Fault Diagnosis of a Sewage Plant

J. Schönwälder M. Hofmann H. Langendörfer
TU Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Bültenweg 74/75
D-3300 Braunschweig

Abstract

In this paper we present a project whose aim is the development of an expert system managing and diagnosing a sewage plant. After a short description how the knowledge acquisition process took place we will explain why the popular model-based diagnosis approach cannot be applied to our problem domain. Instead we had to consider associative knowledge to solve the diagnostic problem. In order to adequately express knowledge about the structure of the sewage plant, knowledge about well understood subprocesses and associative knowledge for the diagnosis of the sewage plant we designed the tool $MOTES_{DM}$ that supports hybrid knowledge representation. $MOTES_{DM}$ allows us to separate associative knowledge from structural knowledge concerning the technical system.

AI topic: knowledge representation, diagnosis

Domain area: monitoring of sewage plants

Language/Tool: Smalltalk-80, MOTES_{DM}

Status: running prototype

Impact: The system will guarantee the correct behavior in small- and medium sized treatment plants having no engineers as members of their staffs.

1 Introduction

Sewage plants using a mechanical and a biological purification stage are locations where rather complex biological and chemical processes take place. Very often, the detection of the cause of a failure can only be done on the background of expert knowledge, while the removal of the failure's cause can be carried out by the filter plant's staff. Since in small- and medium sized plants a civil engineer is usually not a member of this personnel due to the high costs of his or her employment, a lot of time oftenly passes until the cause of a failure is detected.

Main goal of our project is the development of a system that can be used on sewage plants to support the plant's staff in the following areas:

- Analyzing incoming data.
- Presenting data in graphical form.
- Filtering measurements for input failures.
- Diagnosing possible causes for failures based on expert knowledge.
- Proposing ways to handle failures.
- Controling the daily routine of the plant's staff.

The development takes place in three main steps. Firstly, we acquired knowledge from our expert (section 2) and we thought about the possible use of model-based diagnosis (section 3). The second main step was the development of a prototypical system. The tool being the foundation of this prototype is described in section 4, while the prototype is introduced in section 5. The third main step is the development of the target system. At the moment, this system is being implemented using the insights we gained by our prototype.

2 Knowledge Acquisition

The project began with a knowledge acquisition phase. We used interviewing techniques in their structured and unstructured form [4]. Our expert is a colleague of the Institute for Sanitary Engineering at our university.

In common sessions the expert described the problem domain. The knowledge engineer interrupted the monologue of the expert to make comments and to point to missing details at the end of each session. The knowledge engineer asked questions about these issues in following sessions. Later we used structured interviewing techniques to study some issues in more detail.

The result of these activities was a heap of knowledge and different knowledge sources that had to be structured in the second phase of our knowledge acquisition process.

We believe that this structuring task is one of the crucial parts in a knowledge acquisition process.

Unfortunately we had to manage lots of knowledge elements stored in many different knowledge sources. So some data measured by our colleague have been stored in a database. The measures taken at the sewage plant are written on paper rolls by an analogous writer. Some data are included in a book to allow the local government to check the keeping of environmental laws. The structure of the filter plant is accessible on a blue-print. The distribution of data and the various elements of knowledge caused severe problems in special stages of the realization of our project. We decided therefore to investigate the facility of an integrated management of knowledge sources and support of knowledge acquisition methods by using a hypertext system [3].

The result of the knowledge acquisition process is a data model on the knowledge level. In this model we distinguish system objects (components and aggregates), events and symptoms, contexts and goals, hypothesis, actions, definitions, parameters and diagnosis. This model is used to derive the final knowledge base and to supply help for future alterations.

3 Model-Based Diagnosis

First generation diagnostic expert systems were build by representing empirical associations in a set of rules. The limitations of these systems have led to the development of model-based diagnosis. After publication of the General Diagnostic Engine (GDE) [2] a tremendous number of attemps have been made to apply this approach to various domains and to identify and develop necessary modifications and extensions [7].

