Proceedings of AILS 04:
Second Joint SAIS/SSLS Workshop
April 15-16, 2004, Lund, Sweden

VARIABLE CAUSALITY FOR DIAGNOSTIC ALGORITHMS

Fredrik Dahlstrand **Jan Eric Larsson** **Bengt Öhman**

GoalArt, Tunavägen 39C, 223 63 Lund, Sweden
E-mail: info@goalart.com, URL: www.goalart.com

ABSTRACT

Model representations for use in diagnostic reasoning usually describe causality, between parameters, faults, or process states. In this way, it is possible to perform reduction of alarm showers, fault diagnosis, and other diagnostic tasks. However, the causality of a target system may vary depending on details in the construction, as well as over time with the process state. This paper presents a general method of describing varying causality in a simple and efficient way. The method has been tested using multilevel flow models.

INTRODUCTION

Causality is a central concept in diagnostic model representations. It is used in deduction on whether one fault depends on others or is a primary (or root) cause of a fault situation. For example, consider the simple lab process shown in Figure 1.

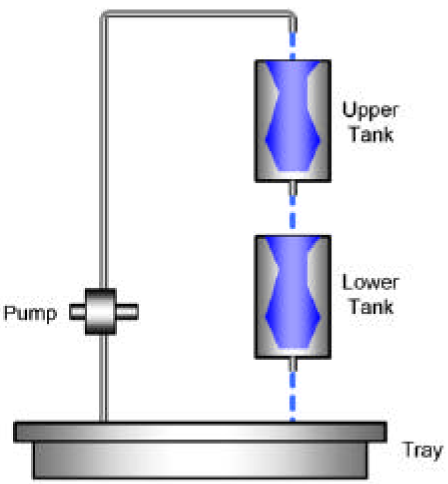

Figure 1. The open tanks process.

In this process, water is pumped from a large tray tank, up into a small, upper tank with a hole in the bottom. From the upper tank, the water flows down into a lower tank, and from there back into the tray tank. The purpose of this process is to train students in basic control theory, by allowing them to try different methods of controlling the level of either the upper of lower tank.

A simple example of a fault situation would be if the pump stopped. This would cause a low volume in the upper tank, which in turn would cause a low volume in the lower tank. There would be three faults, whereof one primary (the pump low flow) and two consequential (the low volumes in the tanks). By modeling the causality of the process, a diagnostic algorithm could reduce the three-alarm situation into one primary alarm.

Now consider a similar process, shown in Figure 2. The difference here is that the two smaller tanks are closed and connected to each other and the surrounding equipment with closed pipes.

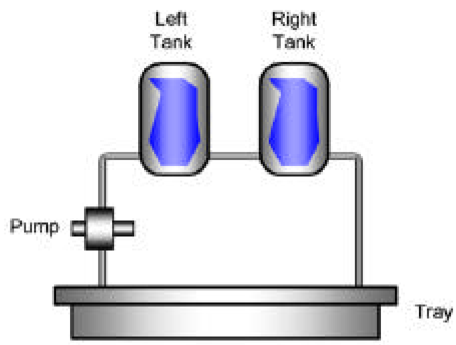

Figure 2. The closed tanks process.

The same analysis as above is valid for this process too. However, the two processes differ in several other aspects.

For example, assume that the right tank is completely full. In the closed tanks case, this high volume will block up the system and cause a high volume in the left tank and a low flow through the pump. In the open tanks case, there is no such "backward" causation. Instead the water will run over the edge of the lower tank, while the upper tank and the pump are unaffected.

A way of handling this problem is to describe the two systems using different kind of modeling entities, with different causal properties. For example, it would be possible to envision a modeling language with concepts such as open and closed tanks, open and closed connections, pumps that are affected by blocks, pumps that are strong enough to force fluid through blocks, etc. The drawback of such an approach is obvious. There will be a need for a very large number of objects, often similar, but with slight differences in causal properties.

Assume further that we have a system of closed tanks with lids, which can be open or closed. Then different causality is present depending on the state of the lids. In another case, the tanks may have an overflow protection, which is either working or broken. Tanks may have underflow protection or not, batteries may have automatic recharging or not, etc., and these may depend on time, operating state, fault status, and so on.

