DTU Library

Consequence Reasoning in Multilevel Flow Modelling

Zhang, Xinxin; Lind, Morten; Ravn, Ole

Link back to DTU Orbit

Consequence Reasoning in Multilevel Flow Modelling

Xinxin Zhang*. Morten Lind*
Ole Ravn*

*Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby 2800
Denmark (Tel: +45 45253562; e-mail: {xinz, mli, or}@ elektro.dtu.dk).

Abstract: Consequence reasoning is a major element for operation support system to assess the plant situations. The purpose of this paper is to elaborate how Multilevel Flow Models can be used to reason about consequences of disturbances in complex engineering systems. MFM is a modelling methodology for representing process knowledge for complex systems. It represents the system by using means-end and part-whole decompositions, and describes not only the purposes and functions of the system but also the causal relations between them. Thus MFM is a tool for causal reasoning. The paper introduces MFM modelling syntax and gives detailed reasoning formulas for consequence reasoning. The reasoning formulas offers basis for developing rule-based system to perform consequence reasoning based on MFM, which can be used for alarm design, risk monitoring, and supervision and operation support system design.

Keywords: Multilevel Flow Modelling, knowledge representation, functional modelling, consequence reasoning, rule-based system, decision support system.

1. INTRODUCTION

While operating complex industrial plants, the operators are required to be able to manage disturbances by performing manual interventions to the system. For disturbance management, it is crucial that operators are able to assess the disturbance situation at the system level. System level situation assessment is a major task for decision support system so it can assist the operators in making intervention plans when disturbances emerge. Petersen (2001) proposed to use a functional modelling approach, namely Multilevel Flow Modelling (MFM), to do situation assessment, and developed a root-cause reasoning system based MFM. In addition to identification of root causes, one of the major elements in situation assessment is to make projections of future plant status based on the understanding of current situations (Endsley et. al. 2003). MFM's capability for consequence reasoning is a major contribution to the situation assessment applications. Other researches also propose that consequence reasoning based on MFM can be used for alarm design (Us et. al. 2011) and risk monitoring systems (Yoshikawa et. al. 2011), which are also related to operation decision support systems.

MFM is a modelling methodology for representing complex systems in different abstraction level of specifications. It has been used for modelling engineering system in several safety critical domains such as nuclear power plant (Lind et. al. 2011, Gola 2012) and oil/gas gathering system (Wu et. al. 2011). The conceptual foundations, the development of MFM modelling language, tools, and applications have been undergoing for more than two decades. The most recent introduction for MFM can be found in (Lind 2011a). One of the most important features for MFM is that the models describe the causality of the system functions in a formalized way so causal reasoning can be performed based on the MFM models (manually or by using software tools). Lind (2011b) has introduced the fundamental principles for reasoning about causes and consequences in MFM. However, the introduction is not sufficient for readers to perform analysis based on MFM.

The aim of this paper is to further elaborate in details how to reason about consequences in MFM. To fulfil this purpose, basic MFM syntax has to be formulated first, as it is pre-required for understanding the reasoning process. Then the paper provides detailed reasoning patterns and the inference formulas for MFM consequence reasoning, so it can serve as the basis for the readers to analyse MFM models. The paper is organized as follows. Section 2 introduces the MFM and its syntax. Section 3 explains different inference formulas according to the predefined MFM patterns. Section 4 presents how rule based system can be implemented for MFM consequence reasoning. An example is given in Section 5 to demonstrate how the rule-based system works and Section 6 concludes the paper.

2. MFM AND MFM SYNTAX

MFM is a modelling method representing an industrial plant as a system which provides the means required to serve purposes in its environment (Lind 2011a). MFM models incorporate goals and objectives of the system, functions and structures that describe the physical components, and relations between functions and structures. It also adopts a predefined graphical modelling language, with symbolic

representation for objectives, functions and relations. This section provides an overview of the MFM and its syntax.