In such systems the model of the artifact is used to predict the correct behavior of the artifact. The diagnostic approach is based on characterizing model-artifact differences as assumption violations in the model. Therefore one needs utmost complete knowledge about the structure and the behavior of the system in order to build a significant model that can be used to infer faults from assumption violations. As a consequence, model-based diagnosis can be applied to man-made systems where at least one expert knows what specific components are involved and how they are meant to interact in order to achieve the functionality of the whole system. Thus model-based diagnosis is extremely useful in well understood technical systems such as electric circuits or mechanical machines.

Sewage plants are man-made systems that are designed upon experiences made with former sewage plants. A complete description of the purification processes is not available since there are numerous influences on the purification processes and relationships between these processes. Research on sewage plants usually tries to identify biological or chemical subprocesses and to develop models for such subprocesses. Up to now these models can not be combined to give a complete model of the behavior of a sewage plant. Therefore we cannot directly apply the model-based approach to our problem domain. Instead we have to consider associative knowledge in our system.

So we decided to build a system that allows us to adequately express knowledge about the structure of the sewage plant, knowledge about well understood subprocesses and associative knowledge for the diagnosis of the sewage plant. We believe that for many tasks in the domain of environmental protection the integration of model-based and heuristic diagnosis is needed.

4 The Tool MOTES$_{DM}$

The tool MOTES$_{DM}$[1] which runs under Smalltalk-80 forms the basis of our prototype. It supports hybrid knowledge representation so that associative diagnostic knowledge can be separated from structural knowledge about the technical system. The knowledge representation formalisms make use of object-oriented concepts such as class membership and inheritance that allow a modular design of the knowledge base.

Figure 1: Concept of MOTES$_{DM}$

Figure 1 shows the concept of MOTES$_{DM}$. Measurings taken at the technical system are transmitted to the model of the system. The model collects this data and derives new data using a constraint network that defines the correct behavior of the technical system. Faults in the technical system enter the model in form of data that do

[1] Modelling Technical Systems for Diagnosis and Monitoring

not fit with the correct behavior of the system. The propagation of such data causes contradictions in the constraint network. These contradictions activate a rule interpreter that uses associative knowledge encoded in rules to find the fault.

4.1 Component Description

The model used by MOTES_{DM} is based on a description of the structure of the technical system. The model is not only used to store data concerning the state of the system but it supports also the propagation of constraints as well as the interpretation of the system's state by the rule interpreter.

The component description language of MOTES_{DM} distinguishes between components, component classes, aggregates and aggregate classes:

- Components represent elementary objects of the technical system. The state of a component is given by its attributes.
- Component classes are used to describe similar objects. Component classes can be specialized by making one component class a subclass of another component class. The subclass inherits all properties of the superclass.
- Aggregates are compound objects. They may enclose other components and aggregates.
- Aggregate classes describe similar aggregates. Aggregate classes can be structured hierarchically analogous to component classes.

The component description incorporates three relationships. Between an aggregate and its parts exists a *part_of* relationship, between a component or an aggregate and its class exists an *instance_of* relationship, and between hierarchical arranged classes exists a *specialization_of* relationship. Thus we can use the abstraction mechanisms aggregation, generalization and specialization that are of great importance in any system that models relationships between (technical) objects [1][6].

The mentioned relationships allow the inheritance of attribute specifications (data type, when and how a system may ask a user, etc.) and constraint definitions.

4.2 Constraints

Constraints are a useful formalism for describing the behavior of components and aggregates. A constraint consists of a set of attributes and a relation upon these attributes. In MOTES_{DM}, we use constraints to compute values for unknown attributes from a set of given attributes and to test the consistency of values.

By sharing attributes among different constraints we get a constraint network in which we propagate data received from the technical system. There are a lot of known propagation techniques for constraint networks with different properties [5]. Currently we use the value propagation technique. The advantage is that we can handle contiguous variables and that every local consistent solution is also global consistent. The drawback of this method is that not all global consistent solutions are detected.

Following our object-oriented approach we distinguish between constraints and constraint classes. A constraint class defines a generic relation between abstract attributes. Instances of this class establish this relation upon a set of concrete attributes. The introduction of this concept allows the inheritance of constraint definitions along *specialization_of* and *instance_of* relationships.