This paper presents another solution to this kind of problem, namely letting the connection between the objects contain information on which causal connections are active at a certain moment. In this way, a small number of modeling objects are sufficient, and the model can easily be given time-dependent behavior.

MULTILEVEL FLOW MODELS

The method presented has been implemented and tested using one specific modeling representation, multilevel flow models (MFM). MFM are graphical models of goals and functions of technical systems. The goals describe the purposes of a system or subsystem, and the functions describe the capabilities of the system in terms of flows of mass, energy, and information. MFM also describes the relations between the goals and the functions that achieve those goals, and between functions and the subgoals, which provide conditions for these functions. MFM was invented by Morten Lind at the Technical University of Denmark, see Lind (1990 a). Several new algorithms and implementations were contributed by Jan Eric Larsson at Lund Institute of Technology, see Larsson (1992, 1994, 1996, 2002).

MFM provides a good basis for diagnostic algorithms. The work of Larsson (1996) describes three algorithms based on MFM: measurement validation, alarm analysis, and fault diagnosis. Other algorithms have been developed later, such as fuzzy alarm analysis, see Dahlstrand (1998), Dahlstrand (2000, 2002), Larsson and Dahlstrand (1998), failure mode analysis, see Öhman (1999, 2001), and sensor fault detection, see Öhman (2001, 2002).

AN EXAMPLE OF AN MFM MODEL

MFM has been thoroughly explained in Lind (1990 a) and Larsson (1992, 1996). Here a small example will be given, to show the basics of MFM modeling. We will use the open tanks process of Figure 1. It should be noted that the open tanks process is just a toy process; MFM is usually used to model large complex systems, for example, a nuclear power plant or the human body.

Each flow in an MFM model is described by flow functions. There are nine different flow functions defined in MFM. Of these nine functions only four are needed to describe the open tanks process.

- The *source* function represents a component's capability of providing mass, energy, or information. The graphical representation of the source is the leftmost symbol labeled "Tray" in Figure 3.
- The *transport* function represents a component's capability of moving (transporting) mass, energy or information. The graphical representation of the transport is the symbol labeled "Pump" in Figure 3.
- The *storage* function represents a component's capability of storing mass, energy, or information. The graphical representation of the storage is the symbol labeled "Upper" in Figure 3.
- The *sink* function represents a component's capability of draining (consuming) mass, energy, or information. The graphical representation of the source is the rightmost symbol labeled "Tray" in Figure 3.

An MFM model of the tanks process is shown in Figure 3. The tray tank is described by a source function, the pump by a transport, the upper and lower tanks by storages, the gravitational outflows by transports, and the tray tank also by a sink function. In addition, the electrical power supply is shown in a sub-flow, which is a condition for the pump to move water. MFM functions in the same flow are grouped together in networks, the networks are shown as rectangles surrounding the functions in Figure 3. The upper network represents the water flow, and the lower network represents the flow of electricity.

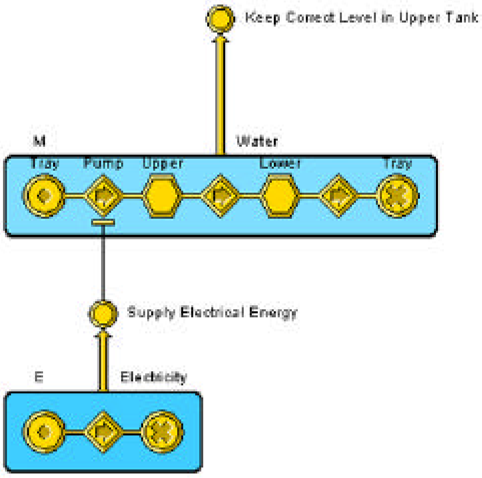

Figure 3. An MFM model of the tanks system.

Each of the two networks is connected to a goal. The goal of the water flow is to "Keep Correct Level in Upper Tank" and the goal of the lower network is to "Supply Electrical Energy." It should be noted that MFM describes how different flows enable each other. In the simple example in Figure 3, it can be seen that the electrical energy flow is a necessary condition for the pump and thereby for the water flow through the process.

Exactly the same MFM model, as the one shown in Figure 3, is used to describe the closed tank process.

For the rest of the paper only the network describing the water flow will be discussed.

