A MULTI-AGENT PREDICTION-BASED DIAGNOSIS SYSTEM

Erzsébet Németh[1], Rozália Lakner[2], Katalin M. Hangos[1]

[1] *Process Control Research Group, Computer and Automation Research Institute HAS, Budapest, Hungary*
[2] *University of Veszprém, Veszprém, Hungary*

Abstract: A multiagent diagnostic system is described in this paper. The proposed system enables the use of a combination of diagnostic methods from heterogeneous knowledge sources. The system is demonstrated on a case study for diagnosis of faults in a granulation process.

Keywords: multiagents systems, diagnosis, granulation process.

1. INTRODUCTION

For complex multiscale process systems that are difficult to model, a combination of model-based analytical and heuristic techniques is usually needed to develop a diagnostic system. The approach of multiagent systems (MAS) (Jennings and Wooldridge, 1998) what emerged in AI represents a promising solution for such a diagnosis task, being based on information from heterogeneous knowledge sources (Wörn *et al.*, 2004). A multiagent system can then be used for describing the behaviour and the structure of the elements in a diagnosis system. These elements include the system model, the observations, the diagnosis and loss prevention methods with each element being established through formal descriptions. This work investigates the use of the architecture and algorithms of multiagent systems for diagnosing faults in process plants. In particular we consider a granulation process and the advice to operators in order to reduce potential losses.

The significance of this work lies in a coherent fault detection and loss prevention framework based on a well-defined formalization of complex processes and the diagnostic procedures. Its novelty lies in the interesting combination of tools and methodologies that can be generalized to other process related applications.

2. MAIN PROCESSES IN FAULT DETECTION AND DIAGNOSIS

Early detection and diagnosis of process faults while the plant is still operating in a controllable region can help to avoid abnormal events and reduce productivity loss, therefore diagnosis methods and diagnostic systems have practical significance and strong traditions in the engineering literature.

The fault detection and diagnosis methods can be classified in two main categories: the analytical model-based methods (Blanke et al., 2003) and the logical and/or heuristic methods.

2.1 Fault detection, diagnosis and loss prevention

The diagnosis of process systems is usually based on symptoms. *Symptom*s are deviations from a well-defined “normal behavior”, such as $T_{low} = (T < T_{min})$ which is defined by using a measurable temperature variable T. In the case of a dynamic system the measurable quantities are time-varied, so the symptoms related to these variables will also change with time. In model-based fault detection and diagnosis one usually assigns a so-called *root cause* to every faulty mode of the system, the variation of which acts as a cause of the fault.
In the case of a fault it is usually possible to take actions in the initial phase of the transient to avoid serious consequences or to try to drive the system back to its original "normal" state. Dedicated *input signal(s)* serve for this purpose separately for each fault (identified by its root cause) where the preventive action is a prescribed scenario for the manipulated input signal.

2.1 HAZOP and FMEA analysis

Hazard and operability analysis (HAZOP) (Knowlton, 1989) is a systematic procedure for determining the causes of process deviations from normal behaviour and consequences of those deviations. The main idea behind HAZOP is that hazards in process plants can arise as a result of deviations from normal operating conditions.
Failure mode and effect analysis (FMEA) (Jordan, 1972) is a qualitative analysis of hazard identification (HAZID), universally applicable in a wide variety of industries. FMEA is a tabulation of each system component, noting the various modes by which the equipment can fail, and the corresponding consequences (effects) of the failures.
System components not only include the physical equipment but they can encompass software and human factors. It is regarded as one of the most comprehensive hazard identification techniques. HAZOP and FMEA provide a comprehensive analysis of the key elements that help constitute an effective diagnostic system

3. THE STRUCTURE OF THE MULTIAGENT DIAGNOSTIC SYSTEM

The proposed framework for a multiagent diagnostic system consists of an ontology design tool and a multiagent software system. The domain specific knowledge is represented as modular ontologies using the ontology design tool Protégé (Protégé, 2004). This knowledge is integrated into a multiagent software system where different types of agents cooperate with each other in order to diagnose a fault.

3.1 The ontologies of the diagnostic system

In order to facilitate the modularity and general applicability of the system, two set of **ontologies** are developed:

1. *A process-specific ontology* that describes the concepts of the processes in question similar to the general ontology for process systems given by OntoCAPE (Yang *et al.*, 2003).
2. *A diagnostic ontology* that contains the semantic knowledge on diagnostic notions, tools and procedures.

