Consequence Analysis Theory for Alarm Analysis

Fredrik Dahlstrand

Department of Information Technology, Lund University

SE-221 00 Lund, Sweden

`fredrikd@it.lth.se`

Alarm analysis is the task of finding the root cause of a failure situation in an industrial process. In this paper a new approach for performing alarm analysis using multilevel flow models (MFM) is introduced. This approach is compared with existing methods for alarm analysis using MFM. The result of the alarm analysis presented in this paper is a set of explanations that fits the observed state of the process. Furthermore, it is shown that the old algorithms is a special case of the algorithm presented here.

1 Introduction

Most modern industrial processes are often equipped with a large number of sensors. The information from the sensors is provided to the plant operators, who perform supervision of the process. In larger processes it is a very complex task for the operators to get an overview of a failure situation since even a small fault may cause the supervision system to trigger several hundred alarms in a short time interval. This stressful situation may cause the plant operators to make more or less serious mistakes since they are unable to sort out which alarms that are important, and which alarms that are not. Even a single fault may give rise to a complex alarm situation since the fault may have consequences that trigger other alarms. If there is a single fault with several of consequential faults it is not a straightforward task for the operator to identify the root cause. Depending on the instrumentation of the process the alarms may arrive to the plant supervisory system out of order, that is, the root cause of the alarm situation may not be the first alarm that arrives in a multiple-alarm situation.

The task of an *alarm analysis* system is to aid the plant operator by determining which alarms are root causes and which alarms are consequences. It is important to note that even if an alarm is a consequence alarm it may still be more important than the root cause. For example, if the feed water pumps in a nuclear power plant would fail, the reactor will not receive enough cooling. The main hazard in this situation is a nuclear meltdown, so the task with the highest priority is not to repair the broken feed water pump, but rather to cool the reactor core.

There exist several approaches to perform consequence analysis (cause-effect analysis) using qualitative models. *Signed Directed Graphs* (SDG) (see for example [13]) is an approach where the vertices in the signed directed graph represent process variables and the edges shows how these variables interact with each other. *Fuzzy Cognitive Maps* (FCM) [6] are also based on signed directed graphs, where the vertices represent events, and the edges represent the causal relationship between these events. The classical method for failure analysis is *Failure Mode and Effect Analysis* (FMEA), which analyses all components, their failure modes, and impact on plant operation. This approach is not able to handle combinations of failures. To give a more detailed failure analysis, the *Fault Tree Analysis* (FTA) method may be used. FTA uses logical relations between component faults to model more complex process failures. Both FMEA and FTA suffer from modeling problems; it is difficult to find all possible ways a component may fail, and it is even harder to figure out which component faults may interact to cause complex failure situations. An approach that attacks the problem from another angle is the *Goal Tree-Success Tree* (GTST) method [5]. This method models the goals and subgoals of the plant and which components that need to be available for achieving these goals. This paper will focus on alarm analysis using *Multilevel Flow Models* (MFM), which is explained later.

In this paper an observation is the measured state of one of the process components, for example the level in a tank. However, it is not necessary that the measurements are taken directly from the process. They could also be laboratory tests, or estimations of non-measurable process variables. The important thing is that the measurements are correct. For the rest of this paper it is assumed that there exist methods that provide the algorithm with correct observations.

In Section 2 multilevel flow models are described. In Section 3 the definitions will be illustrated by using a small example. In Section 4 there is a somewhat larger example, and the result are compared with the old alarm analysis algorithms. The last section contains a discussion of the results, and future work that needs to be done.

2 Multilevel Flow Models

Multilevel flow models (MFM) [11] is a graphical language for representing the goals and intended functionality of an (industrial) process. A major difference between MFM and other modeling approaches is that it models the *means-end* dimension, rather than the *part-whole* dimension. There are three important entities in MFM: *goals, functions*, and *components*.

Goals represent the intended purposes of the process (the answer to the question "Why?"). The functions represent the capabilities of the system (the answer to the question "How?"). Components are the physical process components, for example, pumps, pipes, and valves. One MFM function can represent the capabilities of one or more components. One component can also be realized by one or more functions.

