Working Group in Model-based Design and Reasoning Part I: Modelling and Diagnosis

Boi Faltings
Laboratoire d'Intelligence Artificielle,
Swiss Federal Institute of Technology (EPFL),
1015 Lausanne, Switzerland,
faltings@lia.di.epfl.ch

Abstract

Classical Artificial Intelligence techniques such as expert systems have often been found to be too brittle for large-scale applications. Model-based reasoning is a technique for making Artificial Intelligence software applicable to problems of realistic size.

In this working group, we have investigated some fundamental issues in model-based reasoning and various applications in diagnosis, control and design. This paper reports on the results in basic techniques as well as diagnosis applications obtained at EPFL, ETHZ, Landis & Gyr and Nestle. The results of the part dealing with design are reported in another paper in this volume.

Figure 1: *Example circuit and measurements.*

1 INTRODUCTION

Conventional rule-based Artificial Intelligence software suffers from two important deficiencies. The first is that a rule-based knowledge representation does not separate procedural and declarative knowledge and is therefore difficult to apply to large-scale problems. The second is that it integrates poorly with existing software.

Model-based reasoning is a technique for solving both problems. Knowledge is represented in the form of declarative models similar to those already in use in conventional software such as CAD systems. Reasoning engines are specific to and optimized for particular tasks, such as diagnosis, configuration or design.

To illustrate the idea behind model-based reasoning, consider the task of diagnosing a simple digital circuit, shown in Figure 1. The diagnostic task is to infer diagnostic *candidates*, i.e. combinations of possibly faulty components, from the measurements made on the circuit. A rule-based expert system would consist of deductive rules such as:

$$\text{incorrect}(F) \wedge \text{incorrect}(G) \Rightarrow \text{faulty}(M2) \text{ CF} = 0.5$$

where **CF** is a *certainty factor* meaning that the rule is valid with about 50 % certainty. Such rules are usually obtained from observation of a human

expert and combine knowledge about device structure, diagnostic strategy and likelihood of failures into one model. The diagnostic problem is solved by applying the rules to the measured data until a sufficiently certain conclusion is asserted.

In model-based reasoning, knowledge about the device would be expressed by *models*, for example:

$$\mathbf{multiplier}(m, x, y, z) \wedge \neg \mathbf{faulty}(m) \Rightarrow \mathbf{equal}(x * y, z)$$

stating that if a multiplier m has inputs x and y , output z and is not faulty, then $z = x * y$. The diagnostic problem is now an *abductive* problem: find the set of faulty components such that the relations implied by the model are consistent with the observations. Usually, the initial observations are not sufficient to allow only a *single* solution to the abductive problem. For the device shown in Figure 1, the measurements

Inputs: $A=B=C=2$, $D=E=3$
Outputs: $F=8$, $G=10$

allow in fact many different diagnostic candidates which can be obtained through the following reasoning. Since F depends on A, B, C and D through components M_1 , M_2 and A_1 , and the result does not agree with the simulation, at least one component in the set $\{M_1, M_2, A_1\}$ must be faulty. Similarly, since G is also incorrect, at least one of $\{M_2, M_3, A_2\}$ must also be faulty. A diagnostic candidate is a combination of components which contain at least one element of the first set and one of the second. For example:

$$\{M_2\}, \{M_1, M_3\}, \dots, \{M_1, M_2, M_3, A_1, A_2\}$$

are all candidate diagnoses for the circuit.

In order to narrow down this set of candidates, a model-based diagnosis system requires additional processes to *rank* diagnostic candidates, and to decide on a *strategy* for taking measurements. Since these are independent of the particular system to be diagnosed, they are more properly included in a general diagnostic *engine*. Researchers in model-based reasoning have developed many such engines: GDE, GDE+, SHERLOCK, ARTIST are some of them. Some of them are in practical use already: SHERLOCK for photocopiers, and ARTIST for power systems and process control in the chemical industry.

