DTU Library

Agent Based Reasoning in Multilevel Flow Modeling

Lind, Morten; Zhang, Xinxin

Published in:
Proceedings of First International Symposium on Socially and Technically Symbiotic Systems

Publication date:
2012

Agent Based Reasoning in Multilevel Flow Modeling

ZHANG Xinxin*, and LIND Morten*

*, *Department of Electric Engineering, Technical University of Denmark, Kgs. Lyngby, DK-2800, Denmark*
(Email: xinz@elektro.dtu.dk and mli@elektro.dtu.dk)

Abstract: Multilevel Flow Modeling (MFM) is a modeling method used for modeling complex industrial plant. Currently, MFM is supported with a standalone software tool called MFM Workbench, which is equipped with causal-relation analysis and other functionalities. The aim of this paper is to offer a new design to launch the MFM Workbench into an agent based environment, which can complement disadvantages of the original software. The agent-based MFM Workbench is centered on a concept called "Blackboard System" and use an event based mechanism to arrange the reasoning tasks. This design will support the new development in MFM concept and its application in monitoring systems.

Keyword: Multilevel Flow Modeling; Multiagent System; Blackboard System

1 Introduction

Multilevel Flow Modeling (MFM) is a modeling method to represent an industrial installation as a system which provides the means required to serve purposes in its environment. [1] The concept of MFM is invented by Professor Morten Lind. [2] This modeling method is proved to be useful for analyzing systems with a broad range of scale, from simple heating system to nuclear power plant [3]. The syntax of the graphical language that MFM has adopted is setup as a template in software Microsoft Visio. This settlement also provides a transformation from a user friendly graphical format of MFM model into a computer oriented text format of MFM model.

Because of the utilization purpose of building MFM models and provide analytic solutions based on it, the expert system software, namely MFM Workbench, has been developed to support the functionality of the modeling. This software is designed to perform reasoning tasks upon MFM Models. For example, the current HAZOP application in the MFM Workbench take over the model information and create a knowledge base from this model, and thereafter response to the change of states in functions that composed the MFM and reason about the possible causal relations inside the model based on the pattern that independent of the model object. The analysis results are delivered in a tree of causal path that each branch of the tree can link to a root cause of the changing state. The diagnosis for the physical system can be drawn after relates the root causes function to its corresponding physical components in the system.

The MFM Workbench is a preliminary design firstly installed in Smalltalk language (originally called ABSTRACTION [4]), and then programmed in JESS (Java Expert System Shell) [5]. JESS is known to be an excellent tool specialized in developing expert system. The causal effect patterns are while defined in the JESS programmed MFM Workbench as the rule base, and the software can reason about the model in a very fast speed (millisecond level). However the MFM workbench is not completely developed for explore all the potential that MFM can provide. And the fact that JESS program lack of simple communication method also limited the further function development of the software (e.g. collecting environmental input from remote sensors and integration with other monitoring systems etc.). Therefore the current MFM workbench is not enough to support the reasoning in MFM, especially when the future function development is taken into consideration.

The center goal of this paper is to provide a suitable design to launch the existing MFM Workbench in a Multi-agent platform to support further development in MFM modeling and reasoning. The paper is organized as follows. After the introduction in Section 1, detailed motivation of the MFM Workbench innovation and justifications of using the chosen development environment are explained in

Received date: Month date, Year
(Revised date: Month date, Year)

Section 2. The new design of the Agent-Based platform is illustrated in Section 3. Preliminary prototype development is introduced in Section 4. Further Discussions is provided in Section 5. The final section concludes this paper and gives future work proposals. Note that to understand the tool design, background knowledge of MFM is required for the readers. This is not the main topic of this paper, yet it can be obtained from [6][7].

2 MFM model configurations and the need of new tool

2.1 Motivation of software innovation

The motivation of this paper is based on application purposes for MFM modeling. The original MFM workbench software is designed to perform single task for a single MFM model. For several realistic reasons, there might be needs to use several sub-models to represent a single physical system (especially with complex modern facilities). And there are also circumstances that different level of abstraction of the physical system should be adopted for the different purposes of the modeling. It is hard to deal with those situations base on the original MFM Workbench. The following sections will address the main motivations for the need of software tool innovation. The practical need for several MFM Model configurations such as model separations or model decompositions are the main driven force.