From the common part of the two different types of ontologies a real-time database is formed storing the values of process variables, actuator variables and related variables.

3.2 The applied multiagent software tools

JADE (Java Agent DEvelopment Framework) (JADE, 2005) has been chosen as the multiagent implementation tool, because it is an open source Java-based MAS development

kit that supports the Foundation for Intelligent Physical Agent (FIPA) specification agent standard and has integration facilities with the Protégé ontology editor and the Java Expert System Shell (JESS) (JESS, 2005).
JADE does not support inferencing techniques but it can be integrated with some reasoning systems, such JESS and Prolog. JESS is a rule engine and scripting environment written in the JAVA language. It possesses both a very efficient forward chaining mechanism using the Rete algorithm as well as a backward chaining mechanism, too.

3.3 The structure of the multiagent diagnostic system in JADE

Similar to the ontology classification, the **agents** of the diagnostic system belong to the following main categories:

1. *Process agents* that assist the user and the other agents in modelling and simulation of the process in question. This can be under different, faulty and non-faulty circumstances. Some types of process agents and their main tasks are as follows:
    - *Process output predictors (PPs):* prediction with or without preventive action(s).
    - *Prediction accuracy coordinator (PAC):* checks the accuracy of the prediction result and calls additional agents to refine the result if necessary.
    - *Model parameter estimators:* to each PPs. The PAC may call this agent when the accuracy of the agent is unsatisfactory.
2. *Diagnostic agents* that perform measurements, symptom detection, fault detection (Venkatasubramanian, 2003), fault isolation and advice generation for avoiding unwanted consequences. These agents may perform logical reasoning and/or numerical computations. Some types of diagnostic agents and their main tasks are as follows:
    - *Symptom generator and status evaluator:* checks whether a symptom is present or not.
    - *State and diagnostic parameter estimator (SPEs)*: advanced symptom generators that use several related signals and a dynamic state space model of a part of the process system to generate a symptom.
    - *Fault detectors (FDs):* use the services provided by SPEs or PPs and detect the fault(s) by using advanced signal processing methods.
    - *Fault isolators (FIs):* isolate the fault based on different techniques (fault-tree, HAZOP, FMEA, etc.).
    - *Loss preventors (LPs):* suggest preventive action(s) based on different techniques that have been used for the HAZID and remedial actions (HAZOP, prediction, etc.).
    - *Completeness coordinator:* checks completeness of the result (detection, isolation or loss prevention) and calls additional agents if necessary.
    - *Contradiction or conflict resolver (CRES):* calls additional agents in case of contradiction.

Beside the two main categories, the diagnostic system contains the following *Real-time agents* for controlling and monitoring the process environment:

- *Monitoring agents:* access and/or provide data from real world or from simulation.
- *Pre-processor agents:* detect the deviances what are the possible symptoms.
- *Control agents:* act as controllers.
- *Corroborating agent:* acts on request from diagnostic agents and provides additional measured values or information on request.

The main elements and the software structure of the proposed multiagent diagnostic system implemented in JADE can be seen in Figure 1.