In the example of Figure 1, the initial ranking would rate $\{M_2\}$ as more likely than the combination $\{M_1, M_3\}$, since it is more likely to have a single fault than multiple ones. The next measurement proposed by an optimal strategy would be to test y . Since this measurement again turns out to be incorrect, we now know that M_2 must be faulty, which also explains both other faults. Thus, we can see that diagnosis can be performed by a logical procedure which is clearly separated from the model of the device to be diagnosed. This separation of reasoning strategy and problem-specific knowledge is common to all model-based reasoning systems.

Model-based reasoning offers many advantages with respect to rule-based expert systems. Since models can be formulated in a systematic way, they can be obtained automatically from design descriptions. It is not necessary to perform “knowledge engineering” on experts, in fact, one can construct model-based diagnosis systems for completely new designs whose behavior is not known yet. Models can also be shared with other programs and taken from databases where they are automatically updated whenever the problem definition changes. Where signature-matching techniques require modeling the behavior of faulty components, which is usually impossible, in model-based diagnosis only the *correct* behavior has to be modelled. Finally, strategies for problem-solving can be formulated and optimized independently of a particular application, thus significantly reducing the cost of constructing individual systems.

Implementing model-based reasoning systems poses the following difficulties:

- it involves abductive reasoning, a combinatorial problem. Techniques of constraint satisfaction and assumption-based reasoning can help cope with this complexity.
- it requires *causal* device models, where properties of results can be traced to properties of components.
- it sometimes requires *qualitative* models, either because measurements are only qualitatively accurate or because no better models are known.

In this working group, we have addressed several of these problems, both relating to the basic techniques and their applications to practical problems.

Figure 2: *An example of a qualitative region structure which might model, for example, a temperature distribution.*

2 TECHNIQUES FOR MODEL-BASED REASONING

2.1 Modeling distributed phenomena

Many physical domains involve parameters which are distributed in space. For example, modelling weather systems, forest fires, and water runoff all involve distributions of continuous parameters. Currently, distributions are approximated using grids of numerical values. However, many of these systems involve inexact measurements, and some even exhibit chaotic behavior. It is difficult if not impossible to predict the effect of measurement uncertainties on the results computed using such models. However, at a higher level of abstraction, such systems are remarkably stable and predictable. The goal of this research is to develop a *qualitative* formalism for modelling such distributions.

In a *qualitative* model, parameter values are modelled not by real numbers, but by intervals. A common set of intervals is $\{[-\infty..0), 0, (0..\infty]\}$, i.e. the sign of the value. Another one which may be useful for the temperature of water would be $\{[-\infty..0), 0, (0..100), 100, (100..\infty]\}$. When parameters are distributed in space, the interval structure divides the space into *regions*

within which the parameter has the same qualitative value. Figure 2 shows an example of such a region structure. Reasoning with qualitative models now implies reasoning about the shape of such regions and their interaction.

Within the context of the project, Monika Lundell of the EPFL has investigated the problems of how to represent regions and reason with them. She first investigated an approach to shape representation based on the curvature of the regions ([5]). However, it turned out that the curvature information is much weaker than what seemed to be the case from the published literature. Furthermore, the shape that is present given a set of measurements can only be determined in a very arbitrary fashion. Due to these drawbacks, she switched to a triangulation between fixed points of measurements as a qualitative representation of the distribution ([6]).

Reasoning about the behavior of a distributed parameter system now requires manipulating the triangulations. The equations and differential equations describing the underlying physical process define relations between the qualitative parameters. These relations hold equally for all points within a qualitative region. Reasoning thus proceeds in three steps:

1. intersect the topological maps to find all combinations of qualitative parameter values which occur simultaneously somewhere in the space,
2. for each combination, apply the qualitative relation to determine the future evolution, and
3. generate new topological maps for each parameter, merging adjacent regions for which identical predictions were made.

Metereologists have studied similar approaches in the 1950's before numerical simulation methods became popular. There thus exists a large body of literature on how to model meterological phenomena using regions, and we are using this as an example to guide our research.