The functions describe the capabilities of the process in terms of flows of mass and energy. The MFM functions are connected together to describe a flow. There are ten different MFM functions, the five most common are:

- *Source*, provides mass or energy, for example, a battery.
- *Transport*, transports mass or energy, for example, a pump.
- *Barrier*, prevents the transport of mass or energy, for example, a closed valve.
- *Storage*, stores mass or energy, for example, a tank.
- *Sink*, a drain of mass or energy, for example, a lamp in an electrical circuit.
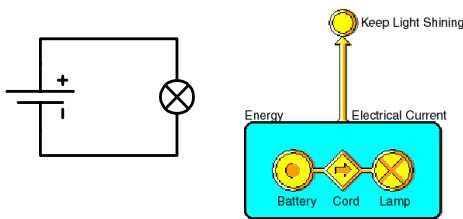

Figure 1. A simple electrical circuit and an MFM model representing the circuit's capability of keeping the light shining.

A simple process is shown in Figure 1. The goal of this process is to illuminate its environment by keeping the light shining. The process has three components; a battery (a source of electrical energy), a lamp (a drain of electrical energy), and a cord (a transport of electrical energy) to connect the battery to the lamp. To the right in Figure 1 is the MFM model of this process. The gray rectangle with rounded corners represents the flow of electrical energy in the circuit (from the battery, through the cord, to the lamp.) The symbols, from left to right, are the MFM functions, source, transport, and sink. The circle at the top represents the goal of the process, in this case to keep the light shining. A collection of MFM functions connected together to describe a flow is called a network (the rectangle). The arrow between the network and the goal is an achieve relation, which means that in order for the goal to be achieved all functions in the network must be working as intended.

Function	Alarm State		
Source	Low capacity	OK	
Transport	Low flow	OK	High flow
Barrier		OK	High flow
Storage	Low volume	OK	High volume
Sink	Low capacity	OK	

Table 1. The alarm states for the MFM functions.

Each MFM function is associated with a number of alarm states, which were introduced by [7]. For example, the source may either be working as expected, or it may be in a state of *low capacity*, that is, the source is not able to provide as much

mass or energy as required. The source is a passive function in the sense that it is not possible for the source to provide more than requested. The transport on the other hand may have a too low, a too high flow, or a normal flow (see Table 1).

MFM has successfully been used in several projects. In the Guardian project MFM was used for monitoring patients in an intensive care unit [10]. MFM was used to model the Steritherm process, which is a process that sterilizes liquid food products, for example milk [7]. In an on-going project MFM is used for alarm analysis of the Barsebäck nuclear power plant [9].

3 Consequence Analysis

The method for performing consequence analysis presented in this paper uses *causal dependency graphs*. These graphs are a representation of the causal relationship between the MFM functions. The graphs are not built directly but are the result of the knowledge put into the MFM model. Even though the graphs may look complicated, the modeling effort is relatively simple, since there is only a small number of MFM functions with a fixed number of connections. An approach of doing alarm analysis by combining MFM and signed directed graphs can be found in [3].
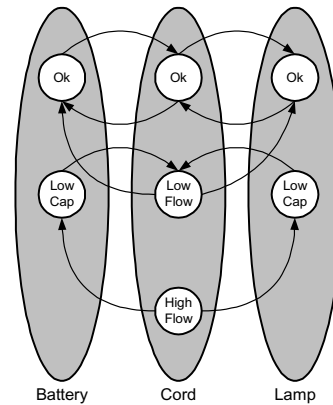

Figure 2. The causal dependency graph of the MFM model in Figure 1.

Definition 1. Let $G = (V, E)$ be a graph where the vertices, V, represent the *fault states* for a function, and the edges, E, represent *causal relations* between the states. Then G is a *causal dependency graph*.

Figure 2 shows the causal dependency graph for the MFM model in Figure 1. For example, if the battery is running low, that is, not being able to provide enough electrical energy, then the electrical current through the cord will be too low. This is the arrow between the *low cap* state in the battery and the *low flow* state in the cord in Figure 2.

Definition 2. Given a set A and a set $B = \{A_0, A_1, ..., A_n\}$ where $A_i \subseteq A$ for all $A_i \in B$. Then B is a partition of A if

1. $A = \bigcup_{A_i \in B} A_i$ and

2. $A_i \cap A_j = \varnothing$ for all $A_i, A_j \in B$ where $i \neq j$.

Each subset A_i is called a *cell* of the partition.