2.1.1 Decrease the workload of a single reasoning engine and increase the efficiency

Firstly, consider the HAZOP application in the original MFM Workbench. MFM model can be extremely complicated to handle by a single executor, and sometimes model separation might be required, especially for modern installations or distributed systems. The basic idea of MFM is to represent a given physical system in an abstract level. In general the complexity of the model is not directly related to the complexity of the physical system since that not all of the physical components of the system are necessary to be represented by a MFM function. For each physical system, models with different level of complexity can be developed according to the application's needs. In the diagnosis application, the model abstract from the physical system have to be reflecting back to the physical representation, so that the derived diagnosis can be understood by human operator and certain actions can be proceed according to the diagnosis. Therefore the MFM of the system used for diagnose application will be relatively complex. A modern system can be too complicated for a single inference engine to handle. Therefore a multiple-inference-engine solution is proposed, to use different reasoning engines and each reasons part of the system. For a complex industrial plant, the MFM model will probably include several flow structures and the fault might not be related to every flow structure in the model. If the software can organize the reasoning process and only trigger related inference machine to work on the case, the efficiency of the diagnosis is increased in certain cases.

In another scenario, when the system requires several MFM models with different abstraction level for different analysis purposes (e.g. both diagnosis and prognosis) or different operational situations, the workload can be too much for a single reasoning engine both to handle reasoning process and to organize different reasoning tasks in the same time. With a multi-thread environment, different inference engines can work in parallel with each other; both reducing the workload for each inference engine and increasing the efficiency of the software with a multi-processor hardware environment.

2.1.2 Increase scalability and flexibility

The physical system won't stay for the same for its whole operating time. Normally, the update of components can be expected from the life cycle of an industrial system. There is no guarantee that the newly updated part can be represented by exactly same MFM models. Another scenario is that in industry production, small systems might be gathered together to form a new system, or big system will be decomposed and reformed. When the physical systems experience those changes, their MFM model should be changed correspondingly. If different reasoning engines work independently, changes in one part of the model may not interrupt others reasoning process. And introduce new module to the system is fairly easy as well (only by running new engines).

2.1.3 Improve Integration capabilities

The MFM Workbench is designed as a stand along software. However to adopt this software into practical applications, integration with other operation support systems is essential. An agent-based system will complement the original workbench's lack of communication method. An agent is not a tool but a concept of a design structure, which integrates the idea of knowledge-based system (the existing MFM Workbench), self-management and communication. It will provide the system with a better interface when response with its environment to serve the modeling purposes.

Also, the existing functions of the MFM Workbench are developed in an intensive manner and it will be hard for integrated with future functions within the scope of MFM application. When launched in a new system, it will be desirable to modularize the MFM Workbench. This modularization process can also help divided the future efforts that will pull into the developing of the software.

2.1.4 Possibility to use different rule base

Generally, the rule base is defined by cause effects patterns. Those patterns can be used for all the models because its independency. That's part of the feature of MFM modeling. However with the current study of MFM concepts, the rule base will be expend greatly, adding new rules for control influence reasoning and consequence reasoning. Multiple- inference-engine provides possibility to use different rule base for different reasoning tasks.

2.2 Using JADE

A multi-inference-engine system is suggested in Section 2.1. The different inference engine should have the function that not only enable them to do the general reasoning for the model in its fact base, but also be able to make contact with other engines and make simple decisions. Therefore the concept of agent is also adopted. Thus, a corresponding software platform should be introduced to realize the design.

Knowledge based system supply with a good solution when building an agent, because it provides an accessible decision making model. The MFM workbench serves perfectly as the center decision models which help the agent to fulfill its task of analyzes and draw conclusions from MFM model. When a multi-inference-engine is designed, multiple agents have to work cooperatively. Therefore, on top of the knowledge-based system, each agent has to install a common communication protocol which enables the agents to communicate with each other.