2.2 Cumulative logic programming

A fundamental difference between model- and rule-based reasoning is that where rule-based reasoning uses *deduction*, a very well-understood process, model-based reasoning is based on *abduction*. The difference can be illustrated using the following example. Consider the facts:

bird(Tweety)
flies(Tweety)

and the rules:

$(\forall x) \text{ bird}(x) \Rightarrow \text{flies}(x)$
 $(\forall x) \text{ plane}(x) \Rightarrow \text{flies}(x)$

In *deduction*, we use the rule to infer **flies(Tweety)** given **bird(Tweety)**. In *abduction*, we infer $(\text{bird(Tweety)} \vee \text{plane(Tweety)})$ from **flies(Tweety)**. An abductive inference is correct only under a *closed-world assumption*: intuitively, the conclusion that Tweety flies is correct only under the assumption that the known rules models all possible cases where something can fly.

One of the challenges in model-based reasoning is to formulate knowledge in a sufficiently structured way that abduction is feasible. Gerhard Schwärzler and Christian Hidber of the Mathematics Department at ETH Zürich have developed a formalism called *cumulative logic programming* (CuLP) which allows such modeling ([1]). CuLP is an algebraic modeling language for real-world entities. In CuLP, a system is modelled as a collection of objects and processes. Object represent components and processes their relations.

The objects and components are formalized using a graph algebra which allows applying algebraic principles to find algorithms to solve CuLP programs as well as prove their properties.

A compiler for translating CuLP programs into PROLOG has been implemented in the course of the project. The compiler also provides a C interface which allows connecting the symbolic algebra package Mathematica as well as other database and visualization packages.

3 APPLICATIONS OF MODEL-BASED REASONING

3.1 Diagnosis with models

Automatic fault diagnosis is an important application area for intelligent systems. Diagnosis is clearly an abductive problem, since the task is to find the reason for an observed failure rather than its effects. Furthermore,

Figure 3: *Example of an HVAC system studied in the project.*

diagnosis requires large amounts of knowledge about the system to be diagnosed; using a model-based technique allows bypassing the knowledge acquisition problem.

Within the context of this project, Landis & Gyr has investigated qualitative techniques for diagnosis of Heating, Ventilation and Air Conditioning (HVAC) systems. An example of a plant studied in the project is shown in Figure 3.

Faults in such systems can be:

- physical damage to components, for example leaves which block air vents,
- user behavior, for example putting a piece of furniture in front of a temperature sensor, or

- installation or even design errors, for example a sensor which is not connected correctly.

Besides causing discomfort to the occupants of the building, such faults can also significantly raise energy consumption. At the same time, they are difficult to detect and localize since they are often detectable only by comparing behaviors under different climatic conditions. There is thus considerable interest in diagnosing and correcting such faults automatically. The problem is an interesting application for model-based reasoning since every installation is different and the use of conventional expert systems would require formulating expert knowledge specifically for every installation.

Within the project, L. Fornera, A. Glass, P. Gruber and J. Tödtli of Landis & Gyr Business Support AG have developed qualitative models of HVAC systems. They have furthermore developed numerical simulation models which allow detecting fault symptoms required for subsequent diagnosis. The models have been integrated in a fault detection system using the “controller-landmark” method developed in the project ([2, 4]). It consists of a *steady-state* detector for determining whether or not an HVAC system has reached a steady-state, and a diagnostic module which is capable of comparing the temperature measurements and the observed controller outputs with the behavior that would be expected given this steady-state. The diagnostic module is based on *qualitative* reasoning to minimize the effects of measurement errors.

Esther Gelle of EPFL has applied the model-based diagnosis approach of the *general diagnostic engine* (GDE) to the example shown in Figure 3. She implemented demonstration systems both for qualitative and constraint-based numerical models. The work has shown that GDE in combination with qualitative models is sufficiently powerful for diagnosing most faults ([3]). However, this work was only based on simulated data so it is not clear what would happen if it were run on an actual system. Insufficient funding prevented us from further pursuing this work.

