# An Agent-Based Diagnostic System Implemented in G2

Attila Tóth, Erzsébet Németh and Rozália Lakner

Abstract—The basic agents and services of a multi-agent diagnostic system implemented in the G2 real-time expert system environment are described in this paper. The following basic services of an agent platform relevant to diagnostic applications have been implemented: a general agent object that is capable of performing cyclic, ticker (clock triggered) and one-shot behaviors, sending and receiving messages and a Supervisor agent. The implemented diagnosis relevant agents include a Monitoring agent, a Symptom agent and a Fault Isolator Agent.

The system is demonstrated on a simple case study for diagnosis of faults in a printing ink delivery system based on HAZOP and FMEA analysis.

## I. INTRODUCTION

For complex process systems, which are difficult to model and diagnose, a combination of model-based analytical and heuristic techniques is usually needed to develop a diagnostic system. Our general aim is to develop a diagnostic system which can be used for responding to abnormal conditions in a process by detecting abnormal events, diagnosing their causes and bringing the process back to its normal (safe) mode of operation.

One type of important information sources commonly used for fault detection and diagnosis are the detailed dynamic models [1] for parts of the process system or for certain operating modes. Additional heuristic sources can be the operational experiences elicited from operators and other plant personnel. The heuristic information can be collected with systematic identification and the analysis of process hazards, as well as the assessment and mitigation of potential damages using so-called Process Hazard Analysis (PHA). There are several methods used in PHA studies such as Failure Modes and Effects Analysis (FMEA) [2], Hazard and Operability Analysis (HAZOP) [3], Fault Tree Analysis (FTA) and Event Tree Analysis (ETA). The key challenge in this area is to integrate the above heterogeneous knowledge sources and to use them in an unified, holistic manner for diagnosis.

The approach of multi-agent systems [4], [5], which emerged in AI, represents a promising solution for such

E. Németh is with the Systems and Control Laboratory, Computer and Automation Research Institute, Budapest, Hungary (e-mail: nemethe@sztaki.hu)

R. Lakner is with the Department of Computer Science at University of Pannonia, Veszprém, Hungary (e-mail: lakner@dcs.vein.hu)

a diagnosis task, being based on the available information from heterogeneous knowledge sources. The multi-agent system can then be used for handling the system model, the observations, together with the diagnosis and loss prevention methods. The knowledge structures in our agentbased diagnostic system are being established through formal descriptions implemented in the form of ontologies [6].

The paper starts with a section on intelligent diagnosis, where the process model and the diagnostic information are introduced together with the notion of HAZOP and FMEA analysis. The next section deals with the principles of the agent-based realization of the diagnostic system discussing the knowledge representation, the agent-based system and the main tasks of diagnosis. Thereafter the real-time expert system implemented in G2 is introduced, where the main agents with their behaviors and the system architecture are detailed. Finally, the proposed method and a prototype diagnostic system based on the proposed approach is illustrated with a printing ink delivery system example.

#### **II. INTELLIGENT DIAGNOSIS**

#### A. The Diagnostic Information

Early detection and diagnosis of faults in a process system can help avoid abnormal events and reduce productivity loss. Therefore diagnosis methods and diagnostic systems have practical significance and strong traditions in the engineering literature [2], [3], [7]. 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 this purpose for each separate fault where the preventive action is a prescribed scenario for the manipulated input signal. The information available for the fault detection and diagnosis task is typically derived from a variety of sources, which have varying characteristics.

#### B. HAZOP and FMEA Analysis

In order to maintain the safety and the operability of complex process systems at the same time, it is important to apply HAZard IDentification (HAZID) procedures, such as Hazard and Operability Analysis (HAZOP) [3] or Failure Modes and Effects Analysis (FMEA) [2] to the plants. These procedures contain collected heuristic information about the plant (regarding its possible failures and their causes), which are knowledge gauged out of the operators and other experts.

The HAZOP analysis [3] is the most widely used methodology for HAZID used in complex process systems, e.g. in the chemical, and nuclear industries. HAZOP is a systematic procedure for determining the causes of process deviations

Manuscript received September 1, 2008. This research has been supported by the Hungarian Scientific Research Fund and the National Office for Research and Technology through grant no. K67625, which is gratefully acknowledged.

A. Toth is with with the Department of Computer Science at University of Pannonia, Veszprém, Hungary and also with the Continental Teves Hungary Ltd., Veszprém, Hungary (corresponding author, e-mail: Attila2.Toth@conti-engineering.com)

from normal behaviour and their consequences. The main idea behind HAZOP is that hazards in process plants arise as a result of deviations from normal operating conditions. The results of a HAZOP analysis are collected in a table, as can be seen on Table I. Detailed description of the HAZOP analysis procedure with illustrative examples are given in [3].

