $TL(\phi, \Theta) =$
$\{\{e' \mid e \in E \cap \text{TBS-ASSM} \wedge e' = \Psi(e)\} \mid$
$E \in LL(\phi) \wedge (E \setminus \text{TBS-ASSM}) \subseteq \Theta\}$

There is a two stage approach to answering queries. The evaluation is done relative to the logical context specified as part of the query.

Lemma: ϕ holds at t_i under assumptions θ iff $\exists\ S \subseteq \theta$:

$$\{t_i\} \cup S \in \bigcup_{tenv \in TL(\phi, \theta)} \bigcap_{TE_{\alpha_i} \in tenv} LL\left(EXT - TE_{\alpha_i}\right)$$

5. Diagnosis at Multiple Times

5.1 The Static Diagnosis Framework

In [11]and [12] we introduced a *static* diagnosis framework that we use as a starting point for our *multiple time* framework. With the approach to consistency-based diagnosis proposed in [6] we share that two basic elements of the logical theory are

- the *behavior model* of the system *(SD)* represented as sets of formulas, and
- a set of *observations* of the (broken) system's actual behavior (*OBS*).

In contrast to [6] we introduced

- a set of possible *working hypotheses* (W-HYP), i.e. retractable assumptions suited to simplify the reasoning process, such as "sensors reliable", "single fault only", "non-intermittent-faults-only" or modeling assumptions ([19]),
- a set of *different, mutually exclusive behavior modes*, modes(C_i), for each component $C_i \in$ COMPS, represented as propositional atoms. Accordingly, SD can contain models of several component faults which are associated with the respective modes. There is an unknown mode which has no model attached to it and, hence, can never be refuted.
- a *partial order on the modes* of each component, $\geq \subseteq \text{modes}(C_i) \times \text{modes}(C_i)$, expressing differences between the modes such as the frequency of occurrence, likelihood, or criticality. We call this a *preference*. The correct behavior mode is the most preferred and the unknown mode the least preferred one.

Diagnosing a system means finding out what is wrong and which components work properly, which translates into appropriately assigning exactly one mode to each component. We represent such *mode assignments* as sets of modes:

$\Pi = \{m_{j_i}(C_i) \mid C_i \in \text{COMPS}\}$

where $m_k(C_i) \in \Pi \wedge m_l(C_i) \in \Pi \Rightarrow k=l$.

Preferences among modes induce a preference order on mode assignments:

For $\Pi = \{m_{j_i}(C_i)\}$ and
$\Pi' = \{m'_{j_i}(C_i)\}$ $\Pi \geq \Pi' :\Leftrightarrow \ \forall i\ m_{j_i}(C_i) \geq m'_{j_i}(C_i)$.

The preferred diagnoses are the most preferred mode assignments that accord with the system description, the observations, and working hypotheses:

Definition: (Preferred Diagnoses)

Let $w \subseteq$ W-HYP be a set of working hypotheses.

A mode assignment Π is a diagnosis under w, iff SD $\cup$ OBS $\cup$ w $\cup$ Π is consistent. Π is a **preferred diagnosis** under w iff no other diagnosis under w is strictly preferred over it: For all diagnoses Π' under w:

$$\Pi' \geq \Pi \ \Rightarrow \Pi' = \Pi$$

Default logic ([18]) allows characterizing preferred diagnoses. Preferences among component modes are translated into defaults. If e.g. the component mode $m_j(C_i)$ of component C_i is preceded by modes $\text{pre}_j(C_i) :=$ $\{m_k(C_i) \mid m_k(C_i) > m_j(C_i)\}$ then the default

$$\bigwedge_{m_k(C_i) \in \text{pre}_j(C_i)} \neg\, m_k(C_i) : m_j(C_i) \,/\, m_j(C_i)$$

controls the assignment of mode $m_j(C_i)$ in the intended way.

Theorem: Let $D = \{def_{ij}\}$ be the set of preference defaults. Π is a preferred diagnosis under w iff Cn(SD $\cup$ OBS $\cup$ w $\cup$ Π) is a consistent extension of the default theory (D, SD $\cup$ OBS $\cup$ w).