MFM AND CAUSALITY

The main strength of MFM is the ability to describe very complex systems using a small number of modeling elements (functions). For each of the legal connections between the MFM functions, a set of *causal rules* is defined. These causal rules describe how qualitative states of the functions affect each other. The idea of implicit causal rules was invented in Larsson (1992).

For example, the connection of a transport to a storage (for example, a pump to a tank) implies the following four causations:

1. Transport low => storage low
2. Transport high => storage high
3. Storage low => transport high
4. Storage high => transport low

The causal rules are used with one or more diagnostic methods, such as, alarm analysis, discrete sensor validation, or failure mode analysis.

Inhibiting Causal Dependencies

In order to explain the inhibition of causal dependencies, consider a simple example. We will use the open tanks process again. In the closed tanks example, all possible causal dependencies are active. Thus, if the right tank overfills, it will stop the outflow from the left tank, which will become overfull too, which will stop the pump flow, and so on.

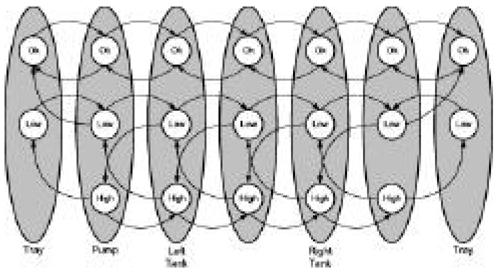

Figure 4. The complete set of causal dependencies for the closed tanks process.

The complete set of causal interactions are shown in Figure 4, where each ellipsis describes one piece of equipment (in fact, flow function), with the three possible states of OK, failed low, and failed high. The arcs show causal effects from one state to another. For example, a low fault in the leftmost ellipsis (the source function of the tray tank) will cause a low fault in the second ellipsis (the transport function of the pump). This is described by the arc from the "low" in the leftmost ellipsis to the "low" in the second ellipsis, and so on.

However, when the tanks are not closed, they can be overfull but the outflows of upstream equipment will not be hindered. In the open tanks case, this means that one of causal dependencies (from an lower tank high to a upper tank outflow low) will not be valid. In the same way, several other causal dependencies are not valid for the open tanks system. In Figure 5, these casual dependencies have been removed.

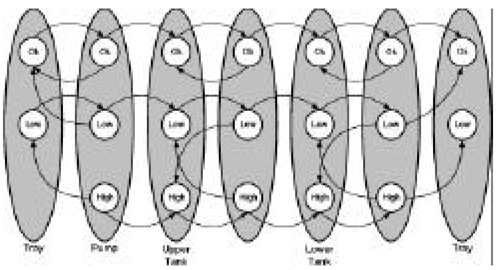

Figure 5. The set of causal dependencies for the open tanks process.

By explicitly activating or deactivating the casual dependencies of each connection between different functions (or equipment), similar physical systems with slight differences in causality can be modeled efficiently.

Variable Causality in MFM

The current method is generally valid for qualitative physics representations, but it was developed with MFM in mind.

In basic MFM it is assumed that all connected functions affect each other in both directions. For example, current MFM assumes that the model in Figure 3 has the causality of the real system in Figure 2, that is, the closed tanks system. There is no way of describing the difference between the systems with open and closed tanks.

As mentioned, one solution is to have more MFM symbols to represent various types of objects such as open tanks, closed tanks, centrifugal pumps, etc. This solution quickly becomes unmanageable, since it may be difficult to find the appropriate symbol to use in a specific system, and the modeling effort becomes large.

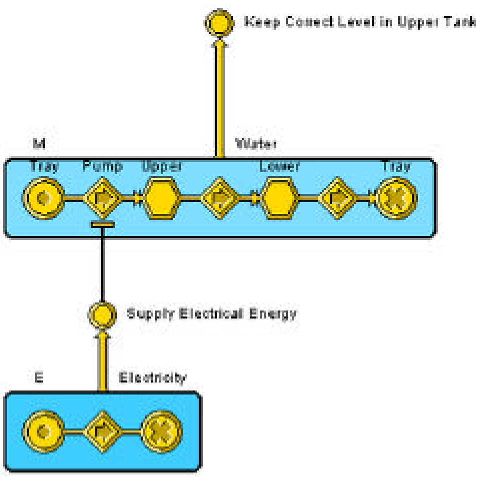

Figure 6. An MFM model of the open tanks process with some active and some inactive causal dependencies.