JADE (Java Agent Development Framework) is a software framework for Multi-agent systems in Java. The JADE platform allows the coordination of multiple FIPA-compliant agents and the use of the standard FIPA-ACL communication language in both SL and XML. [8]

JADE can creates multiple containers, with each contains several agents. The set of containers can work in the same computing system or different systems, and together they form a platform. Each platform has a Main Container which holds two specialized agents: 1) The Agent Management System (AMS), who is the only authority in the platform who can create and kill other agents, kill containers, and shut down the platform. 2) The Directory Facilitator (DF) agent implements a yellow pages service which advertises the services of agents in the platform so other agents requiring those services can find them.

Integrated with the standard communication language and the predefined platform and management facility, JADE is a good solution to build Multi-agent system. In addition, JADE can be perfectly integrated with JESS code. Other multiagent programming languages might also be used to develop the multiagent system, but JADE is the only platform that can fully use the advantages of the existing MFM Workbench.

3 Multiagent MFM Workbench

3.1 Overall design

The design is based on the utilization level of MFM and it is good to describe the overall structure of the envisioned new design with a general working environment. MFM is normally used for modeling industrial plant and the diagnosis application with model separation is used as the set point for illustrate the general idea of the Multiagent MFM Workbench. Fig.1 shows the overall of the new design.

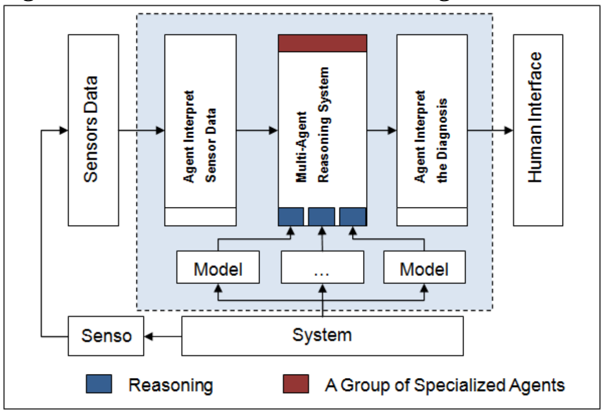

Fig.1 Overall design of the Multiagent MFM Workbench

For a diagnosis system in a certain industrial environment, the tasks including collecting environment data (commonly through sensors), interpreting the collected data, reasoning upon the situation, interpreting the diagnosis result and finally display it in a human interface in a manner that the information can be understand by the human operator. The diagnosis system components are present in Fig.1. For the new design, we also want each section of the MFM model to be handled by one agent, and agents with special purpose are also needed in the structure which will be explained later in Section 3.2. The system itself will be the model object and multiple models will be loaded. Then they will be assigned to different reasoning agents.

The interaction between diagnosis system and the environment is not the major concern in this paper, for the reason that these tasks is very much depending on the environment that the individual model object works in. With different systems, the data collection and information interpretation are completely different. The Multiagent diagnosis system, which is independent of the modeling objects, will be the focus. With the Model Building template in Microsoft Visio, we can already produce the model base on the knowledge provided by human experts who understand the process in the system and MFM modeling. The original MFM Workbench provides a rule base for the reasoning algorithm for the diagnosis. The core task in the design is to introduce a valid and efficient arrangement for the Multiagent system.

3.2 Blackboard System Solution

To fully use the existing recourses from the original software, the structure of original workbench need to be studied. This is showed in Fig.2. Reasoning in MFM models is based on dependency relations between states of objectives and functions. Model knowledge and states evidence are fed into the inference engine to produce causal tree based on dependency relations and pruning rules.

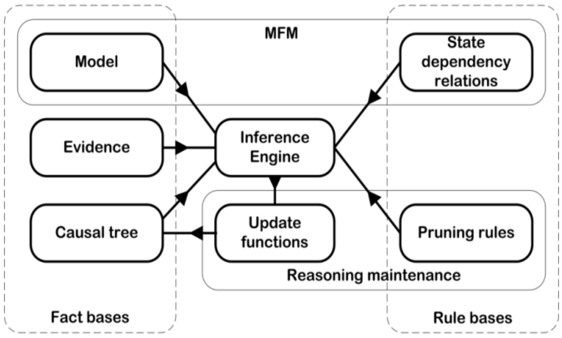

Fig.2 Overall reasoning functions of the original MFM Workbench [1]

To envision the expand workbench, we can see that there are several agents need to cooperate to work out a common solution for each case. And a mean of management is necessary for organizing the diagnose activities and synthesize the partial results. This resemble to the element of a blackboard system.

