Model-based Troubleshooting of Complex Technical Systems Using Integrated Qualitative Techniques

F. A. Lackinger*
Technical University of Vienna Vienna, Austria A-1040

I. Obreja
Alcatel-ELIN Research Center, Vienna Vienna, Austria A-1210

Abstract
Monitoring and diagnosis of safety-critical dynamic systems are challenging tasks for knowledge-based programming. While current control systems usually rely on the total availability and computability of numerical information, they necessarily fail if these assumptions do not hold. As a consequence, we are in urgent need of intelligent control systems which have the ability to perform monitoring and diagnosis even if only incomplete or abstract pieces of information are available. Additionally, we have to integrate monitoring and diagnosis into an overall on-line control cycle in order to apply our concepts to dynamic systems which are characterized by time-varying parameters. In this work we present an implemented expert system which fulfills these requirements.

1 Introduction
In this paper we present research work done in the field of model-based diagnosis of Nuclear Power Plants (NPP) using qualitative reasoning techniques.

The main goal of our work is to overcome the limitations of most current control systems. These limitations originate from the used quantitative system descriptions which require the total availability of numerical data. Additionally, most of these control systems use heuristically derived shallow knowledge (e.g. cause-consequence rules) about the behavior of the system under control.

Conversely, we accept the partial unavailability of numerical data in the course of system malfunction and therefore resort to qualitative reasoning techniques. Our approach integrates parameter- and component-oriented qualitative system models into a hybrid representation language which is based on the QSIM - constraint language [14].

Qualitative simulation of the system models is applied in order to predict possible system behavior for dynamic consistency checks. This simulation phase is used to verify the qualitative models with respect to fault coverage and expressiveness.

We combine model-based monitoring (used for fault detection) and diagnosis (used for fault localization) in order to deal with behavioral abnormality in the NPP in an efficient and effective manner. The features of our implemented expert system DIAMON ¹ will be demonstrated during the control process of an essential part of the NPP, the Emergency Feedwater System (EFWS). The presented model-based approach guarantees the detection and localization of single and multiple faults in the highly safety-critical EFWS.

Figure 1 depicts the modeling/inferencing tasks of our approach and motivates the following discussion.

We open the discussion with a presentation of the complex quantitative model of an EFWS in Section 2. In Section 3, we continue with the construction of the qualitative system model and present an extended constraint language for knowledge representation.

In Section 4, we introduce the basic concepts of our integrated control algorithm. We will then present the achieved results of some control sessions in Section 5.

Finally, in Section 6 we compare our approach to related research work in the field and give some topics for future research in Section 7.

2 A Quantitative Description of the EFWS
The EFWS is a standby system which would not be operated during normal plant operation. The task of

¹DIAMON stands for DIAgnosis and MONitoring Tool.
the EFWS is to provide full cooling of the Reactor Coolant System in emergency conditions [1].

A simplified schema of the EFWS at Seabrook is shown in Figure 2.

The EFWS is automatically activated in three cases of NPP malfunction:

1. loss of offsite power (LOOP)
2. low-low level in any steam generator
3. loss of alternative current (LOAC)

In the third case only the turbine driven pump (TDP-37A) will be operating. Due to the complexity of the device, we have divided the EFWS in three subsystems (see Figure 3):

- supply part \((BLOCK_A)\)
- common header \((BLOCK_B)\)
- supply lines to steam generator \((BLOCK_C)\)

In this paper we concentrate on monitoring and diagnosis of the supply part subsystem. Therefore, a detailed description of the other subsystems of the EFWS is given in [1].

The supply part subsystem consists of two pumps (TDP-37A, turbine driven, and MDP-37B, motor driven), each supplied by individual suction lines from the Condensate Storage Tank (CST). Each pump suction line to the CST contains two manual isolation valves (V154, V155, V158 and V159).

During operation of the EFWS, both pumps discharge into a common header (consisting of e.g. V64, V125, etc.). The common header in turn supplies four individual lines to each of the four steam generator main feed lines (SG11A-SG11D).