Definition 3. Let $G = (V, E)$ be a causal graph, and let F be a partition of V such that all the states in cell $F_i \in F$ are mutually exclusive. The partition F is called a *fault state partition* and the cell F_i is called a *fault state*.

For example, the current through the cord in the circuit in Figure 1 may be either too high, too low or normal. However, the current cannot be too high and too low at the same time. The is shown in Figure 2 as gray ellipses containing three mutually exclusive states. For example, the cord may be either in the *low cap*, *high cap*, or the *ok* fault state.

Definition 4. Let $G = (V, E)$ be a causal graph with fault state partition F, and $v_i, v_k \in V$. Then there is a *causal path* $P = \{v_i = p_0, p_1, ..., p_n = v_k\}$ between v_i and v_k if $|F_i \cap P| \leq 1$ for all $F_i \in F$.

Definition 5. Let $G = (V, E)$ be a causal graph, let F be a partition of V, and let $O = \{o_1, o_2, ..., o_n\}$ be a set of observed fault states in G. Then $G^I = (V^I, E^I)$ is the *inferred causal graph* where

$$V^I = V - \left(\bigcup_{F_i \in F} F_i - O \right) \text{ and}$$

$$E^I = \left\{ (v_i, v_k) \mid (v_i, v_k) \in E \wedge v_i, v_k \in V^I \right\}.$$

Figure 3 shows the inferred causal graph when the observation is that the electrical current through the cord is too high. Since we know the fault state of the cord is *high flow*, we do not have to consider the cases which involve the other fault states of the cord. Thus, those states are removed from the graph together with all causal relations to and from those states. This is how an causal dependency graph is transformed into an inferred causal graph.

Figure 3 also shows that since we know the fault state of the cord, we can make an educated guess of the fault state of the other functions. The battery is in a state where it provides more electrical energy than it was designed for, and will be drained out sooner than expected. There is also a risk that the lamp will be destroyed since it receives a higher current that it was designed for. This means that both the battery and the lamp are in an undesired state and should be considered as faults.
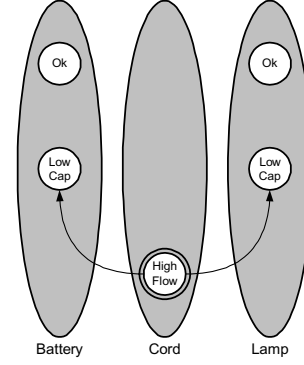

Figure 3. The inferred causal graph (from the graph in Figure 2) when a high flow through the cord has been observed.

Definition 6. Let $G = (V, E)$ be a causal graph. Then the *causality closure* of G is defined as $G^* = (V, E^*)$ where

$$E^* = \{(v_i, v_k) \mid \text{there is a causal path between }$$
$$v_i \text{ and } v_k \text{ in } G\}$$

Definition 7. Let $G = (V, E)$ be a causal graph. Given a set of observations O, $G^I = (V^I, E^I)$ is the inferred causal graph. Further, let $G^* = (V, E^*)$ be the causality closure of G^I. Then the *possible root causes*, C^R, is defined as

$$C^R = \{v_k \mid (v_k, v_i) \in E^* \wedge v_i \in O\}.$$

Definition 8. Let $G = (V, E)$ be a causal graph. Given a set of observations O, $G^I = (V^I, E^I)$ is the inferred causal graph. Further, let $c^R \in C^R$ be a possible root cause. Then the *effects*, c^E, of that root cause are defined as

$$c^E = \left\{ v_i \mid \text{if there is a causal path between } c^R \text{ and } v_i \text{ in } G^I \right\}.$$

If a state is a possible root cause there exists an causal path from that state to at least one of the observed states. Figure 4 shows the possible root causes (the states within squares) when the electrical current through the cord is observed as being too low. Note that a state always has a implicit causal path to itself. All states on the causal path from a root cause is considered effects of that root cause.