2.1 MFM Symbols

A list of the basic MFM symbols is shown in Table 1. A typical MFM model contains several flow structures and each flow structure contains partial function models to serve as means to realize functions in other flow structures or objectives/threats. Causal relations are used to describe the cause effect relations between functions within the same flow structure, while means-end relations describe the dependencies across levels. Each means-end relation is labelled with a main function which directly influences the end function in another flow level or an objective/threat. A full model shown in Fig. 1 is a MFM representation of a watermill that introduced in (Lind 2011a).
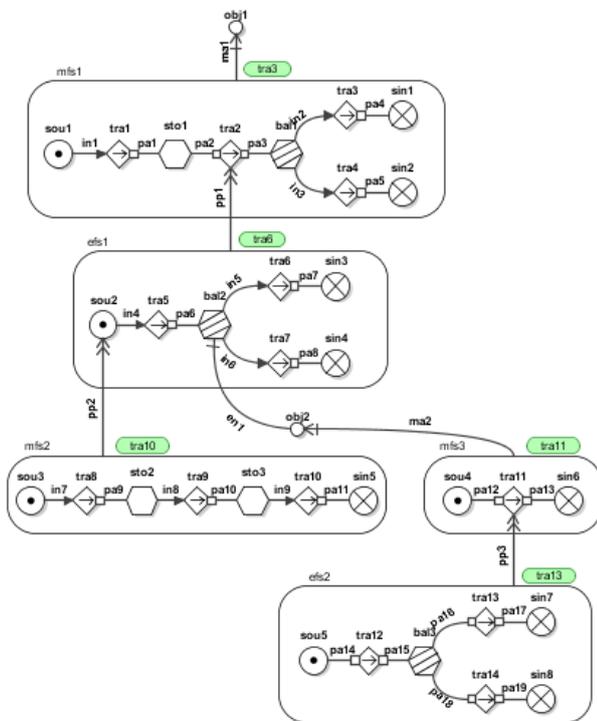

Fig. 1. MFM model for a watermill

Table 1. MFM Symbols

Because MFM semantic is in development, the scope of the paper needs to be specified. The following topics are not included in the discussion of this paper.

- Note that MFM can also represent control process (Heussen et. al. 2010, Lind 2011c) in the system. However other concepts are required for reasoning about control functions, which are not the focus of this paper. Therefore the control functions will not be included below.

- Also notice that the four special balances all share the property of a normal balance but with additional meanings which is required by some industries such as chemical engineering. These four functions will not be distinguished in this paper.

- Barrier is a special flow functions because it often serves the safety purpose in a system. The consequences of barrier failure need special treatment which is beyond the scope of this paper. (Lind 2012)

2.2 MFM Reasoning

As exemplified in Fig. 1, MFM constructs the model by using building blocks that correspond to functions and goals. It describes energy and mass flow structures in a physical system with different level of decompositions, and the representation is in an abstracted way which is independent of individual components in the physical systems.

MFM modelling is not only a way of representation, but also a convenient tool to analyse and reason about the system performance. Reasoning in MFM models is based on dependency relations between states of objectives and functions. Each function can be either enabled or disabled. For any enabled functions, the possible states are listed in Table 2. Note that the disabling of the function is not another state of the function, but means that the function will no longer be available for the system. For example, a no flow transport is different from a disabled transport, for the former lost the object that it has been transport but the latter lost the ability to transport thus the node can be viewed as a barrier after disabling.

Table 2. MFM States

Function	Possible States
Source:	normal, high volume, low volume
Sink:	normal, high volume, low volume
Transport:	normal, high flow, low flow, no flow
Storage:	normal, high volume, low volume
Barrier:	normal, leak
Balance:	normal, fill, leak, unbalance
Threat:	exist, non-exist
Objective:	false, true

The dependency relations defined in MFM are independent of the particular modelling object, and only based on predefined patterns. The patterns are created by different combinations of MFM entities, states and the MFM relations. They are defined by cause-effect relations. This will be thoroughly explained in section 3.