### 2.1 Quantitative Model

We introduce the quantitative equations which describe the correct behavior of the EFWS.

\(Q_i\) denotes the flow rate of the coolant water for pipe \(i\), \(P_{13}\) is the pressure at the position 13 - see Figure 2, \(P_0\) is the input pressure, \(g\) is the gravity acceleration, \(\rho\) is the fluid density, \(H\) is the fluid height in the CST, \(K\) is the pressure drop coefficient (for valve, pipe, and pump)).

- **conservation of water-flow:**
  \[
  \sum_{i=1}^{n} Q_i = 0 \text{ for each pipe crossroads (} n \text{ amount of pipes)}
  \]  
  (1)

- **input pressure for a system component (e.g. for the pump TDP-37A):**
  \[
  P_{13} = P_0 + \rho g H + (K_{V154} + K_{V155} + K_{pipe} + K_{pump})Q^2
  \]  
  (2)
The value of the fluid flow obtained at the output has to be constant in case of correct behavior of the device. Any deviation from a constant value corresponds to a system malfunction.

The possible malfunctions that can occur in the EFWS, their causes and effects on system variables are well known. They are associated with cracks in the pump or CST casing, pipe or valve ruptures, and pump or valve operation failure. All valves are normally open and locked in position.

Two of these malfunctions and their effects on system variables are:

a) crack in pipe, which causes $P_{13}$, $Q$ and $H$ to decrease faster than normally;

b) failure of any valve, which causes $P_{13}$, $Q$ and $H$ to decrease slower than normally;

As the EFWS is working only in emergency conditions, the occurrence of a malfunction in the EFWS would lead to catastrophic results.

3 Qualitative Model

Using this quantitative description, we show how we derive a qualitative model of the EFWS.

The qualitative model is realized in 3 layers of abstraction. The first layer (MONITORING) is a simplified black box view of the EFWS for monitoring. The second layer (DIAGNOSIS-1) divides the EFWS into BLOCKA (the supply part), BLOCKB (the common header) and BLOCKC (4 lines) together with the two connection modules CONNECTAB and CONNECTBC (see Figure 3). The third layer (DIAGNOSIS-2) contains the pipes, pumps and valves of the above layer. These devices are involved in the case of repair actions during malfunctions.

The component model of the EFWS is depicted in Table 1.

![Figure 3: Qualitative Model of the Seabrook NPP EFWS](see [14]). In order to be able to localize physical failures during diagnosis, we have to extend the QSIM - language [14] with respect to the integration of component-oriented information.

According to this extension, we distinguish between four classes of constraints which are described in the following section. Note that these constraints are used to denote the hierarchical model of the EFWS.

**Teleologic Constraints:** A teleologic constraint relates parameters which are controlled during the monitoring cycle in order to detect initial fault symptoms. It is quite obvious that the device itself (at its highest level of abstraction) is the component-oriented part in such a constraint. Teleologic constraints are therefore used to express the purpose of the device and build up a parameter-based system model.

**Example 1** In our example, the purpose of the EFWS is fulfilled if the output flow $Q_{out}$ remains constant. Therefore we use the following teleologic constraint:

$$\text{CONSTANT} \ (Q_{out})$$

**Tautologic Constraints:** These constraints denote relations between parameters which can not cause inconsistencies due to a component fault. We use them to denote *continuity conditions*, like conservation of flow. Clearly such a constraint is not connected to any component as it can not provide any diagnostic information. However, tautologic constraints are needed to describe the system's behavior more exactly.

**Example 2** We use a tautologic constraint to include the *continuity condition* (see [5]) for the water flow at the three...
way pipe crossroads into our model:

\[ \text{SUM.ZERO} (Q_1, Q_2, Q_3) \]

Singleton Constraints: Some constraints are explicitly related to one component. They mean restrictions concerning the behavior of a component which does not affect its environment. An inconsistency between such a constraint and the observed parameter values is therefore explained by a single fault in the concerned physical component.