5.2 The Multiple Time Diagnosis Framework

For the temporal extension towards multiple times (points or intervals) it suffices to identify the system description and (most of) the working hypotheses as temporally generic formulas whereas observations and assumptions about modes depend on the time when they are made and when they apply, respectively.

Definition: (Preferred Diagnosis at Time t)

Let $w \subseteq$ W-HYP be a set of working hypotheses. A mode assignment Π is a **diagnosis at time t** under w, iff SD $\cup$ OBS@t $\cup$ w $\cup$ Π@t is consistent where OBS@t is the set $\{\alpha@t \mid \alpha \in$ OBS and t is the time at which the observation holds$\}$ and Π@t is the set $\{m@t \mid m \in \Pi$ and t is the time for which diagnoses are considered$\}$. Π is a **preferred diagnosis at time t** under w iff no other diagnosis under w at time is strictly preferred over it.

The purpose of the time parameter is to select observations made at time t and modes applicable at time t. *This is a very limited use of time.* Therefore it is not surprising that the characterization of preferred diagnoses at time t can be done in a way similar to the static framework. In analogy to the temporal extent of a proposition we define the temporal extent of a diagnosis.

Definition: The temporal extent of a diagnosis Π under working hypotheses w, TE(Π, w), denotes the set $\{t_i \mid \Pi$ is a diagnosis at time t_i under w$\}$

Theorem: Let D=$\{def_{ij}\}$ be the set of preference defaults indexed by time t, i.e. defaults of the form

$$\bigwedge_{m_k(C_i) \in pre_j(C_i)} \neg\, m_k(C_i)@t : m_j(C_i)@t \,/\, m_j(C_i)@t$$

Π is a preferred diagnosis at time t under w iff Cn(SD $\cup$ OBS@t $\cup$ w $\cup$ Π@t) is a consistent extension of the default theory (D, SD $\cup$ OBS@t $\cup$ w).

The characterization of preferred diagnoses in [11] lends itself to an efficient implementation (see [12]). Preferred diagnoses are computed by submitting a set of justifications related to the defined preferences and assumptions about component modes to the ATMS. After doing so, the logical label of a certain proposition Φ represented in the ATMS characterizes the preferred diagnoses (for details see [11]). With the introduction of temporal labels as shown in section 4, the same proposition's label now not only characterizes the preferred diagnoses but also their temporal extent.

Lemma: Let Φ be the ATMS node whose label characterizes the preferred diagnoses as in [11] and let TBSASSM $\subseteq$ ASSM be the subset of ATMS assumptions corresponding to temporal base symbols.

Then Π is a preferred diagnosis at time t under working hypotheses w iff $\exists\ S \subseteq w$:

$$\{t_i\} \cup S \in \bigcup_{tenv \in TL(\phi, w)} \bigcap_{TE_{\alpha_i} \in tenv} LL\left(EXT - TE_{\alpha_i}\right)$$

$\wedge\ \Pi \subseteq Cxt(\{t_i \cup S\})$ where $Cxt(\{t_i \cup S\})$ denotes the set of atoms in the deductive closure of $\{t_i \cup S\}$ (as computed by the ATMS).

The temporal extent of a diagnosis Π under working hypotheses w is

$$TE(\Pi, w) = \bigcup_{E \in LL(\Phi) \wedge E \cap WHYP \subseteq w \wedge \Pi \subseteq Cxt(E)} \bigcap_{TE_{\alpha_i} \in E \cap TBSASSM} TE(\alpha_i)$$

5.3 Diagnosis across Different Times

The non-intermittency assumption as discussed in [17] can be treated as a working hypothesis about component modes. When at some other time, due to changed system variables, a component could be presumed innocent or operating in a different preferred mode, the non-intermittency assumption restricts these possibilities. It provides a link between modes at different times. Consequently, either the current or the diagnoses at other times might change. Since we are, however, monitoring the artifact "on board" from the start of its operation, we shall necessarily be noting a switch from ok mode to fault mode when a malfunction is detected. Therefore, it is crucial to apply the non-intermittency assumption to fault modes only. Otherwise, we would have to classify the ok-mode of the component as intermittent.