Guide Word	Deviation	Possible Causes	Consequences	Action Required
No	No Flow	Tank leak	No feed to pump (PA)	Install low level alarm on TA
			No feed to press / no product	Inspection of tank for leaks
		Tank not filled	No feed to pump (PA)	Check ink delivery
			No feed to press / no product	
Less	Less Flow	Line blocked	Pump cavitation Less feed to press / poor product	Install inlet strainer Install low level alarm on TA

TABLE I The results of HAZOP analysis

FMEA [2] is a qualitative analysis method for hazard identification, 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. FMEA focuses on individual components and their failure modes. An FMEA table structure is illustrated on Table II.

Component	Failure Mode	Failure Mode Causes	Local effects	System effects
TA	External break	Puncture by Projectile	Solvent Spill	Possible Fire/Explosion
				Other problems related to solvent spill
			No solvent supply	No product
	Operator damage			
	Faulty material			
	Corrosion			

TABLE II The results of FMEA analysis

## C. Blending HAZOP and FMEA Methods

Using a blended HAZID method, such as blending HA-ZOP and FMEA [8], a greater coverage of process and operational hazards can be achieved and a more comprehensive approach is available for root cause analysis and diagnosis. HAZOP and FMEA constitute two complementary and overlapping analysis methods. The connection between them can be seen on Fig. 1.



Fig. 1. Overlapping HAZOP and FMEA (HAZOP items are on the top and FMEA items on the bottom)  $% \left( {{\rm A}_{\rm A}} \right) = \left( {{\rm A}_{\rm$ 

## D. Process Models

The hierarchy of process models [1] with different granularity is driven by the level of details based on the requirement of the diagnostic analysis tasks. The model of the overall complex process system is structured in a hierarchical way where from top to down the plant, plantsection, equipment and component levels can be identified. As previously mentioned, ontologies [6] have been selected as formal tools for describing the syntactical and semantical relationship between the model ingredients together with the description of their physical connections. The structured description of the process model allows us to define the diagnostic knowledge which needed for diagnostic process in a structured, natural way together with the relationships between the two different kind of knowledge. The main structure of the ontologies can be seen later, on Fig. 3.

#### E. Diagnosis Based on HAZOP and FMEA

In a real-time fault detection and isolation, the diagnostic based on the blended HAZOP-FMEA method [9], [10] starts with detecting a *symptom*, that corresponds to a *deviation* in the HAZOP table. This initiates reasoning along the HAZOP table to find all of the *possible causes* of the *deviation*. These causes can be related to process variables which may be further *deviations* indicating further reasoning or can be connected to the components of the process system that is called *root causes*. The *root causes* are present in the FMEA table as (*component*) failure modes, so the reasoning can continue within the FMEA table.

The HAZOP table in combination with the FMEA table allows reinforcing of the root cause through checking that local effects indeed correspond to the original symptom. The steps of diagnosis based on the blended HAZOP-FMEA method can be seen on Fig. 2. The coloured arrows are used for highlight the different reasoning ways.



Fig. 2. Diagnosis based on the blended HAZOP-FMEA method

#### III. DIAGNOSIS USING AGENT PLATFORMS

The framework for a multiagent diagnostic system is a multiagent software system. The domain specific knowledge is represented as modular ontologies. This knowledge is integrated into a multiagent software system where different types of agents cooperate with each other in order to diagnose the faults. The principles of the agent-based realization of a diagnostic system are reported in [10].

## A. Agent-based Systems

An intelligent agent [5], [11] is an entity which can observe and act upon its environment and directs its activity towards achieving goals (i.e. it is rational). Intelligent agents can learn and may also use knowledge to help them achieve their goals. An multi-agent system (MAS) [5] is basically a set of cooperative agents and their environment.

A prototype diagnosis system has been developed by our group [10] in the form of a multi-agent system. It is built up from separate agents the behaviours of which realize subtasks of the diagnosis. In most of our cases the behaviours of the agents implement reasoning methods and the agents naturally communicate with each other.

The communication among the agents is performed through message passing represented in FIPA Agent Communication Language (FIPA ACL) [12].

## B. Diagnosis Subtasks

In our work the diagnosis of process systems [9], [10] is based on *symptoms*. Symptoms are deviations from a welldefined "normal behaviour", such as  $L_{high} = (L > L_{max})$ which is defined by using a measurable level variable L. For dynamic systems the measurable quantities are time-varying, so the symptoms related to these variables will also vary in time. The possible symptoms of the system are derived from the HAZOP and FMEA analysis' result tables.

The main tasks of our agent-based diagnostic system are the fault detection and diagnosis steps that are executed in a cyclic manner as follows:

- 1) *Performing measurements and symptom detection* The possible symptoms are determined based on the measured signals from the system.
- 2) Focusing and primary fault detection

Focusing on the proper hierarchy level and/or the part of the model the possible causes of the detected symptom are derived by model-based reasoning with the help of HAZOP knowledge. The reasoning stops at a "root cause" that is linked to a failure of a system component.

3) Fault isolation

The FMEA knowledge is used for fault isolation, where every possible "root causes" are investigated with model-based reasoning over the FMEA table to find symptom that is consistent with the observation.

This diagnostic procedure is performed by separate individual agents with different kind of behaviours. These agents interact and share information among each other. The process of the whole diagnosis is described in details in [10].