Example 3: The flow \( Q^2 \) is a monotonic function of the pressure drop coefficient \( K \) of a valve \( V \):

\[ V : M^+ (K, QQ) \]

Set Constraints: This type of constraints connects at least two components. Although these components are usually situated at the same level of abstraction, set constraints are used to connect components at different levels of abstraction as well. This is a useful strategy for sophisticated focusing techniques as we have to deal with a complex hierarchical model. Clearly, an inconsistency between a set constraint and the observed parameter values can only be explained by the assumption that not all associated components simultaneously behave correctly.

Example 4: The monotonic functional dependency (which is expressed by an \( M^+ \)-constraint (see [14])) between the amount of water in the pipe-system and the pressure of water concerns both the pipes and the pump:

\[ \text{PUMP, PIPE} : M^+ (\text{PRESSURE, AMOUNT}) \]

3.2 Using Qualitative Simulation for Model Verification

It is well-known that qualitative simulation results in ambiguous envisionments which contain physically impossible and even faulty behavior patterns (see [13, 15]). In [15], we have shown that a qualitative system model which allows the derivation of faulty behavior patterns can not be used for control tasks like diagnosis and monitoring.

As a consequence, we have to perform qualitative simulation in order to check the system model for faulty behavior patterns. Only if no faulty behavior patterns can be simulated, we may use the model for monitoring and diagnosis.

In our approach, we apply the constraint propagator of QSIM to theorem-proving in the course of simulation, monitoring and diagnosis.

Having designed the qualitative model we will now turn our interest to the qualitative representation language and the inference procedures of DIAMON.

4 DIAMON

This section discusses DIAMON, a model-based monitoring and diagnosis system. A detailed presentation of DIAMON can be found in [15] or [16].

4.1 Principles

DIAMON is a model-based expert system which uses teleological parameter models for monitoring and component models for diagnosis. Thus it combines two essential tasks of dynamic system control (fault detection and fault localization). In the future we will integrate a repair component into DIAMON which uses the teleologic system model to derive optimal repair strategies. In contrast to most current approaches to model-based diagnosis which exclusively concentrate on static devices [19, 6], DIAMON is dedicated to the troubleshooting of dynamic systems which are characterized by time-varying parameters.

4.1.1 Knowledge Representation

DIAMON is based on a hybrid constraint language which integrates the parameter - constraint-oriented view of [14] and the component - connection-oriented view of [5] about qualitative reasoning about physical systems. The constraint language is an extension of the QSIM-language [14] and associates physical components with qualitative differential equations.

4.1.2 Algorithm

DIAMON is built on top of the HS-DAG - algorithm [10] which is a corrected version of the original Reiter-algorithm [19] for model-based diagnosis. The included correction now guarantees that no conflict sets can be lost due to the application of pruning strategies. We have extended the basic algorithm to deal with dynamic systems similarly to the algorithm recently presented in [17]. In the following we discuss two features of DIAMON in more detail - the diagnosis strategy and the hierarchical modeling architecture.

Diagnosis Strategy: Previous approaches to monitoring usually concentrated on the surveillance of predefined thresholds (e.g. [12]) or the detection of predictable fault conditions. We, instead, view monitoring from the model-based, consistency-preserving
point of view which shows its close relation to diagnosis. This approach allows us to deal with unanticipated fault scenarios which is a necessary condition for safety-critical applications.

Therefore, we use parameter-oriented constraints for our teleologic monitoring model and define a fault symptom to be an inconsistency between the monitoring model and the observed set of measurements.

The same concepts are also valid for diagnosis. Again we rely on the consistency-preserving approach and use the traditional notion of a diagnosis to be a minimal hitting set among the conflict sets. (see [19]).

**Dynamic Model Zooming:** Hierarchical modeling usually results in discrete model layers. Different levels of abstraction are predefined, and during diagnosis the algorithm performs discrete shifts between the different layers.

We, on the other hand, have developed a continuous strategy of dynamic model zooming. According to the diagnosis process, constraints and parameters are zoomed in and used for the continuous refinement of the already computed diagnoses.

In consequence we can easily focus on important parts of the mechanism and thus perform diagnosis efficiently and quickly.