2.3 A brief summary of MFM Syntax

Before we go into the detail of the reasoning process, basic MFM syntax needs to be formulated. Firstly, it is important to distinguish two levels of connection in MFM models: flow connection and causal connection. Graphically, MFM describes flow functions as symbols and dependency relations as links. The mass or energy flow through a function structure only by means of transport functions. Therefore, transport functions connect other flow functions in a sense (though the functions are not graphically connected together but through a causality link). For example, a cooling system can be described in a highly abstracted function model by using MFM symbols as Fig. 2 (a). Energy (heat) generate from a heat source "sou1" and being transported to a heat sink "sin1". Without the transport function "tra1", there's no means for the energy to be transported from the sou2 to sin2. Fig. 2 (b) shows a syntactically wrong model. In MFM, a transport function is the only function to change the location of the object. Therefore, each pair of flow functions has to be connected by a transport function. The causality link "in2" does not suggest transport of mass/energy flow.
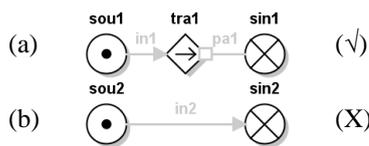

Fig. 2. Correct vs. incorrect connections

There're two types of causal influences in MFM as introduced in (Lind 2011b). Transport functions have direct influence on their neighbouring functions because how much is transported in or out will affect the states of those of the neighbouring functions. However, the upstream (sou1) and downstream (sin1) function may or may not always influence the state of the transport (tra1). The causal influence from non-transport functions to their neighbouring transports (which are categorised as indirect influence) is described by MFM causal relations. These relations are represented by arrow links between functions as shown in section 2.1. The influence direction is always from a non-transport function to a transport function, thus the arrow tip is always on the transport. A pointed tip represents an influencer relation, which means that abnormal states will influence the transport state. A square tip represents a participant relation, which describe that the neighbouring functions will not influence the transport as long as it has enough source for input and enough space for output. These relations will result in differences in the reasoning for the same function combinations.

Means-end relations describe the relations between flow structures, and thus cannot be used within one flow structure. MFM means-end relations can be categorized in two groups. Producer-product and mediate relations indicate the influence between two flow structures is instantaneous thus no temporal delay occurs during the shift of goal function perspectives. Whereas the other four means-end relations

indicate the influence goes through an objective, which is a condition for another flow structure. Thus produce, maintain, destroy, and supress relations link one flow structure and an objective/threat which are conditioning a function in another flow structure. There is a limited set of combinations that one can use when modelling means-end relations because the other combinations are not semantically correct. The allowed combination of means-end relations and their targets are showed in Fig. 3. (Lind 2012)
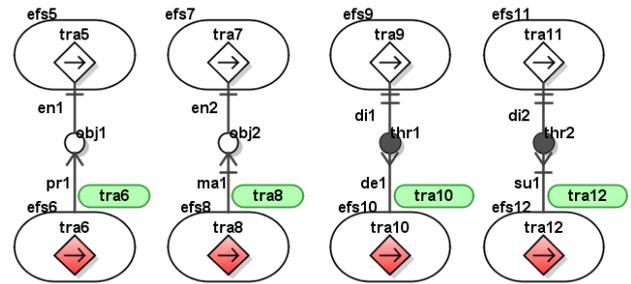

Fig. 3. Allowed link combinations across level

Only objectives are subject to be produced and maintained, while threats can only be destroyed and supressed. Also, only an objective can be the condition of enabling a function while the appearance of a threat will disable the function. This are consistent with common sense knowledge.

Among MFM functions, the source has only one out-port connection and a sink has only one in-port connection. This means that each sink or source function can only be connected with one transport function. Storage and balance are multiple-in-port and -out-port functions. This means that a storage function or a balance function has at least one transport connected to each side respectively. A transport has single in-port and single out-port, which means that a transport function can only connected two other flow functions. MFM syntax also demands that all the flow functions in one model must be and can only be included in one flow structures and each function has to be linked to other functions with causal relations (no floating functions). Each flow structures have to be linked to one or more other flow structures through means-end relations (no floating flow structures).

3. CONSEQUENCE REASONING

In this section, reasoning about MFM patterns within the same flow structure are first considered. Then we introduce cross flow structure reasoning. With MFM patterns that are in the same flow structure, we distinguish direct and indirect influence. In section 3.1 and 3.2 only patterns without balance is considered. Transport balance patterns with both direct influence and indirect influence are explained in section 3.3. Finally, how to reason across level will be introduced in section 3.4 and 3.5. In this section, all the reasoning formulas are describing possible consequences from abnormal states (causes) and the inference are not reversible.