3.2 Diagnosis with models and cases

For many technical systems, there are no accurate numerical models available. Engineers are forced to operate with rules of thumb. These rules of thumb can often be formulated as *qualitative* models which specify the main

parameters and relations between their signs, but not the details of the functions which govern these relations. In certain cases, the conclusions which can be drawn on the basis of qualitative models alone are sufficient, but very often *numerical* estimates are still required.

Qualitative models can in fact be used to derive quantitative predictions by combining qualitative reasoning with *case-based reasoning*. The use of the qualitative models is now not to draw the final conclusions, but only to locate relevant precedents, called *cases*. Cases are examples of combinations of precise numerical values which have been observed previously in the system behavior. The set of precedents found to be relevant by the qualitative model defines limits for the numerical values of certain system parameters.

In collaboration with Peter Duxbury-Smith of Nestle-YORKRECO, Bradley Richards of the EPFL has applied this methodology to diagnosis of a certain type of coffee roasters which Nestle has found very difficult to control. The system we have developed and implemented predicts future roaster behavior as well as some key parameters of the resulting product on the basis of current and past measurements. We consider two ways in which a qualitative model can be used to select relevant cases:

- the qualitative model can define a similarity metric which compensates for the distortions caused by dependencies between parameters. Predictions are then taken from a *single* case which is found to be most similar to the current measurements ([8]).
- monotone relations in the qualitative model can show which cases can be used to set lower and upper bounds on predictions. Let X be a parameter which has been measured to be $X = x_{meas}$, and Y the parameter which has to be predicted. If Y is monotonically increasing with X , then a case where $X = x_{case}$ and $Y = y_{case}$ sets a lower bound $y_{case} \leq y_{real}$ whenever $x_{case} \leq x_{meas}$, and an upper bound $y_{case} \geq y_{real}$ whenever $x_{case} \geq x_{meas}$. The greatest lower and the least upper bounds determine an interval within which y_{real} must be located.

The methods have been tested on simulated examples and have shown very promising results. We are just completing implementation of a prototype which will be tested by operators of an actual coffee roaster and compared with real measurements.

3.3 Control

Fuzzy logic has become very popular in the automatic control community in recent years. Its main advantage is that it allows controlling systems which are only incompletely modelled using heuristic *rules*. Since the expert systems paradigm used by fuzzy control has had such enormous success in automatic control applications, the obvious question is whether model-based reasoning will be advantageous in such applications as well.

Within the project, Eric Sauthier of the EPFL has developed and implemented a technique where control consists of 3 components ([9]):

- *monitoring* the system to detect deviations from the desired behavior,
- *diagnosing* such deviations to determine their causes, and
- *repairing* the current control regime based on the diagnosis obtained.

The model-based diagnosis algorithms described earlier can then be directly applied to the monitoring and diagnosis tasks. The repair task can be integrated with diagnosis by formulating a diagnosis in terms of the repairs required to fix the problem. This strategy is straightforward as long as there is in fact a solution, i.e. a way to control the system which corrects all current problems.

A prototype application of this approach to control of urban traffic has been implemented and tested on data taken from a section of Lausanne. The results have been encouraging in that traffic throughput in a simulation has been significantly higher than with the control currently used. However, since there are no accurate simulation models for urban traffic, these results are not conclusive. Because of the difficulty involved in modeling traffic, we have since developed our approach on a more abstract level.

Complications arise when there is no solution which corrects *all* problems, but only solutions which correct a subset of them. Selecting one of them amounts to a *multi-criteria optimization* problem which is very difficult to solve. In this case, the best one can do is to find a *pareto-optimal* solution. We have implemented the condition of pareto-optimality using *minimal environments* in assumption-based truth maintenance systems (ATMS) and thus found an efficient computational mechanism for computing and representing all pareto-optimal control solutions.