The other solution, which is described in this paper, is to add an attribute to the causal relation indicating in which states the relation is active or not. Specifically in the MFM models this may be indicated graphically by adding one new modeling element.

By selecting casual connections in the MFM model graph, the user can activate or deactivate casual dependencies. The activation state is shown with arrows, so called “shark fins,” see Figure 6.

The implementation is simple. Each diagnostic algorithm uses the set of causal rules, and if a causal dependency is deactivated, the corresponding rule is not used in the reasoning.

The General Case

In the examples above, three values have been used to describe the working state of each object: OK, failed low, and failed high. However, it is possible to base a qualitative physics on more values, such as *very low, low, slightly low,* etc. It may also be that one object has a different number of states than the one it is connected to. In a general case, a causal connection implies n*m potential dependencies, each in either right or left direction, for a total of 2*n*m arcs.

	Obj. 2 State 1	Obj. 2 State 2	...	Obj. 2 State m
Obj. 1 State 1	← →	← →	...	← —
Obj. 1 State 2	← —	← —	...	— →
...	...	...	...	...
Obj. 1 State n	— →	— →	...	← →

Figure 7. A general inhibition matrix for a connection consisting of 2*n*m potential causal dependencies.

The input matrix shown in Figure 7 can be used to switch all 2*n*m dependencies on or off.

Time-Varying Causality

Another complication concerning variable causality is that of dynamical behavior. The validity of causal dependencies may vary according to time and process state. For example, a closed tank may have a lid, which when opened turns the tank into an open one.

In the proposed solution, each arc (or place in the input matrix) can be connected to a logical signal, which in turn may be a function of time and operating state. In this way, it is possible to use the same model to capture time-varying causality, such as tanks with opening lids, batteries with over-charge protection units that break, etc.

A Comparison with Petersen

The problem of varying causality in MFM was first observed in Larsson (1992). The solution suggested there was to use different causal rule sets for different domains, which probably is an intractable solution in practice.

Petersen (2000) introduces two kinds of causal MFM connections, with different causal inhibitions. Petersen’s solution contains the seed of the method proposed in this article, but goes only part of the way.

To illustrate the difference between Petersen and the current method, assume that an MFM transport is connected to a storage. The possible casual dependencies are:

1. Transport low => storage low
2. Transport high => storage high
3. Storage low => transport high
4. Storage high => transport low

In Petersen's first case, all four dependencies are active, while in his second case, dependency 3 is deactivated, while the others are still active. Petersen recognizes only two different cases. In this way, Petersen's solution is limited to using two cases for any 2 * n * m choice. Furthermore, Petersen only discusses *static* differences, described once and for all in the model, and he ignores dynamical changes in causality.

RELATED WORK

The main contributions to MFM have been made by Morten Lind and his group. Lind (1990 a, 1994) describes the basics of MFM, while Lind (1990 b) contains an early suggestion for a diagnostic system. Lind has also treated real-time diagnosis, Lind (1990 c), and design of operator interfaces, Lind (1989).

MFM has also been used in nuclear safety research, De et al. (1982) and Businaro et al. (1985), in operator interfaces for fault diagnosis, Duncan and Prætorius (1989), for constructing COGSYS diagnostic systems, Sassen (1993), for fault diagnosis in process industry, Walseth (1993), and in intelligent man-machine systems for nuclear plants, Monta et al. (1991).

Larsson has used MFM in monitoring and diagnosis for intensive-care units, Larsson and Hayes-Roth (1998), Larsson, et al. (1997 a, b).

MFM is the main basis for the startup company GoalArt, which works with diagnostics for various branches of industry, for example nuclear power generation, see Larsson (2000).