3.1 Inference of Direct Influence

Direct influence is a cause-effect relation between a transport state to the states its neighbouring functions. After considering the MFM syntax, it is easy to deduce that only source and storage can be the upstream function for transport, meanwhile only storage and sink can be the downstream function connected to a transport. Both influencer and participant relations describe the indirect influences, but these relations do not affect the direct inference. An inference will start from a proposition (either evidence or a prediction) of the transport state under the assumption that the non-transport function is enabled. The consequence inferences are shown in Table 3.

Table 3. Inference Formulas of Direct Influence

Inference upstream:		
	Cause	Consequence
	tra1-4 high flow	sto1-2, sou1-2 low volume
	tra1-4 low flow	sto1-2, sou1-2 high volume
Inference downstream:		
	Cause	Consequence
	tra5-8 high flow	sto3-4, sin1-2 high volume
	tra5-8 low flow	sto3-4, sin1-2 low volume
Assumption: sou1-2, sto1-2, sin1-2 are enabled		

When a transport function is in a high flow state, the possible consequence is that its upstream function is in a low state because the transport draws more mass or energy out of its upstream function than the normal state. Vice versa, high volume of the upstream source or storage is a possible consequence for a low flow downstream transport. When reasoning downstream, high flow transport result in high volume in downstream function, while low flow transport result in low volume in downstream function.

3.2 Inference of Indirect Influence

To reason about indirect influence, it is necessary to separate the influencer or participant relation. Table 4 and Table 5 show the consequence inferences for indirect influence between functions.

The high volume in the downstream sink or storage will give a saturation effect, and therefore result in low flow of the upstream transport; whereas a low volume will draw more mass or energy from the upstream transport. When reasoning in downstream direction, high states of the upstream source or storage will result in high states of the downstream transport and the same with low states.

Indirect influence with a participant means the transport controls the flow level. Therefore, low volume in a downstream storage or sink will not give any consequence to its upstream transport, the same is the high volume source or upstream storage to downstream transport. Only when there's not enough mass or energy in the upstream function or saturation in downstream function will influence the transport state.

Table 4. Inference formulas of indirect influence with influencer relations

Inference upstream:		
	Cause	Consequence
	sto3, sin1 high volume	tra5, tra7 low flow
	sto3, sin1 low volume	tra5, tra7 high flow
Inference downstream:		
	Cause	Consequence
	sou1, sto1 high volume	tra1, tra3 high flow
	sou1, sto1 low volume	tra1, tra3 low flow
Assumption: tra1, tra3, tra5, and tra7 are enabled.		

Table 5. Inference formulas of indirect influence with participant relations

Inference upstream:		
	Cause	Consequence
	sto4, sin2 high volume	tra6, tra8 low flow
	sto4, sin2 low volume	N/A
Inference downstream:		
	Cause	Consequence
	sou1, sto1 high volume	N/A
	sou1, sto1 low volume	tra6, tra8 low flow
Assumption: tra2, tra4, tra6, and tra8 are enabled.		

3.3 Inference Pattern with balances

A balance function ensures that its input and output flow are equal. When reasoning about direct influence from transport to balance, the transport on the other side of the balance has to be taken into account. We first examine the balance with single in-port and single out-port. Table 6 and Table 7 show the inference formulas of single branch balance patterns.

In Table 6, we can see that the transport function is different from a storage function though they all have upstream and downstream transports. A balance has the ability to transfer the influence from a transport state to the other side if it has influencer relation with the other side transport. In such cases the state of the balance function won't be jeopardised (for they still regulate the flow so the input is equal to the output).

When the balance has an influencer relation with its further upstream transport, the downstream transport will influence across the balance. Otherwise the upstream transport will

remain the same flow, until the balance filled up when it has a low flow downstream transport.