Definition 9. Let C^R be a set of possible root causes of observation O. Then the *simple solutions* to this situation is defined as

$$S = \left\{ (c_i^R, c_i^E) \,\middle|\, c_i^R \in C^R \wedge c_i^E \text{ are the effects of } c_i^R \right\}.$$
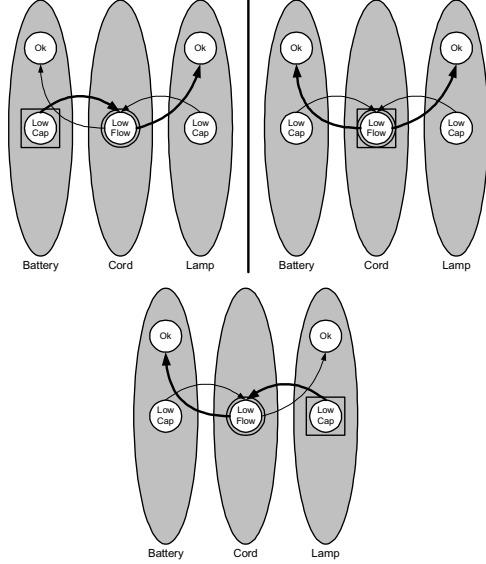
Figure 4. When the flow (electrical current) through the cord is observed to be too low, there exist three possible explanations. The states within squares represent the root cause.

Definition 10. Let $s_1 = (c_1^R, c_1^E)$ and $s_2 = (c_2^R, c_2^E)$ be two simple solutions then s_1 is larger or equal to s_2, denoted $s_1 \geq s_2$, if $c_2^E \subseteq c_1^E$.

Definition 11. Let $S = \{s_1, s_2, ..., s_n\}$ be a set of simple solutions. Then the *largest solution*, max S, is defined as
$$\max S = \{ s_k \mid s_k \geq s_i \text{ for all } s_k, s_i \in S \}.$$

Definition 12. Let $S = \{s_1, s_2, ..., s_n\}$ be a set of simple solutions. Then $S^* = \{S_1^*, S_2^*, ..., S_n^*\}$ is a *solution partition* of S such that
$$S_m^* = \{ s_i \mid (s_i \subseteq s_k) \vee (s_k \subseteq s_i) \text{ for all } s_i, s_k \in S_m^* \}.$$

Definition 13. Let $S = \{s_1, s_2, ..., s_n\}$ and let $S^* = \{S_1^*, S_2^*, ..., S_n^*\}$ be the solution partition of S. Then the *partial explanation*, S^E, is defined as
$$S^E = \{ \max S_k^* \mid S_k^* \in S^* \}.$$

	Battery	Cord	Lamp
A	**Low Cap.**	Low Flow	OK
B	OK	**Low Flow**	OK
C	OK	Low Flow	**Low Cap.**

Table 2. The three explanations, A, B, and C, for the alarm situation in Figure 4. The states with gray background are the root causes respectively.

The alarm situation, the electrical current is too low, in Figure 4 gives raise to the three partial explanations in Table 2. Case *A* means that the battery has run low, and causes the cord to transport too little electrical energy. This will however not affect the fault state of the lamp, which will not break if it receives too little electrical energy. In case *B* the root cause is the fact that the flow is too low, that is, the cord may be broken. In the final case (*C*) the explanation for the situation is that the lamp is broken, that is, is not able to receive any electrical energy. All three explanations are equally plausible, and it is impossible to distinguish between them without making any further observations.
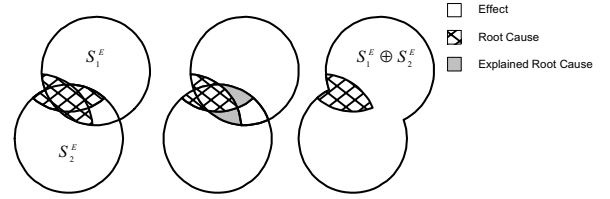
□	Effect
⊠	Root Cause
▨	Explained Root Cause

Figure 5. The result of combining the two explanations S_1^E and S_2^E.

In a larger causal graph it is possible that one partial explanation does not cover all observed states. That means that there must be more than one root cause. To find a explanation that satisfies all observations, two or more explanations may be joined together. Figure 5 show what happens when two explanations are joined together. Some of the root causes in one of the partial explanations may be explained by a root cause in one of the other explanations, and should not be considered as root causes.

Definition 14. Two partial explanations, $S_1^E = (C_1^R, C_1^E)$ and $S_2^E = (C_2^R, C_2^E)$, are *joinable* if $\left| (C_1^R \cup C_2^R) \cap F_k \right| \leq 1$ for all $F_k \in F$, where F is the partition of all states in the model. This means that two partial explanations are joinable if their root causes do not conflict with each other, that is, there is at most one state from each fault state partition.