A blackboard system is an artificial intelligence application based on the blackboard architectural model, where a common knowledge base, the "blackboard", is iteratively updated by a diverse group of specialist knowledge sources, starting with a problem specification and ending with a solution. Each knowledge source (KS) updates the blackboard with a partial solution when its internal constraints match the blackboard state. In this way, the specialists work together to solve the problem. [10] Based on this concept, we can easily relate the elements in Blackboard system to the Multiagent MFM reasoning system.

Element Relation:

- Problem Specification => Facts Set 1 (Sensor Data Interpretation)
- Partial Solution => Facts Set 2 (Possibilities)
- Solution => Facts Set 3 (Diagnosis)

- Knowledge Source => Diagnose Agent (with Rule Base and sub-model)

Blackboard system uses event-based activation [11], which means that the system is triggered in response to blackboard and external events. Rather than having each KS (which are sub-model agents) scans the blackboard to find relative information, each KS informs the blackboard system about the kind of events in which it is interested. The blackboard system records this information and directly considers the KS for activation whenever that kind of event occurs.

In the Multiagent MFM Workbench application, each sub-model agents need to inform the blackboard agent that which function is contains in the local sub-model. The blackboard will issue the start command to the sub-model agents correspondently. For the efficiency reason, all the sub-model agents won't be triggered all together, but the sub-model agent that contains the changed function is to response first. Most probably is that only part of the system is responsible for the fault (if every function detected normal in one model part, then the causes of the disturbance in the system is not likely locate in this part). If this is the case non-relative agents won't run the diagnosis.

However, after some agent start reasoning, function on one side of the means- end relationship might be found as one root cause. If the reasoning result from the already triggered sub-model agents suggests this way, then the blackboard agent should be able to detect this situation, and then the information gathered the other agent who is relevant will be triggered.

This blackboard solution fits the scenario of the Multiagent based system and the communication procedure can be preliminarily concluded as in Fig.3.

The complexity of how the blackboard agent act as a reasoning maintenance facility in an inter-model scale is much depends on how sub-models are divided. The aim is to develop Multiagent MFM Workbench resembles the functionality of the original workbench, but have additional functions of separate model and coordinate causal path deducted from each sub-model.

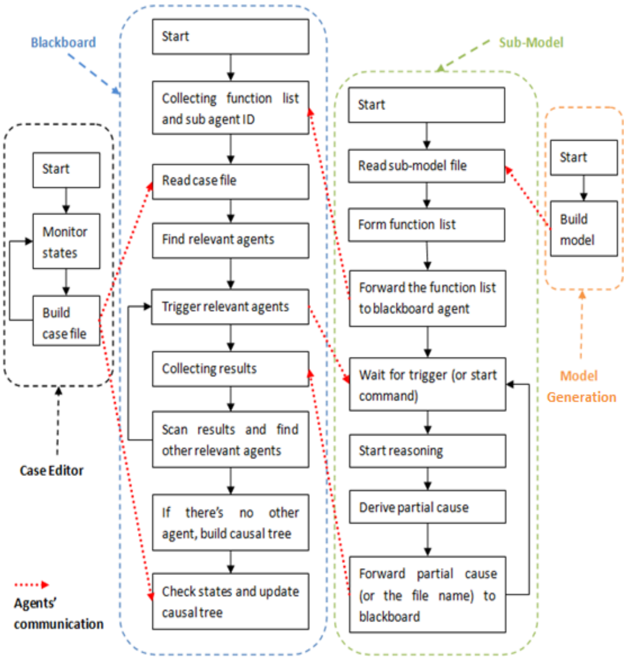

Fig.3 Communication process for Multiagent MFM Workbench

Besides the blackboard system, other multiagent planning mechanisms are also considered. For example the reasoning activity can be programmed fully autonomously and sub-model agents can resolve reasoning conflict through "discussion" with other sub-model agents. Or to another extreme the reasoning agents can be arranged strictly in orders. The former case will create chaos unless extremely sophisticated communication process being adopted. The latter case will detract the advantage of system efficiency. Blackboard system is a suitable solution gives a medium amount of freedom for the blackboard agent to organize the reasoning task but also keep the process in order.