Note that we use full sets of measurements at each level of abstraction. This is due to the fact that otherwise possible diagnoses could be missed (for an example see [16]). Although one might argue that this strategy is costly, it is inevitable if one has to control safety-critical devices like an EFWS.

### 4.2 Monitoring and Diagnosis

The combined monitoring/diagnosis cycle of DIAMON consists of five main inference steps:

- **READ MONITORING DATA:** The initial state of the monitoring cycle represents the input of the monitoring data. We do not discuss here how the real-valued input data are filtered and transformed into qualitative values (see [20]).

- **CHECK CONSISTENCY:** In a next step, the theorem prover of DIAMON checks the monitoring data and the monitoring model for inconsistencies. If no inconsistency is detected, the monitoring cycle goes back to the input phase. If an inconsistency is detected, the algorithm switches to the diagnosis cycle.

- **COMPUTE DIAGNOSES:** According to the detected inconsistency, a first set of diagnoses is computed. If the result is satisfying (e.g. the SRU-level is reached\(^3\)), a repair strategy will be applied. Otherwise the diagnosis process continues.

- **ZOOM MODEL:** According to a user-defined zooming strategy the diagnosis model is continuously refined. Additional parameters and constraints are zoomed in, the HS-DAG is adapted by the splitting of the refined conflict set nodes.

- **READ DIAGNOSIS-DATA:** The parameters of the refined diagnosis model are additionally read in. The algorithm then returns to the COMPUTE-DIAGNOSES step.

Note that if we introduce parameters during system control, they are continuously measured throughout the whole diagnosis process. This is a necessary condition to preserve the history of the mechanism.

### 4.3 Notes on Implementation

DIAMON is implemented in Common LISP and currently runs on SUN4 workstations. We use the constraint propagator of QSIM as the theorem prover for the HS-DAG-algorithm. In the present implementation, DIAMON follows a breadth-first zooming strategy for the dynamic refinement of the hierarchical diagnosis-models. However, we plan to integrate a best-first zooming strategy which will allow us to focus on more important parts of the model with respect to additional heuristic information (e.g. fault probabilities).

### 5 Experimental Results

In this part we apply DIAMON to the EFWS control problem.

In particular, we describe two possible fault scenarios in the EFWS using two qualitative models of different complexity.

#### 5.1 Example 1: A Single Fault in the PIPE-SYSTEM

A possible fault scenario concerns leaks in the pipe system. In order to demonstrate the capabilities of DIAMON, we have modeled the EFWS by means of 21 qualitative parameters and 18 constraints which mainly describe the pipe-connection CONNECT\(_{AB}\) between BLOCK\(_A\) and BLOCK\(_B\).

\(^3\)SRU stands for Smallest Replaceable Unit
Tables 2 and 3 depict the parameters and the continuing control process, showing the actual control layer and the actual diagnoses.

In state $s_1$, the decreasing tendency of flow parameter $Q_{out}$ indicates a fault in the EFWS. Additional parameters and constraints are zoomed in in order to refine the qualitative model during states $s_2$ and $s_3$. Finally, $PIPE_1$ is identified to be faulty.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Monitoring</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnoses</td>
<td>NIL</td>
<td>[EFWS]</td>
</tr>
</tbody>
</table>

Table 2: Single Fault Diagnosis - Part 1

<table>
<thead>
<tr>
<th>Layer</th>
<th>Monitoring</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnoses</td>
<td>[CONNECTAB, BLOCKA]</td>
<td>[CONNECTAB, TDF137A, V155]</td>
</tr>
</tbody>
</table>

Table 5: Multiple Fault Diagnosis - Part 2

5.2 Example 2: A Triple Fault in the EFWS

In order to demonstrate DIAMON's features in the case of multiple faults, we present a triple fault scenario. The EFWS is modeled by means of 45 parameters and 42 constraints. Tables 4 and 5 depict the parameters and the continuing control process, showing the actual control layer and the actual diagnoses.