Further we distinguish between predicate constraint classes and constructive constraint classes. Instances of constructive constraint classes allow the computation of unknown attribute values from known values. Constructive constraint classes are defined by a set of functions that calculate unknown values. Instances of predicate constraint classes can not be used to derive unknown attribute values. Instead, they are used to describe relations that may hold or not. They are defined by an expression that returns a logical value.

An important property of our constraint network is that new data may enter and overwrite existing data in the model of the technical system. When such an event occurs, we expect MOTES_{DM} to guarantee that all affected attributes are updated. Therefore every attribute value is stored with a time stamp, a set of assumptions used to compute the value and a set of attributes depending directly on the value. These informations are used to collect all attributes depending on a specific attribute value at a given time, to update these attributes and to reevaluate all affected constraints.

4.3 Rules

The associative knowledge for model state interpretation is represented in rules. Every rules is attached to a context. A context is a set of rules with an additional predicate that defines if the context is active or not. Only rules in active contexts are considered by the rule interpreter. Different levels of abstraction can be expressed by making one context a more specific context of another context. In general, one can define a hierarchy of contexts. A rule in a more specific context becomes active, if its more general contexts are active and if the predicate of the context evaluates to true.

Rules consist of a a premise and a conclusion. The premise is a logical expression that defines when a rule can be applied. The conclusion will be evaluated if the context of the rule is active and the premise of the rule evaluates to true. Thus our current rule interpreter is forward chaining.

5 The Prototype

We used MOTES_{DM} to implement the knowledge that has been comprehended during the knowledge acquisition process. The foundation of the knowledge base is a structural description of the sewage plant in Salzgitter-Bad that is assembled out of 136 components and aggregates that are instances of 55 classes. On top of the component description we defined constraints that describe dependencies between water and sludge quantities in the sewage plant, define the rotor steering mechanism in the biological basin, describe chemical properties of sewage and constraints that establish stability conditions for biological purification processes.

The knowledge about ammonia concentrations and effluent sludge is packed in suitable contexts and appertaining rules. We tested our prototype using data that were measured on the sewage plant Salzgitter-Bad during summer 1989. The ability to write Smalltalk-80 processes allowed us to simulate the asynchronous behavior of the measuring instruments.

With the development of this prototype, we have reached two goals: On the one hand we could demonstrate that our approach to use artificial intelligence techniques is reasonable in the present task. The separation of structural knowledge from associative knowledge combined with the use of object-oriented concepts that increase abstraction allowed us to modify the knowledge base straightforwardly. On the other hand the prototype helped us to find gaps in the knowledge base.

6 Conclusion

The development of our prototype finished. According to the insights we have gained by the operating prototype, we have designed the target system. This system has to gain the acceptance of the sewage plant employees. Therefore we are now developing a suitable user interface that is easy to use. Another important point is the integration of an existing database. However, often it is the appearance of a software system that decides if it will be used in the long term.

References

[1] D.S. Batory and A.P. Buchmann. Molecular Objects, Abstract Data Types and Data Models: A Framework. In *Proc. VLDB '84*, Singapur, 1984.

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

[3] M. Hofmann, U. Schreiweis, and H. Langendörfer. An Integrated Approach of Knowledge Acquisition by the Hypertext System CONCORDE. In *Proc. ECHT '90*, Paris, November 1990.

[4] A.L. Kidd, editor. *Knowledge Acquisition for Expert Systems.* Plenum Press, 1987.

[5] P. Meseguer. Constraint Satisfaction Problems: An Overview. *AI Communications*, 2(1), March 1989.

[6] D.C.P. Smith and J.M. Smith. Database Abstractions: Aggregation and Generalization. *ACM Transactions on Database Systems*, 2(2):105–133, June 1977.

[7] P. Struss. Model-Based Diagnosis – Progress and Problems. In *Proc. Wissensbasierte Systeme*, Informatik Fachberichte 227, Springer, 1989.