When the working hypothesis 'non-intermittent-faults-only' has been introduced, simply stating for fault modes $m_k(C_l)$ that

non-intermittent-faults-only $\rightarrow$

$$\forall (t_i \in TW) \quad : [m_k(C_l)@t_i \leftrightarrow m_k(C_l)@t_{i+1}]$$

suffices. TW is a window of time indices that controls over which time period the non-intermittency assumption is enforced. By the implication above, the modes are now effectively "locked" over this period. It is important that the non-intermittency assumption is *not* in effect during prediction. Otherwise, we would be introducing global feedback effects again through this back-door. Prediction is therefore done as before. This means that during one prediction run conflicting assumption sets (conflicts) will still be indexed by a single common time index.

The conflicts discovered over the period TW are taken into consideration by activating the non-intermittency assumption during diagnosis candidate generation. Additionally, all of the time indices $t_i \in TW$ (i.e. assumptions from the TMS viewpoint) are put into focus. The preferred diagnoses in the time window TW are then given as the preferred diagnoses at every $t_i \in TW$.

Definition: Π is a preferred diagnosis in the time window TW under working hypotheses w iff $\forall (t_i \in TW)$: Π is a preferred diagnosis at t_i under w.

Theorem: Let the non-intermittency assumption be in effect in the time window TW.

Π is a preferred diagnosis in the time window TW under working hypotheses w iff Π is a preferred diagnosis at some $t_i \in TW$ under working hypotheses $w \cup \{non-intermittent-faults-only\} \cup TW$

6. Empirical Results

Figure 3 shows a typical distribution of run time and of necessary new predictions for the monitoring case. The important point is the strict alignment between the two. In [10] we report experimental results on different device configurations. *All* these experiments show similar patterns. In the beginning the prediction cost (run time) is substantial. No previous predictions have been cached and every possible derivation has to be done explicitly. Later on *when* a number of *variables change their qualitative value*, prediction cost increases but does not reach the initial cost. Without prediction sharing run time is in the range of the initial cost all of the time! We are gaining a lot by inference caching. The results for the integrated

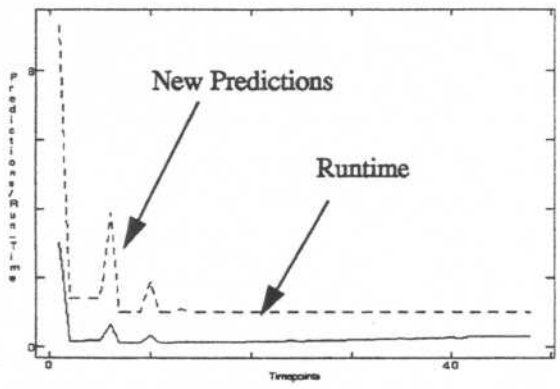

Figure 3 Monitoring the 3-Tank system from figure 1, considering 10 different test vectors occuring at most 10 times. Prediction time for first timepoint: 3.04s, average for additional timepoints: 0.21s

diagnosis & monitoring system are similar. There are,

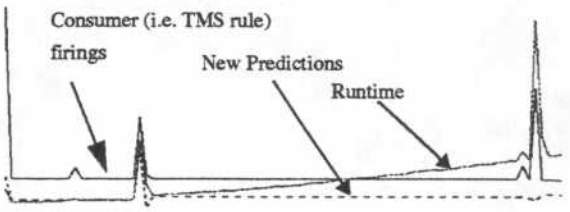

Figure 4 A diagnosis scenario in the ballast tank system from figure 1. A double fault is diagnosed while the tanks are filled. At t_{20} a first fault is encountered. The second, overlapping fault (a leaking valve) becomes visible at about t_{170}

however, additional peaks in the run time distribution (diagnosis time!) when the artifact exhibits faulty behavior, i.e. diagnosis becomes necessary. When a diagnosis has been determined, *no* extra diagnosis time is needed at later times unless a new fault is encountered in the scenario data. Figure 4 again shows an alignment between the curves. The linear increase of run time (vs. new predictions) is caused by a poor implementation of a focus switching routine. In principle, new predictions and runtime should be parallel curves.