The first fault symptom in the EFWS, the unexpected decrease of the output flow $Q_{out}$, is detected in system state $s_1$. The first diagnosis level is zoomed in and the diagnosis algorithm computes the double fault $[CONNECTAB, BLOCKA]$ in state $s_2$. In state $s_3$, the diagnosis process continues with the refined model and an additional fault symptom $(Q_{pump})$. Finally, DIAMON determines the triple fault $[CONNECTAB, TDF137A, V155]$.

Table 4: Multiple Fault Diagnosis - Part 1

6 Conclusions and Comparison

In the following we compare our approach to related research work which deals with the knowledge-based control of dynamic systems in general and with nuclear power plants in particular.

Almost all previous approaches to NPP monitoring and diagnosis use heuristically derived cause-consequence rules [2]. Thereby only predictable faults (i.e. cause-consequence associations) are covered. Conversely the model-based paradigm (e.g. [18]) also allows the detection of unpredictable faults.

Only a few model-based approaches to dynamic system control have been introduced up to now. [7] present MIMIC, a sophisticated program for model-based monitoring combining qualitative simulation and machine learning. MIMIC uses qualitative simulation of fault models to provide a decision tree for the run-time generation of fault hypotheses. In [17], the original Reiter-algorithm [19] is extended with respect to the incremental diagnosis of dynamic devices.
Our work is closely related to that of [7] and [17] and it can be seen as a continuation of their results. However, we differ from them in some important ways.

First, we use a combined approach to model-based reasoning which integrates consistency-preserving concepts for both monitoring and diagnosis. Previous approaches concentrated either on monitoring [7, 12] or on diagnosis [17]. Additionally, we use hierarchic models and apply a continuous zooming strategy for dynamic model-refinement.

Second, our fault coverage is more complete than those of [7] and [17]. In [7] faults can be missed due to the use of pre-simulated fault models which do not allow the detection and localization of unanticipated faults.

The algorithm of [17] can miss faults if the heuristically chosen incomplete sets of measurable parameters (as proposed in the example of [17]) do not represent the actual fault scenario. Additionally, the application of the Reiter-algorithm can lead to the loss of diagnoses due to faulty pruning steps (see [10]).

6.1 Pros and Cons

- (+) Due to the flexibility of the chosen model-based approach, we are able to detect and localize unexpected faults (single and multiple faults).

- (-) DIAMON can not (yet) deal with all types of non-permanent faults.

- (+) DIAMON is able to use multiple layers of abstraction for complex devices (e.g. the NPP).

- (-) The current LISP implementation is not very efficient.

- (+) Our model-based approach can easily be extended w.r.t. the integration of heuristic knowledge (fault probabilities).

7 Future Work

We have presented a successful integration of model-based monitoring and diagnosis and its application to complex dynamic systems. Nevertheless, there are still important questions which have to be solved in the future.

A very important problem concerns non-permanent abnormal behavior. For example, [3] writes

"Given that intermittents are tremendously more important to diagnose and that they do occur frequently, I propose that finding an orderly means to diagnose them is the most important unsolved problem in this area today."

Surprisingly, since then no research results which concern this problem have been published in the field of AI. All approaches to model-based diagnosis (e.g. [9, 4]) contain the implicit or explicit non-intermittency assumption about faults. We will have to tackle this problem in the future.

Measurement selection and interpretation has to be addressed in the future. Many static diagnosis systems rely on information theory (e.g. [6, 8]) to determine optimal measurements.

However, the selection of the optimal measurement points is essentially more complicated in dynamic systems where measurements are time-dependent (see [11]).

In contrast to the well understood algorithms used in model-based diagnosis, qualitative modeling has not yet reached a satisfying level. We decided to use the QSIM-language for our system because it is quite expressive. Nevertheless, the current limits of this language are a temptation to develop even more powerful modeling concepts in the future.

Acknowledgements

We especially thank Benjamin Kuipers who has provided the Common LISP-implementation of QSIM (Q2) and several new technical reports of his research group. DIAMON was developed together with Wolfgang Nejdl and Alois Haselböck.

References