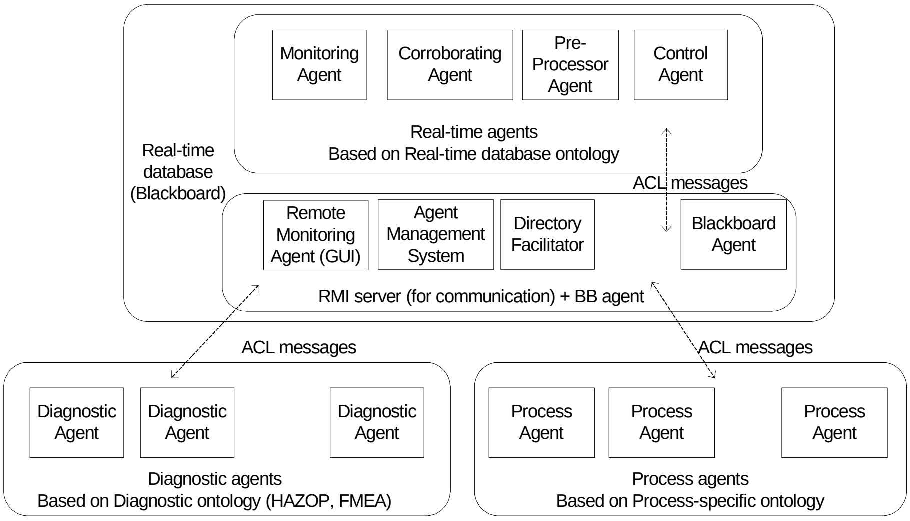

Fig. 1. The structure of the multiagent diagnostic system

4. CASE STUDY

The proposed methods and the prototype diagnostic system are demonstrated on a commercial fertilizer granulation system (Balliu, 2004).

4.1 The granulation process

The granulator circuit contains the granulator drum where fine feed or recycle granules are contacted with binder or reaction slurry. The binder can be added at various points along the axial direction of the drum, thus controlling moisture content in the drum and thereby growth. Growth occurs depending on a complex set of operational and property factors. Drying, product separation and treatment of recycle material then occurs.

4.2 Simulation results

In order to illustrate the operation of the proposed agent-based diagnostic system, only a part of the system, namely the agent-set, based on logical reasoning is demonstrated. The structure of this agent-sub-system can be seen in the left-hand side of Figure 2.

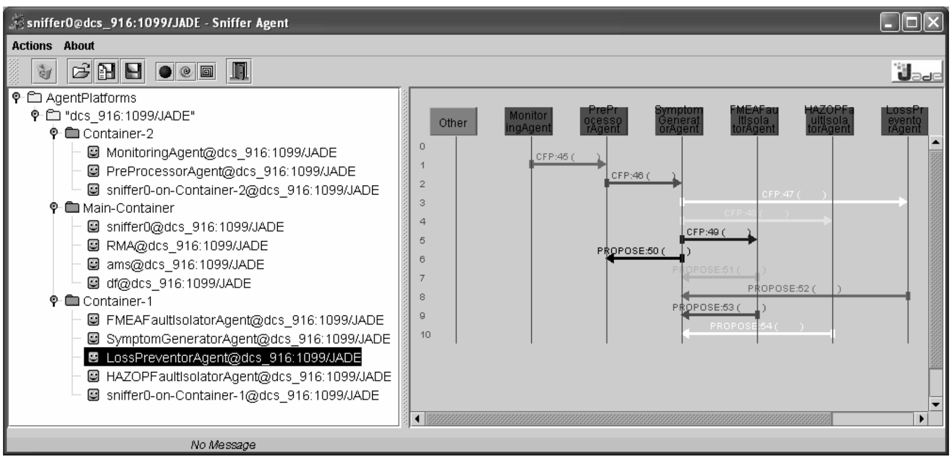

Fig. 2. The structure and the communication of the agent system

Apart from the built-in main-container's agents the agent platform contains two containers: one for the real-time agents (MonitoringAgent and PreProcessorAgent) and the other for the

diagnostic agents (SymptomGeneratorAgent, FaultIsolatorAgents - based on both HAZOP and FMEA analysis - and LossPreventorAgent). The main behaviour of these diagnostic agents is the logical reasoning based on heuristic knowledge (HAZOP, FMEA) of the diagnostic system with the help of the JESS rule engine.
The communication and the operation of the agent system can be seen in the right-hand side of Figure 2. Based on the variable-values supplied by the MonitoringAgent the PreProcessorAgent determines the deviances in the system. In case of a detected deviance the SymptomGeneratorAgent checks the presence of symptoms and calls the FaultIsolator- and LossPreventorAgent if it is necessary. These agents determine the possible faults and suggest preventive actions to avoid the hazard or operational problem that has been detected. This action exercises the communication and cooperation procedures defined among those agents.
The diagnostic process performed by the above agents is illustrated on the example of a symptom, when there is no fresh feed flow detected. This situation corresponds to the row of the HAZOP table seen in Table 1.