7. Conclusions

Work on a theory of modeling for diagnosis has begun only recently (see e.g. [19]). As we try to cover larger application classes we will need a variety of modeling formalisms and languages. A careful analysis of them is needed from a *strictly diagnosis point of view*. Otherwise, the straightforward combination of modeling schemes and diagnosis engines will most likely lead to non-satisfying results both in terms of accuracy and efficiency.

Acknowledgements

Discussions with Toni Beschta, Claudia Böttcher, Philippe Dague, Michael Montag, Peter Struss, LouiseTraves-Massuyes, and attendees of the MQD Workshop in Chambery were helpful through a series of counterexamples and „counterexamples" which led to the exposure of hidden assumptions. The process continues.

This work was done at the Siemens Corp. Research Labs which I have left recently.

References

[1] C. Böttcher and O. Dressler. Diagnosis Process Dynamics: Holding the Diagnostic Trackhound in Leash. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann Publishers, San Mateo, CA,

1993.

[2] P. Dague, P. Deves, O. Jehl, P. Luciani, and P. Taillibert. When Oscillators Stop Oscillating. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*, pages 235–241. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[3] Philippe Dague, Philippe Devès, Pierre Luciani, and Patrick Taillibert. Analog systems diagnosis. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*, pages 229–234. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[4] J. de Kleer. An Assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[5] J. de Kleer. Focusing on Probable Diagnoses. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 842–848, Anaheim, 1991. Morgan Kaufmann Publishers, San Mateo, CA.

[6] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing Diagnoses and Systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[7] J. de Kleer and B. C. Williams. Diagnosis with Behavioral Modes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1324–1330. Morgan Kaufmann Publishers, San Mateo, CA, 1989.

[8] J. de Kleer. and B.C. Williams. Diagnosing Multiple Faults. *Artificial Intelligence*, 32:97–130, 1987.

[9] O. Dressler, C. Böttcher, M. Montag, and A. Brinkop. Qualitative and Quantitative Models in a Model-Based Diagnosis System for Ballast Tank Systems. In *Proceedings of the International Conference on Fault Diagnosis (TOOLDIAG)*, pages 397–405, Toulouse, France, April 1993.

[10] O. Dressler and H. Freitag. Prediction Sharing Across Time and Contexts. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1136–1141. AAAI Press / The MIT Press, Menlo Park Cambridge London, 1994.

[11] O. Dressler and P. Struss. Back to Defaults: Characterizing and Computing Diagnoses as Coherent Assumption Sets. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 719–723, John Wiley & Sons, 1992.

[12] O. Dressler and P. Struss. Model-based Diagnosis with the Default-based Diagnosis Engine: Effective Control Strategies that Work in Practice. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 677–681. John Wiley & Sons, 1994.

[13] G. Friedrich and F. Lackinger. Diagnosing temporal misbehavior. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1116–1122. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[14] T. Guckenbiehl and G. Schäfer-Richter. SIDIA: Extending Prediction-based Diagnosis to Dynamic Models. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*, pages 309–317. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

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

[16] Hwee Tou Ng. Model-Based, Multiple Fault Diagnosis of Time-Varying, Continuous Physical Systems. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*, pages 242–246. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[17] O. Raiman, J. de Kleer, V. Saraswat, and M. Shirley. Characterizing Non-Intermittent Faults. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 849–854. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[18] R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[19] P. Struss. What's in SD? Towards a Theory of Modeling for Diagnosis. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[20] P. Struss and O. Dressler. Physical Negation - Integrating Fault Models into the General Diagnostic Engine. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*, pages 153–159. Morgan Kaufmann Publishers, San Mateo, CA, 1992.[7] J. de Kleer and B. C. Williams. Diagnosis with Behavioral Modes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1324–1330. Morgan Kaufmann Publishers, San Mateo, CA, 1989.