Another problem is that often control cannot correct problems instantaneously, but requires planning a temporal *sequence* of states. For example, controlling an inverted pendulum often requires sequences of very different control actions to achieve a simple motion. In fuzzy control, such strategies are explicitly coded in rules. In model-based reasoning, they have to be inferred using *simulations* of the device. We are currently investigating how qualitative simulation can be used for planning sequences of control actions.

4 CONCLUSIONS

Artificial Intelligence suffers from the fact that the potentials of rule-based expert system technology had been largely oversold. In fact, rule-based systems are a very poor way to formulate knowledge, as they do not provide the separation of procedural and declarative knowledge which is expected of knowledge-based systems. Therefore, expert systems have become just another programming technique which suffers from the same problems as most others.

Model-based reasoning is a technology which makes it possible to achieve a clean separation of domain- or problem-specific knowledge in the form of models on the one hand, and strategies for its application on the other. Models can often be obtained directly from design descriptions, or at least formulated in a principled manner, thus bypassing the knowledge acquisition bottleneck.

Problem-solving strategies can be encoded into task-specific *engines*, for example for diagnosis or design, and optimized to solve it in the most efficient manner possible. In particular, problem-solving engines can be coded once for a multitude of tasks and require no significant adaptation in order to be used for another application of the same kind.

These characteristics make model-based reasoning a successful technique which actually delivers on most of the expectations which had been placed in knowledge-based systems technology. The techniques developed in this project are now finding their way into applications: collaboration with Landis & Gyr continues through consulting by EPFL researchers, and continuation project funded by Nestle and the CERS (Commission pour l'encouragement de la recherche scientifique) has been started.

Acknowledgment

The author would like to acknowledge the members of the SPP project No. 5003-034269 for their contributions reported in this paper: Eric Sauthier, Monika Lundell, Bradley Richards of the EPFL Artificial Intelligence Laboratory; Linda Fornera, Andrew Glass, Peter Gruber and Jürg Tödtli of Landis & Gyr; Gerhard Schwärzler and Christian Hidber of ETH Zürich; and Peter Duxbury-Smith of Nestle YORKRECO.

References

- [1] **E. Engeler, G. Schwärzler:** “An Algebraization of Hierarchical and Recursive Distributed Processes,” in E. Engeler (ed.): *The Combinatory Programme*, pp 73–92, Birkhäuser, 1994.
- [2] **L. Fornera, A. Glass, P. Gruber, J. Tödtli:** “Qualitative Fault Detection Based on Logical Programming Applied to a Variable Air Volume Air Handling Unit,” *2nd IFAC Workshop on Computer Software Structures*, Lund, August 1994
- [3] **E. Gelle, B. Faltings:** “Diagnosis of Heating, Ventilation and Air Conditioning Systems,” *Proceedings of SPICIS '94*, Singapore, November 1994
- [4] **A. Glass, P. Gruber, M. Roos, J. Tödtli:** “Preliminary Evaluation of a Qualitative Model-based Fault Detector for a Central Air-Handling Unit,” *Proceedings of the 3rd IEEE Conference on Control Applications*, Glasgow, August 1994
- [5] **M. Lundell:** “Spatial Ambiguity in a Qualitative Curvature-Based Representation of Single Shapes,” Working Paper SPP Project No 5003-034269, 1993
- [6] **M. Lundell:** “Qualitative Reasoning with Spatially Distributed Parameters,” *International Workshop on Qualitative Reasoning about Physical Systems*, Nara, Japan, 1994

- [7] **M. Lundell**: “Model-Based Reasoning about Spatially Distributed Natural Phenomena,” *Workshop on Artificial Intelligence in Agriculture and Natural Resource Management*, AAAI, Seattle, 1994
- [8] **B. Richards**: “Qualitative Models as a Basis for Case Indices,” *Proceedings of the 2nd European Workshop on Case-based Reasoning*, Chantilly, November 1994
- [9] **E. Sauthier, B. Faltings**: “Model-based Control”, *Proceedings of the 11th European Conference on Artificial Intelligence*, Wiley, **1994**