Table 6. Inference formulas of direct influence with single branch balance pattern

Inference upstream:		
(diagram: tra1, bal1, tra2, in1, in2; tra5, bal3, tra6, in4, pa2)	Cause	Consequence
	tra2, tra6 high flow	tra1, tra5 high flow
	tra2, tra6 low flow	tra1, tra5 low flow
Assumption: bal1, bal3 are enabled and in normal states, and tra1, tra5 are enabled.		

(diagram: tra3, bal2, tra4, pa1, in3; tra7, bal4, tra8, pa3, pa4)	Cause	Consequence	
	tra4, tra8 high flow	bal2-3 unbalance	
	tra4, tra8 low flow	bal2-3 fill	*tra7, tra3 low flow
Assumption: tra3, tra7 are enabled and in normal states, and bal2-3 are enabled. *tra3 and tra7 will become low flow eventually when tra4 and tra8 are too low.			

Inference downstream:		
(diagram: tra1, bal1, tra2, in1, in2; tra3, bal2, tra4)	Cause	Consequence
	tra1, tra3 high flow	tra2, tra4 high flow
	tra1, tra3 low flow	tra2, tra4 low flow
Assumption: bal1, bal2 are enabled and in normal states, and tra1, tra3 are enabled.		

(diagram: tra5, bal3, tra6, in4, pa2; tra7, bal4, tra8, pa3, pa4)	Cause	Consequence	
	tra5, tra7 high flow	bal2, bal4 fill	
	tra5, tra7 low flow	bal2,bal4 unbalance	*tra6, tra8 low flow
Assumption: tra6, tra8 are enabled and in normal states, and bal2-3 are enabled. *tra6, tra8 will become low flow eventually when tra5 and tra7 are too low.			

Table 7. Inference formulas of direct influence with single branch balance pattern

Inference upstream:		
(diagram: tra1, bal5, tra2, in1, in2; tra3, bal7, tra4, in4, pa2)	Cause	Consequence
	bal5, bal7 fill	tra1, tra3 low flow
	bal5, bal7 leak	tra1, tra3 high flow
(diagram: tra5, bal6, tra6, pa1, in3; tra7, bal8, tra8, pa3, pa4)	Cause	Consequence
	No influence upstream	
Assumption: tra2, tra4, tra6, tra8 are enabled and in normal states, tra1, tra3, tra5, tra7 are enabled.		

Inference downstream:		
(diagram: tra1, bal1, tra2, in1, in2; tra3, bal2, tra4, pa1, in3)	Cause	Consequence
	bal1, bal2 fill	tra2, tra4 low flow
	bal1, bal2 leak	tra2, tra4 high flow
(diagram: tra5, bal3, tra6, in4, pa2; tra7, bal4, tra8, pa3, pa4)	Cause	Consequence
	No inference downstream.	
Assumption: tra1, tra3, tra5, tra7 are enabled and in normal states, tra2, tra4, tra6, tra8 are enabled.		

The downstream reasoning is similar with the upstream reasoning. One possible transport state is "leak", which is not a consequence for its neighbouring functions but a possible malfunction could cause other consequences. When a transport influences the states of balances, the balance may become unbalanced if the balance does not have influence on the other side transport.

When reasoning about indirect influence, the participant relation will prevent the abnormal state to propagate from the balance. (See Table 7) For a transport that have multiple in-ports and out-ports, we can make the same inference as shown in Table 6 and 7 as long as the other transports connected to the balance are assumed normal. When assuming all the in-port side transports are in normal state, the consequence of one abnormal in the out-port side transport may affect another out-port transport. The inferences are showed in Table 8.

Table 8 Inference across balance on the out-port side

Inference upstream:		
(diagram: tra4, tra1, tra10, pa2, tra7, bal1, bal2, in1, in2, in3; tra5, tra2, tra11, tra6, tra3, tra12, pa1, tra8, tra9)	Cause	Consequence
	tra1, tra7 high low	tra2, tra8 low flow
	tra1, tra7 low flow	tra2, tra8 high flow
(diagram: tra4, tra1, tra10, pa2, tra7, bal1, bal4, in1, in2, in3; tra5, tra2, tra11, tra6, tra3, tra12, pa1, tra8, tra9)	Cause	Consequence
	No inference	
Assumption: tra3-6, tra9-12 are enabled and in normal states and tra2-3, tra8-9 are enabled.		