Definition 15. The *combined explanation* of two partial explanations, $S_1^E = (C_1^R, C_1^E)$ and $S_2^E = (C_2^R, C_2^E)$, is defined as

$$S_1^E \oplus S_2^E = \left(\left(C_1^R - C_2^E \right) \cup \left(C_2^R - C_1^E \right) \cup \left(C_1^R \cap C_2^R \right), C_1^E \cup C_2^E \right)$$

if the explanations are joinable.

Definition 16. An explanation, $S^E = (C^R, C^E)$, is *complete* if $\left| C^E \cap F_k \right| \geq 1$ for all $F_k \in F$. Where F is the partition of all states in the model.

All the explanations in Table 2 are complete.

3.1 Choice of Explanations

One of the difficult problems when writing an alarm analysis algorithm is to choose which of the explanations should be presented to the plant operator. This is not a problem that is specific to MFM, but rather a general problem [2, 12]. For example, one of the following methods may be used:

1. Present the first explanation with the least number of root causes.
2. Present the whole group of explanations with the least number of root causes.
3. Present everything.

The drawback of method 1 and method 2 is that they may miss the true explanation. However, it makes sense to choose an explanation with as few root causes as possible, since it is more probable that there is a single fault rather than a more complicated failure situation. The drawback of method 3 is that it may explode and that the amount of information presented to the operator may be huge. However, if the explanations could be presented in some ordered fashion, this may help in the process of locating the true fault (faults).

It is also possible to make some other assumptions. For example, the algorithms by Larsson [8] and Dahlstrand [1], use the assumption that the root cause may only be one of the observed states. In the case of the alarm situation in Figure 4, both these algorithms will choose case *B*, and present it to the operator. The drawback of this is that sometimes these methods try to find a more complicated explanation than necessary, and even worse the explanation may be wrong. Furthermore, the algorithm by Larsson may give different results depending on the order in which the alarms arrive to the supervision system.

4 The Two Tank Process

As an example of a somewhat larger process, with more complicated alarm situations, the two tank process in Figure 6 will be used. The process has a pump that pumps water from an external water supply to the first tank. The two tanks are connected by a pipe, and the second tank lets the water out to some

external environment. The purpose of this process is to keep the water levels in the two tanks at a specified level.
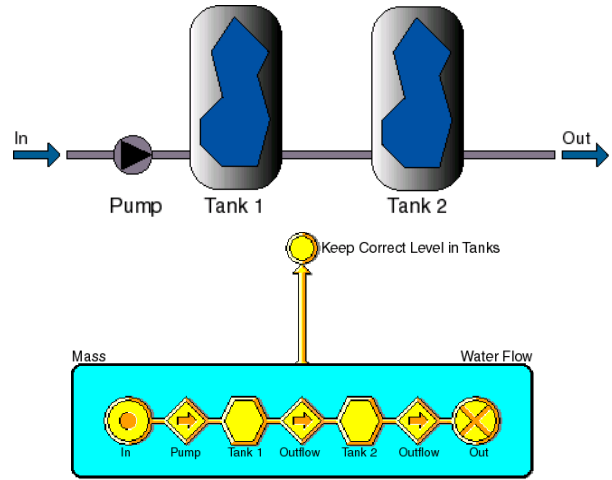

Figure 6. The two tank process and an MFM model representing the flow of water in the two tank process.

The MFM functions in Figure 6 are, from left to right: A *source* that is realized by the external water supply. A *transport*, which is realized by the pump, a *storage* that is realized by tank 1, a transport that is realized by the pipe between the two tanks, another storage that is realized by the second tank, yet another transport, which is realized by the outflow from tank 2, finally a *sink* that is realized by the environment's capability of receiving water. The process is equipped with two sensors. The first sensor measures the level in the first tank, and the second sensor measures the level in the second tank.

Figure 7 shows the causal dependency graph, constructed from the MFM model in Figure 6.
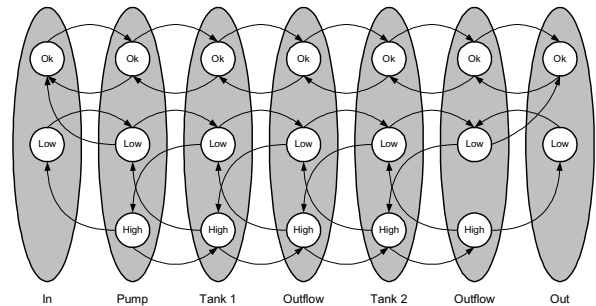

Figure 7. The causal dependency graph generated by the MFM model in Figure 6.