MFM can be compared to other modeling and diagnosis methodologies, such as rule-based expert systems, fuzzy logic, qualitative physics based on Reiter's algorithm, Hamscher et al. (1992), Reiter (1987), Greiner et al. (1989), classical statistical methods, methods from control theory, Frank (1996), and neural networks. In comparison to expert systems and fuzzy logic, MFM imposes a deep model structure of means and ends, as opposed to a shallow rule-based representation. It differs from qualitative physics in that it explicitly represents goals and functions, avoids general logic, and is computationally more efficient, while qualitative physics has been geared towards diagnosis of electrical circuits, a task which MFM is not very well adapted for. MFM differs from statistical and control theory methods in that it uses discrete and more abstract representations, and thus is useful on a higher level of decision and diagnosis. For example, control theory methods are usually aimed at fault detection on control loop level, while MFM is aimed at diagnostic reasoning on a plant-wide level. Finally, MFM differs strongly from neural networks in that it explicitly represents human knowledge using linguistic concepts, and that the model construction relies almost completely on available human knowledge and not on automatic generalization of test cases.

CONCLUSIONS

Due to small differences in construction and behavior of connected equipment, the causality of such connections may change. This poses a potential problem for causal and functional modeling languages, such as MFM. The article presents a general solution, which allows efficient modeling of systems with varying causality without a large multiplication of language objects. The proposed solution also allows varying causality depending on time and operating state.

ACKNOWLEDGEMENTS

The authors would like to thank Morten Lind, who invented MFM and have supported our efforts from the beginning, and Johannes Petersen for his contributions on varying causality in MFM connections. We would also like to thank the staff at GoalArt, Antonio Calzada, David Sjölander, Jan Tuszynski, and Anu Uus for providing inspiration, help, and a nice mental environment in our daily lives.

REFERENCES

Businaro, T., A. Di Lorenzo, G. B. Meo, M. I. Rabbani, and E. Rubino, "An Application of MFM Method for Nuclear Plant State Identification," Proceedings of the Halden Programmer's Group Meeting on Computerized Man-Machine Communication, Gothenburg, 1985.

Dahlstrand, F., "Alarm Analysis with Fuzzy Logic and Multilevel Flow Models", Proceedings of the 18th Annual International Conference of the British Computer Society Special Group on Expert Systems, ES98, Cambridge, England, pp.173–188, 1998.

Dahlstrand, F., *Methods for Alarm Reduction with Multilevel Flow Models of Industrial Processes,* Licentiate thesis, Department of Information Technology, Lund University, Lund, Sweden, 2000.

Dahlstrand, F., "Consequence Analysis Theory for Alarm Analysis," *Knowledge-Based Systems,* vol. 15, no. 1-2, pp. 27-36, 2002.

De, M. K., J. A. Rumancik, A. J. Impink, and J. R. Easter, "A Functional Design Approach to PWR Safety," Proceedings of the International Meeting on Thermal Nuclear Reactor Safety, Chicago, Illinois, 1982.

Duncan, K. D. and N. Prætorius, “Flow Displays Representing Complex Plant for Diagnosis and Process Control,” Proceedings of the 2nd European Meeting on Cognitive Science Approaches to Process Control, Siena, 1989.

Frank, P. M., “Analytical and Qualitative Model-Based Fault Diagnosis -? A Survey and Some New Results,” *European Journal of Control,* vol. 2, pp. 6–28, 1996.

Fussell, J. B., “A Review of Fault Tree Analysis with Emphasis on Limitations,” Proceedings of the 6th Triennial World Congress of the International Federation of Automatic Control, Pittsburgh, Pennsylvania, 1975.

Greiner, R., B. A. Smith, and R. W. Wilkerson, “A Correction to the Algorithm in Reiter’s Theory of Diagnosis,” *Artificial Intelligence,* vol. 41, pp. 79–88, 1989.

Hamscher, W., L. Console, and J. de Kleer, (Eds.), *Readings in Model-Based Diagnosis*, Morgan Kaufmann, San Mateo, California, 1992.

Ingström, D., “MFM Modeling and Alarm Analysis of the Barsebäck Nuclear Power Plant,” Master’s thesis, Department of Information Technology, Lund Institute of Technology, Lund, 1998.

Inoue, K. and E. J. Henley, “Computer-Aided Reliability and Safety Analysis of Complex Systems,” Proceedings of the 6th Triennial World Congress of the International Federation of Automatic Control, Pittsburgh, Pennsylvania, 1975.

Larsson, J. E., *Knowledge-Based Methods for Control Systems,* Doctor’s thesis, TFRT–1040, Department of Automatic Control, Lund Institute of Technology, Lund, 1992.