3.3 Extension of the Original Workbench

The new design of Multi-agent structure is an extension of the original workbench, which can be explained in Fig.4. The fact base are separated, that means each agent maintain its own fact base. Sub-model agents deal with the partial model (or its own version of a whole model in other applications) that assigned to them. Then they update the deductive knowledge to the blackboard agent (or writing assumptions in a common file).

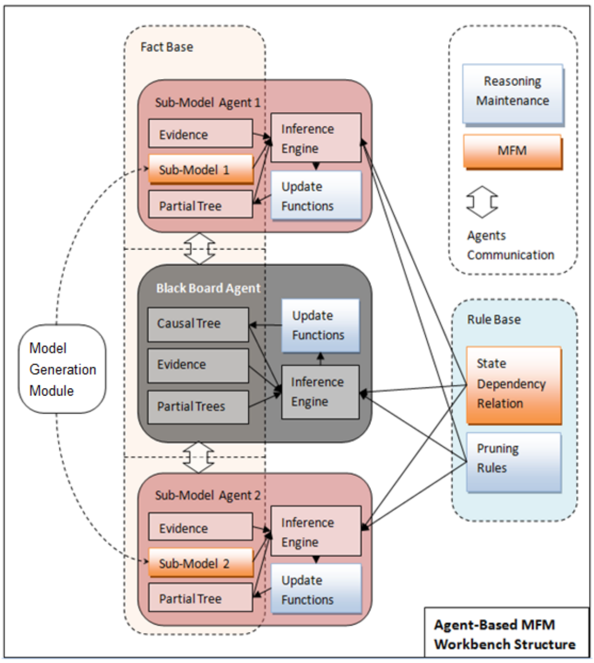

Fig.4 Agent based MFM Workbench Structure

Note that the architecture proposed in this section can be used for diagnosis with separated MFM model but not limited to this specified application. It is a generic framework for multiagent planning in organizing reasoning tasks.

4 Preliminary implementation

It is convenient to integrate Jess rule based system with JADE. During the integration JADE offers the environment and facilitates message sending/ receiving. Jess enables a declarative implementation of the decision module. [9] It can be described in Fig.5.

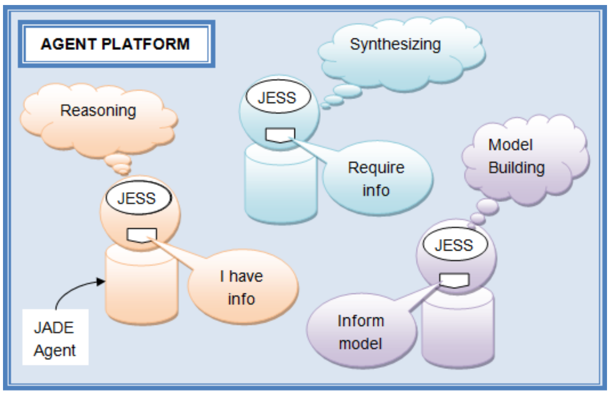

Fig.5 Integration of JADE agent with JESS

The general plan is to use JADE to send message and converse it to JESS Fact. This is depends on how much information need to be shared between agents. Short messages can be hand over directly in the message content, or the two agents are able to access the common file in which information is stored. The reasoning method and modeling process (MFM) model is very well developed. The MFM workbench software is available for this project. The installation tasks can be listed as follow:

a) Set up different types of Agents and their goals: Model generation Agent, Reasoning Sub-Model Agent, Blackboard Agents.
b) Set up communication procedure between different types of agent.
c) Equip each type of agents with its unique intelligent inference engine. (Modularize the original MFM Workbench.)
d) Test for new functions.