Assume that the level in the first tank is observed as too high, and that the level in the second tank is observed as too low. If so, the inferred causal graph is the one shown in Figure 8. Definition 7 gives the possible root causes. Those are the states within squares.
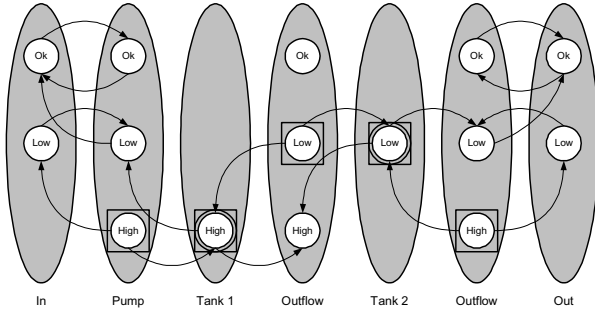
Figure 8. The inferred causal graph when the level in the upper tank is observed as too high, and the level in the lower tank is observed as too low. The states in squares are the possible root causes.

From the inferred causal graph and Definition 9 the five partial explanations for the observations are shown in Table 3. The states with gray background are the root causes. Only case *A* is a complete explanation. Cases *B* through *E* are partial explanations that can be combined in four different ways. The complete explanations in Table 4 are all combinations of joinable explanations in Table 3, see Definition 14 and Definition 15.

	In	Pump	Tank 1	Out-flow	Tank 2	Out-flow	Out
A	OK	Low Flow	High Vol.	**Low Flow**	Low Vol.	Low Flow	OK
B	Low Cap.	**High Flow**	High Vol.	High Flow	-	-	-
C	OK	Low Flow	**High Vol.**	High Flow	-	-	-
D	-	-	-	High Flow	**Low Vol.**	Low Flow	OK
E	-	-	-	High Flow	Low Vol.	**High Flow**	Low Cap.

Table 3. The partial explanations for the inferred causal graph in Figure 8.

Based on the observations that the level in the first tank is too high, and the level in the second tank is too low there exist five complete explanations as shown in Table 4. Without any other observations it is impossible for any alarm analysis algorithm to distinguish between these solutions.

Depending on the choice of algorithm, different explanations will be presented to the operator. The original MFM alarm analysis algorithm by Larsson, and the algorithm by Dahlstrand will both present explanation *D* to the operator. Both these algorithms have the hidden assumption that only the observed states may be the root cause.

In the failure situation in Figure 8 the most likely solution is case *A* (Table 4), that is, that the pipe between the two tanks is clogged. Case *A* is the most probable scenario since it is more probable that there is a single fault rather than a more complex failure situation. However, the other cases are not impossible.

For example, in case *B* the pump is providing too much water to the first tank, and the level in tank two is too low. However, if this situation continues, the level in the second tank will probably fill up, and the level will be too high. The consequence analysis done here should be considered as a snapshot of the alarm situation. There is no temporal information, for example, about the order in which the observations arrives to the supervision system.

	In	Pump	Tank 1	Out-flow	Tank 2	Out-flow	Out
A	OK	Low Flow	High Vol.	**Low Flow**	Low Vol.	Low Flow	OK
B	Low Cap.	**High Flow**	High Vol.	High Flow	**Low Vol.**	Low Flow	OK
C	Low Cap.	**High Flow**	High Vol.	High Flow	Low Vol.	**High Flow**	OK
D	OK	Low Flow	**High Vol.**	High Flow	**Low Vol.**	Low Flow	OK
E	OK	Low Flow	**High Vol.**	High Flow	Low Vol.	**High Flow**	Low Cap.

Table 4. All complete explanations for the inferred causal graph in Figure 8.