Table 1 A part of the corresponding HAZOP table

Guide Word	Deviation	Possible Causes	Consequences	Detection and Action
Fresh Feed Flow	NONE	(1) Feed hopper empty	❖ loss of production ❖ shift in granule size distribution ❖ decrease in recycle and output ❖ overdose of binder ❖ increased moisture content out of granulator	(a) detection by loss in production and feed sensors (b) level sensor on recycle hopper
		…	…	…

A part of these diagnostic agents' conclusions can be seen in Figure 3. where the messages about the operation of the LossPreventorAgent and the FaultIsolatorAgents are listed.
The above listed diagnosis and loss prevention results that are based on the corresponding row of the HAZOP table could be refined in the case of multiple causes and/or possible preventive actions if it were supplemented by the diagnostic results based on the FMEA analysis. This is done by the CompletenessCoordinatorAgent when necessary.

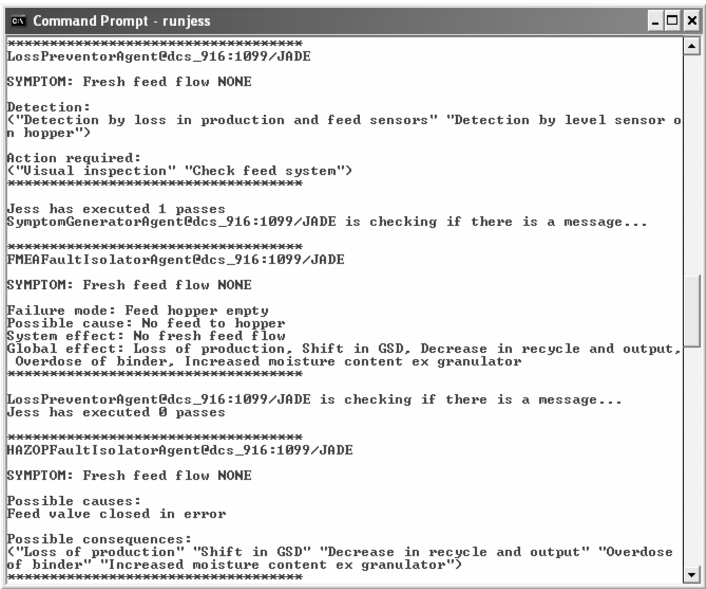

Fig. 3. A part of the diagnostic agents' conclusion

5. CONCLUSION

The prototype multiagent diagnostic system implemented in a Protégé-JADE-JESS environment has clearly shown the advantages of such a technology in building complex diagnostic systems based on heterogeneous knowledge sources. However, the interfacing of the system components with each other and with other system components, such as dynamic simulators is far from trivial. In addition, the reliability of such a complex software system has not reached a sufficient level to be fully deployed into an industrial application. Further work is needed to enhance inter-operability and to provide a more comprehensive set of analysis tools to attain high reliability systems.

REFERENCES

Balliu, N. (2004). An object-oriented approach to the modelling and dynamics of granulation circuits, PhD Thesis, School of Engineering, The University of Queensland, Australia 4072.

Blanke, M., Kinnaert, M., Junze, J., Staroswiecki, M., Schroder, J., Lunze, J., Eds (2003). *Diagnosis and Fault-Tolerant Control.* Springer-Verlag.

Cameron, I.T., Raman, R.(2005). *Process Systems Risk Management*. Elsevier.

JADE - Java Agent DEvelopment Framework. (2005). http://jade.tilab.com

Jennings, N., R.,Wooldridge, M., J. (1998). *Agent Technology.* Springer-Verlag, Berlin.

JESS, the Rule Engine for the Java platform. (2005). http://herzberg.ca.sandia.gov/jess/

Jordan, W. (1972). Failure modes, effects and criticality analyses. In: *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE Press, 30-37

Knowlton, R., E. (1989). *Hazard and operability studies : the guide word approach*, Vancouver: Chematics International Company.

The Protégé Ontology Editor and Knowledge Acquisition System, (2004) http://protege.stanford.edu

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., N. (2003). A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. *Computers and Chemical Engineering* **27**, 313-326

Wörn, H., et al. (2004). DIAMOND: Distributed Multi-agent Architecture for Monitoring and Diagnosis, *Production Planning and Control* **15,** 189-200

Yang, A., Marquardt, W., Stalker, I., Fraga, E., Serra, M., Pinol, D. (2003). Principles and informal specification of OntoCAPE, Technical report, COGents project, WP2.