The programming detail is beyond the scope of this paper and can be found in [12]. The result of the implementation is that four types of agents are identified and the communications between different agents are being set up.

5 Further Discussions

As it is described in Section 3, the blackboard system is a generic solution for task arrangement in MFM reasoning applications. Recent study suggests that multiple tasks are required under different situations, such as different operation modes [13]. Hierarchical management structures can be built by using blackboard system as a sub reasoning unit to form bigger blackboard system. An example is showed in Fig.6. The top level blackboard system decides which analysis will be performed and then trigger the corresponding middle level blackboard to start the reasoning, the middle level blackboard will organizing an assigned task (either consequence analysis or root cause analysis) across different operation modes and trigger the corresponding lower level blackboard to start to organize reasoning in a specific mode with a specific tasks. The reason to put on this framework rather than using one standalone software is that for each blackboard, decision making model can be applied by design an individual knowledge base. The hierarchy can be flattened by adding more intelligence to the blackboard agents.

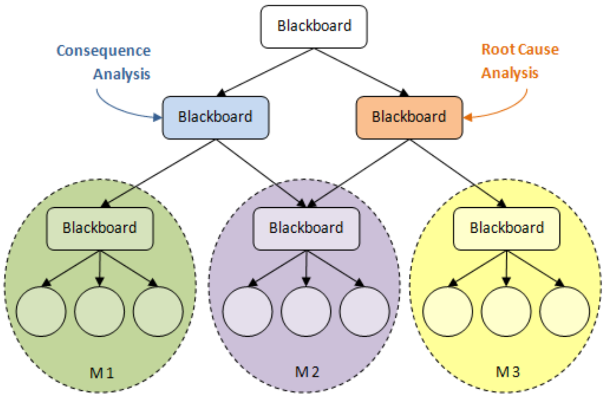

Fig.6 Blackboard System in hierarchy

6 Conclusion and future works

A Framework to develop the Multiagent MFM Workbench is designed based on the original MFM Workbench's functionality and deigned to cope with situations that the MFM model need to be manipulated in various ways. Some illustrative implementations are done to assist the discussion. However, it will also take times and efforts to implement the supplementary rule base to fully realize the envisioned functionality. After implement the Workbench in the Multiagent system, tasks of developing further functions or enrich the existing functions become much more independent from each other. This design serves as a good starting point of putting the MFM modeling in an Agent based environment. It has the full potential to become a software product that can easily adapt to different applications and integrated with other systems.

References

[1] LIND M.: Multilevel Flow Modeling: A Tutorial, Department of Electrical Engineering Technical University of Denmark, 2011

[2] FANG M.: MFM Modeling Method and Application, Department of Electrical Engineering Technical University of Denmark, 1994

[3] LIND M., YOSHIKAWA, H., JEORGENSEN, S. B., YANG, M., TAMAYAMA Kiyoshi., OKUSA K., Multilevel flow modeling of Monju Nuclear Power Plant, Nuclear Safety and Simulation, 2011, 2(3): 274-284.

[4] FANG M.: MFM Model Based Diagnosis, Department of Electrical Engineering Technical University of Denmark, 1994

[5] LIND M.: MFM Workbench Overview, Department of Electrical Engineering Technical University of Denmark, 2010

[6] LIND M.: An introduction to multilevel flow modeling, International journal of nuclear safety and simulation, 2011, 2(1): 22-32.

[7] LIND M.: 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, European Safety and Reliability Association, 2011: 2359-2367

[8] VAUCHER J., NCHO A.: JADE Tutorial and Primer,2003

[9] CARDOSO H. L.: Integrating JADE and Jess, 2007

[10] WEISS G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, the MIT Press, 2000:103-104

[11] WOOLDRIDGE M.: An Introduction to Multi Agent Systems, John Wiley & Sons 2005:

[12] ZHANG X.: Agent Based Reasoning in Multilevel Flow Modeling, Department of Electrical Engineering Technical University of Denmark, 2011

[13] JEORGENSEN, S. B., LIND M.: Modeling Operating Modes during Plant Life Cycle, 2012.