The inference result for the in-port side (downstream) are the same as the out port side (upstream). Here still when the transport in focus sees the balance as a participant, the abnormal states will not propagate to that transport.

3.4 Producer-Product Relation and Mediate Relation

From this section, we will examine the cross level inference by starting with the two means-end relations namely producer-product (PP) and mediate that links flow structures directly with a function in another flow structure. The formulas are shown in Table 9.

Table 9. Inference formulas of PP and mediate relation

Inference from means to end		
(diagram: efs1, tra1, efs3, tra3; pp1, tra2, me1, tra4; efs2, tra2, efs4, tra4)	Cause	Consequence
	tra2, tra4 high low	tra1, tra3 high flow
	tra2, tra4 low flow	tra1, tra3 low flow
Assumption: tra1, tra3 are enabled. Transport function can be replaced by other functions, the inference result are the same.		

In normal situations, the states in the lower level function will give the same impact to the upper level function. However, the PP relation indicates adverse states between flow structures have no conceptual errors. If this kind of influence is observed in the real system, the reasoning rules should be altered for those relations.

3.5 Other Means-end Relation

Inference formulas for the remaining four means-end relations are more arbitrary than the causal reasoning within the same levels. The formulas for reasoning across means-end relations depend on the model object to some degree. This means that for any means-end relations that link from flow structures to objectives or threats, a success function has to be specified so that the reasoning can be done properly. The formulas are shown in Table 10.

Table 10. Inference about other means-end relations

(obj3 diagram)	Cause	Cnsq.	(obj4 diagram)	Cause	Cnsq.
	tra6 state1	obj3 true		tra8 state1	obj4 true
	tra6 state2	obj3 false		tra8 state2	obj4 false
(thr3 diagram)	Cause	Cnsq.	(thr4 diagram)	Cause	Cnsq.
	tra10 state1	thr3 exist		tra12 state1	thr4 exist
	tra10 state2	thr3 non-exist		tra12 state2	thr4 non-exist
Assumption: obj3-4 and thr3-4 are available.					

For each objective, some states of the main-function will cause it to be in a false state while the other states will cause its state to be true. This is the same for the threats. When the state set is specified, the reasoning can be done accordingly. The fulfilment of an objective can be the condition for enabling a function in another flow structure, while the existence of a threat may disable a function in another flow structure. The consequence reasoning from an objective or threat state to another function state is shown in Table 11.

Table 11. Inference formulas for conditions

(efs5/tra1/en1/obj1 diagram)	Cause	Consequence
	obj1 true	tra1 enable
	obj1 false	tra1 disable
(efs9/tra9/di1/thr1 diagram)	Cause	Consequence
	thr1 exist	tra9 disable
	thr1 non-exist	tra9 enable
Assumption: tra1 and tra9 are available.		

Up to this point, all the basic MFM inference formulas are introduced. Consequence reasoning of MFM can be done by combining all the formulas after a starting node and abnormal state for this node. It should be noticeable that the causes and consequences in each table are not interchangeable because they describe the functional causality of the physical system; the temporal aspect of causality is described implicitly in each rule that causes must occur prior to their consequences. Thus when a consequence becomes the cause within the same pattern in MFM, its influence towards the other flow function in the pattern can be different from the state that causes it. For example, flow function F1 with state (S^{F1}_1) will lead to the a possible consequence that flow function F2 to go into state (S^{F2}_1); then F2 with (S^{F2}_1) may lead to a possible consequence that F1 go into (S^{F1}_2), when S^{F1}_1 may differ from S^{F1}_2.

4. RULE-BASED SYSTEM

All the reasoning patterns and inference formulas introduced in section 3 can be implemented into a rule-based system as reasoning rules. Existing rule-based system development environments offers inference engines with reasoning algorithm that can perform the reasoning automatically. A rule-based software tool has been developed by the authors' research group by using Jess (Java Expert System Shell). A reasoning rule contains two parts. Jess uses an enhanced version of the Rete algorithm to process rules. Rete is a very efficient mechanism for solving the difficult many-to-many matching problem. Jess has many unique features including backwards chaining, inheritance capability, and working memory queries. It is suitable for applications when rules needed to be fired repeatedly based on newly generated information.