Depending on the placement of the sensors in the observed process the conflicts may occur. That is, there may be situations where the fault state may, for example, be guessed as either high or low. This is the case when the levels in both tanks are observed as too high. The complete explanation for this situation is shown in Table 5. Note that in case *F* the outflow from the upper tank is undetermined, that is, it impossible to determine the fault state for the outflow. This is not as strange as it may seem. If the level in tank 1 is higher than the level in tank 2 (both is still too high) the pressure will cause the water to flow from tank 1 to tank 2, however if the opposite is true, the higher level in tank 2 may cause a backflow through the pipe between the tanks.

	In	Pump	Tank 1	Out-flow	Tank 2	Out-flow	Out
A	Low Cap.	**High Flow**	High Vol.	High Flow	High Vol.	High Flow	Low Cap.
B	OK	Low Flow	**High Vol.**	High Flow	High Vol.	High Flow	Low Cap.
C	OK	Low Flow	High Vol.	Low Flow	**High Vol.**	High Flow	Low Cap.
D	OK	Low Flow	High Vol.	Low Flow	High Vol.	**Low Flow**	OK
E	OK	Low Flow	High Vol.	Low Flow	High Vol.	Low Flow	**Low Cap.**
F	OK	Low Flow	**High Vol.**	High Low Flow	**High Vol.**	High Flow	Low Cap.

Table 5. All complete explanations for the alarm situation when the level in both tanks is observed as too high. Note the conflict in case F.

The MFM alarm analysis algorithm by Dahlstrand will present case *F*, while the algorithm by Larsson will present

either case *B* or case *C* depending on in which order the alarms arrive to the supervision system.

5 Conclusions

In this paper an approach for analyzing the cause-effect relations in processes with flows of mass and energy has been presented. The structure of MFM makes the modeling effort a relatively easy task, compared to ordinary expert systems. The old alarm analysis algorithms [1, 8] are special cases of the algorithm presented here.

The result of the alarm analysis algorithm presented here may be several possible explanations that fit the current set of observations. The explanations may then aid the plant operator in locating the faulty components.

To handle uncertainties in the process, it would be interesting to extend the approach used in this paper to include fuzzy logic as well.

One of the difficult problems that is not addressed at all in this paper is the question of how the present the result of the alarm analysis to the plant operator. It is important that he can retrieve all relevant information easily without being overwhelmed by the amount of information.

6 Acknowledgements

References

1. 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.
2. de Kleer, J. "An Assumption-Based TMS," *Artificial Intelligence*, vol. 28, no. 2, pp 127–162, 1987.
3. Fang, M., "MFM Model Based Diagnosis and Implementation," Technical Report, 94-D-712, Institute of Automatic Control Systems, Technical University of Denmark, Lyngby, 1994.
4. Frank P. M., "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy—A Survey and Some New Results," *Automatica*, vol. 26, no. 3, pp.459–474, 1990.
5. Jalashgar, A., *Identification of Hidden Failures in Process Control Systems through Function-Oriented System Analysis*, Doctor's thesis, Risö-R-936(EN), Risö National Laboratory, Roskilde, Denmark, 1997.
6. Kosko, B., *Fuzzy Engineering*, Prentice-Hall, New Jersey, 1997.
7. Larsson, J. E., *Knowledge-Based Methods for Control Systems*, Doctor's thesis, TFRT-1040, Department of Automatic Control, Lund Institute of Technology, Lund, 1992.
8. Larsson, J. E., "Diagnosis based on explicit means-end models," *Artificial Intelligence*, vol. 80, no. 1, pp. 29–93, 1996.
9. Larsson, J. E. and B. Öhman, "Model-Based Alarm Analysis for Large Plants," invited paper, Proceedings of the International Conference on Systems, Signals, Control, Computers, Durban, South Africa, 1998.
10. 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.
11. Lind, M., "Representing Goals and Functions of Complex Systems—An Introduction to Multilevel Flow Modelling," Technical report, 90-D-38, Institute of Automatic Control Systems, Technical University of Denmark, Lyngby, 1990.
12. Reiter, R., "A Theory of Diagnosis from First Principles," *Artificial Intelligence*, vol. 32, no. 1, pp 57–95, 1987.
13. Shiozaki, J., B. Shibata, H. Matsuyama, and E. O'Shima, "Fault Diagnosis of Chemical Processes Utilizing Signed Directed Graphs—Improvement by Using temporal Information," *IEEE Transactions on Industrial Electronics*, vol. 36, no. 4, pp. 469–474, 1989.