Larsson, J. E., “Diagnostic Reasoning Strategies for Means-End Models,” *Automatica,* vol. 30, no. 5, pp. 775–787, 1994.

Larsson, J. E., “Diagnosis Based on Explicit Means-End Models,” *Artificial Intelligence,* vol. 80, no. 1, pp. 29–93, 1996.

Larsson, J. E., “Avoiding Human Error,” Proceedings of the International Conference on Control and Instrumentation in Nuclear Installations, Bristol, England, 2000.

Larsson, J. E., “Diagnostic Reasoning Based on Means-End Models: Experiences and Future Prospects,” *Knowledge-Based Systems,* vol. 15, no. 1-2, pp. 103-110, 2002.

Larsson, J. E. and F. Dahlstrand, “New Algorithms for MFM Alarm Analysis,” invited paper, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Diego, California, 1998.

Larsson, J. E. and B. Hayes-Roth, “Guardian: An Intelligent Autonomous Agent for Medical Monitoring and Diagnosis,” *IEEE Intelligent Systems,* vol. 13, no. 1, pp. 58–64, 1998.

Larsson, J. E., B. Hayes-Roth, and D. M. Gaba, “Goals and Functions of the Human Body: An MFM Model for Fault Diagnosis,” *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 27, no. 6, pp. 758–765, 1997 a.

Larsson, J. E., B. Hayes-Roth, D. M. Gaba, and B. E. Smith, “Evaluation of a Medical Diagnosis System Using Simulator Test Scenarios,” *Artificial Intelligence in Medicine,* vol. 11, pp. 119–140, 1997 b.

Lind, M., “Human-Machine Interface for Diagnosis Based on Multilevel Flow Modeling,” Proceedings of the 2nd European Meeting on Cognitive Science Approaches to Process Control, Siena, 1989.

Lind, M., “Representing Goals and Functions of Complex Systems — An Introduction to Multilevel Flow Modeling,” Technical report, 90–D–38, Institute of Automatic Control Systems, Technical University of Denmark, Lyngby, 1990 a.

Lind, M., “Abstractions Version 1.0 — Descriptions of Classes and Their Use,” Technical report, 90–D–380, Institute of Automatic Control Systems, Technical University of Denmark, Lyngby, 1990 b.

Lind, M., “An Architecture for Real-Time MFM Diagnosis,” Technical report, Institute of Automatic Control Systems, Technical University of Denmark, Lyngby, 1990 c.

Lind, M., “Modeling Goals and Functions of Complex Industrial Plants,” *Applied Artificial Intelligence,* vol. 8, no. 2, pp. 259–283, 1994.

Monta, K., J, Takizawa, Y. Hattori, T. Hayashi, N. Sato, J. Itoh, A. Sakuma, and E. Yoshikawa, “An Intelligent Man-Machine System for BWR Nuclear Power Plants,” Proceedings of AI91 — Frontiers in Innovative Computing for the Nuclear Industry, Jackson, Wyoming, 1991.

Petersen, J., *Knowledge-Based Support for Situation Assessment in Human Supervisory Control,* Ph.D. thesis, Department of Automation, Technical University of Denmark, Lyngby, Denmark, 2000.

Öhman, B., “Failure Mode Analysis Using Multilevel Flow Models,” Proceedings of the 5th European Control Conference, Karlsruhe, Germany, 1999.

Öhman, B., *Real-Time Diagnosis of Industrial Processes Using Multilevel Flow Models,* Licentiate thesis, Department of Information Technology, Lund University, Lund, Sweden, 2001.

Öhman, B., “Discrete Sensor Validation with Multilevel Flow Models,” *IEEE Intelligent Systems,* vol. 17, no. 3, pp. 55-61, 2002.

Reiter, R., “A Theory of Diagnosis from First Principles,” *Artificial Intelligence,* vol. 32, pp. 732–737, 1987.

Sassen, J. M. A., *Design Issues of Human Operator Support Systems,* Doctor’s thesis, Faculty of Mechanical Engineering and Marine Technology, Laboratory for Measurement and Control, Delft University of Technology, Delft, 1993.

Walseth, J. Å., Diagnostic Reasoning in Continuous Systems, Doctor’s thesis, ITK-rapport 1993: 164–W, Division of Engineering Cybernetics, Norwegian Institute of Technology, Trondheim, 1993.