In Jess rules, the left-hand side (LHS) of the rules contain the conditions that need to be matched, while the right-hand side (RHS) of the rules produce the inference result if the left-hand side is matched. When running Jess applications, LHS of the rule need to be matched with knowledge base facts. For MFM reasoning, the LHS contains two parts, one is the MFM reasoning patterns, the other is a proposition indicate a state of one of the functions in the examined pattern. The reasoning engine will try to search the fact base for facts that satisfied all the conditions specified in the LHS, and when a match is found, the rule will be activated. Then the RHS suggest a new proposition according to the inference formula. The proposition that implemented in the software including 1) the information of the inferred function and state; 2) justifications that the inference based; 3) the rules that is used; and 4) the assumptions associate with the inference. All of above information is necessary to test the availability and truthfulness of the proposition.

The reasoning software works in two distinguishable steps. One is proposition generation and the other is reasoning maintenance. After a trigger (starting node) and the evidences (abnormal states) are registered to the reasoning system, the inference engine will first generate further propositions based on the rules (encoded patterns and formulas), and then test the availability of the propositions with all the assumptions. Sophisticated strategies and dependency structures are

included to test the propositions and retract the false or conflicted ones. All the propositions, after being generated and validated, are organized in a tree structure so that several casual paths can be identified. The assumptions and the dependency structures are useful for interpret the reasoning.

Another advantage of using Jess as programming language except its fast algorithm is that it is fully integrated with Java program and can reason about Java objects (as Jess facts) directly. The rule-based system developed by the authors' research group is now integrated with a MFM model editor, a Java based model building tool developed by Thunem (et. al. 2011) so that the reasoning result can be displayed graphically with the models.

5. EXAMPLE

The watermill example in Fig. 1 is used as an example to demonstrate how the rule-based system works. First we specify the "tra9" as the trigger with a high flow state. Then the reasoning package is run through the model editor. A screen shot of the reasoning result is provided in Fig. 4. From the figure, several possible consequence paths are generated according to the reasoning rules that introduced in Section 3 (the final consequences are listed in the upper-right panel in Fig. 4); one of the paths that lead to the top objective is highlighted in red.

The reasoning result from the consequence propagation can be used to predict system performance and is therefore very useful for plant supervision and operation support. Research is currently conducted by the authors to explore the requirement to design decision support system based on MFM. Multi-agent system is proposed as the framework for integrated different reasoning modules such as root-cause analysis and consequence analysis. (Zhang 2012)
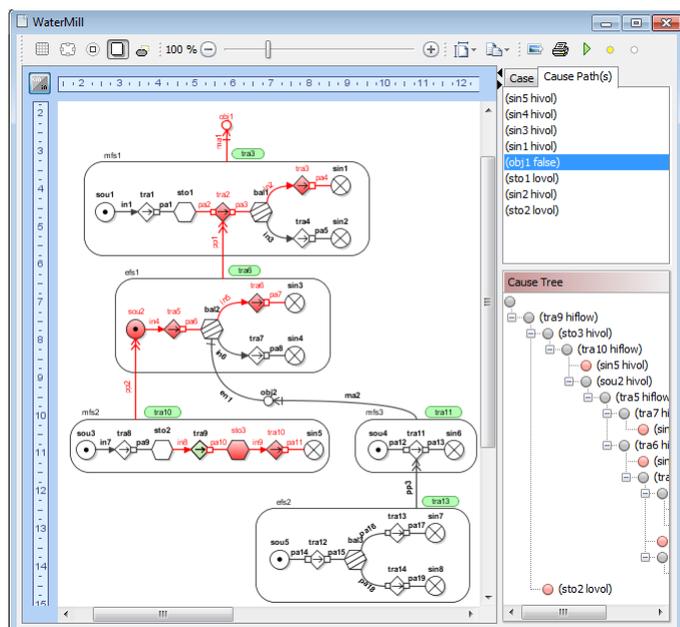

Fig. 4. Screen shot of the software tool

As can be seen from Fig. 4, the consequences propagation which is generated from one trigger node is organized in a tree structure. The consequence tree is verified with the evidences in the system after the tree is generated. Thus the invalid tree branches can be cut off and the affirmed states can be highlighted. The current software can perform consequence analysis with off-line reasoning cases. For develop on-line applications, reasoning depth has to be taking into considerations because the reasoning result should be compared with real event propagation in the physical system, so that the invalid prediction can be cut out. How to coordinate the online reasoning with plant data update is still an open issue.

6. CONCLUSIONS

The paper offers a detailed explanation of the consequence reasoning with MFM models. The purpose of the analysis is that the reasoning analysis can be programed as rule-based system which can be used for applicable domains, especially for situation assessment. More challenges are expected to emerge to develop online applications because they will certainly bring critical considerations such as time and resources into the system design. As mentioned in section 2.1, control functions and barriers are excluded in the scope of this paper because the reasoning concerns these two can be considered as independent analysis tasks. The current research (Lind et. al. 2012) also suggested multiple model representations for different operation modes. These will certainly bring more challenge into expanding the consequence reasoning rules. This paper provides a basis of the future development for consequence reasoning of MFM and its applications.

REFERENCES

Endsley M.R. Bolte B. and others (2003), *Designing for Situation Awareness, an Approach to User-Centered Design*, Taylor & Francis Inc., USA and Canada.

Gola, G. Lind, M. and others (2012). Functional Representation of Process Operation with Multilevel Flow Models for Diagnostic Decision Support, In *OECD Halden Reactor Project Report*. HWR-1059, 2012.

Heussen, K., Lind, M. (2010). Representing Causality and Reasoning about Controllability of Multi-level Flow-Systems. In *2010 IEEE International Conference on Systems, Man, and Cybernetics: Intelligent Systems for a Safe and Secure World*. IEEE.

Petersen, J. (2001). Situation Assessment of Complex Dynamic Systems Using MFM. In *Proceedings of 8th IFAC Symposium on Analysis, Design and Evaluation of Human-Machine Systems*, 645-650.

Lind, M. Yoshikawa, H. and others (2011), Multilevel flow modeling of Monju Nuclear Power Plant, In *Nuclear safety and simulation*, 2(3): 274-284.

Lind, M. Yoshikawa, H. and others (2012), Modeling Operating Modes for the Monju Nuclear Power Plant, In Proceedings of the 8th Inter-national Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies, San Diego, CA, United States, 2012.

Lind, M. (2011a). An introduction to multilevel flow modeling. *In Nuclear safety and simulation*, 2(1): 22-32.

Lind, M. (2011b). Reasoning about causes and consequences in Multilevel Flow Models, *Advances in Safety, Reliability and Risk Management - Proceedings of the European Safety and Reliability Conference*, ESREL 2011, 2359-2367.

Lind, M. (2011c). Control functions in MFM: basic principles. In *Nuclear safety and simulation*, 2(2): 132-140.

Lind, M. (2012). Modeling Safety Barriers and Defense in Depth with Mulitlevel Flow Modeling. In *Proceeding of First International Symposium on Socially and Technically Symbiotic Systems*, Okayama, Japan.

Thunem, H.P. Thunem, A.P. Lind, M. (2011) Using an Agent-Oriented Framework for Supervision, Diagnosis and Prognosis Applications in Advanced Automation Environments, In *Proceedings of ESREL 2011, European Safety and Reliability Association*.

Us, T. Jensen, N. and others (2011), Fundamental Principles of Alarm Design. In *Nuclear safety and simulation*, 2(1): 44-51.

Wu, J. Zhang, L. and others (2013), A novel failure mode analysis model for gathering system based on Multilevel Flow Modeling and HAZOP, In *Process Safety and Environmental Protection*, 91(1-2): 54-60.

Yoshikawa, H. Yang, M. and others (2011), Design of risk monitor for nuclear reactor plants, In *Nuclear safety and simulation*, 3(2):236-246.

Zhang, X. Lind, M. (2012) Agent Based Reasoning in Multilevel Flow Modelling, In *Proceedings of the first International Symposium on Socially and Technically Symbiotic System*, Okayama, Japan.