## C. Knowledge Representation in Ontologies

For the sake of common understanding and modularity, the domain specific knowledge is represented as modular ontologies. An ontology is a explicit formal specifications of the terms in the domain and relations among them [6]. Two related ontologies have been developed and implemented in our earlier agent-based diagnostic system [10] with the help of Protégé [13] ontology editor: the *Plant ontology* and the *Analysis ontology*.

The *Plant ontology* captures the main attributes and the relationships among the components and states of the process system. Any atomic (i.e. non-divisible) part of the system is regarded as a *SystemComponent* (such as instruments, pipes valves tanks, etc), while the measurable values are described by *ProcessStates* (such as level, pressure, temperature, etc). The class hierarchy is defined among the sub-classes of both *SystemComponents* and *ProcessStates* driven by the natural hierarchy of the components and states of the complex process system. A specific process system can be described as instantiated elements with defined attributes.

The Analysis ontology describes the structure of the diagnostic knowledge which can be applied in HAZID methods: in the HAZOP and FMEA tables. The knowledge from human expertise and operation about the behaviour of the system in case of malfunction, together with the causes, consequences and possible corrections is described here. From knowledge representation point of view, a row in the FMEA or in the HAZOP table constitutes an elementary structured knowledge item in *Analysis ontology*. The main attributes of the HAZOP and FMEA structures are related to the header of the proper table. In addition - as a careful syntactical and semantic analysis shows - common elements can be identified in the two structures, as it is seen in Fig. 1.

#### D. Experiences with an Agent-based Diagnostic System

In our previous work [10], JADE (Java Agent DEvelopment Framework) [14] has been chosen as the multi-agent implementation tool for the realization of our diagnostic system. JADE is an open source Java-based multi-agent system (MAS) development kit that supports the Foundation for Intelligent Physical Agent (FIPA) [12] specification agent standard and has integration facilities with the Protégé [13] ontology editor and the Java Expert System Shell (JESS) [15] that is used for diagnostic reasoning.

The prototype multi-agent diagnostic system implemented in a Protégé - JADE - JESS environment has clearly shown the advantages of such a technology. All of these implementation tools are based on JAVA, so they can be integrated each other easily in case of small, simple systems. In case of large, complex systems, however, the integration is not obvious. In addition, interfacing the Protégé - JADE - JESS system components with either a real-time system or a dynamic simulator is not trivial.

Moreover, the reliability of a complex process system implemented in Protégé - JADE - JESS software environment has not yet reached a sufficient level to be fully deployed into an industrial application, partially because of the prototype and open-source nature of these components. Therefore, we need to work on to enhance the inter-operability and to attain robust and high reliability implementation of the agent-based diagnostic system for large-scale industrial applications. This is the reason, why we have decided to re-implement our agent-based diagnostic system in another tool, in G2.

## **IV. IMPLEMENTATION IN G2**

For improving the reliability and robustness of our agentbased diagnostic system, it has been re-implemented in an another software environment that contains only one implementation tool. Our choice is Gensym Corporation's G2 [16], which is a real-time business inference engine platform. The G2 platform combines real-time reasoning technologies, including rules, workflows, procedures, object modelling, simulation, structured natural language, and graphics, in a single development and deployment environment.

G2 enables to deliver intelligent solutions that dramatically improve the consistency, efficiency, flexibility, and quality of operations. G2 applications can follow multiple lines of reasoning and analyze large amounts of data and numerous trends concurrently.

The main advantages of G2, like object-oriented technology, rule-based inferencing, real-time and built-in simulation facilities give a promising solution for implementation of our agent-based diagnostic system in a homogeneous software environment.

## A. The Knowledge Representation

The knowledge representation in the G2 implementation of our diagnostic system was accomplished in the same structure as introduced in III-C. Utilizing the object-oriented nature of G2, the structure of the two core ontologies (*Plant ontology* and *Analysis ontology*) can be described by a class hierarchy defining the common characteristics of the objects. The application-specific knowledge can be defined using instances, which are objects representing specific occurrences of the same class. The class hierarchies of the *Plant ontology* and *Analysis ontology* with some actual instances can be seen in Fig. 3.

With the help of G2's built-in real-time simulator, the actual process model can be realized and used for dynamic simulation both in normal and faulty modes.

## B. The Multi-Agent System

G2 does not possess built-in agents and an agent simulation tool, so special classes had to be defined for the implementation of the agent-system. The main steps in creating an agent system were the description of a general agent, then to define different kind of agents and their behaviours, and finally achieving the communication and coordination among the actors in the multi-agent system.

1) Agents and Their Supervisor: An actual agent is defined as an instance of the general agent object definition. The behaviour of an agent is realized by procedure(s), which is a property of the agent instance. The basic implemented behaviours of the agents are as follows:

• the *cyclic behaviour* is designed to never complete and its operation is performed repetitively until the agent terminates,

- the *one-shot behaviour* is designed to complete in one execution phase and its operation is performed only once,
- the *ticker behaviour* operates periodically and waits a given period after each execution.

Besides the basic agent-behaviours, a rule-based reasoning mechanism is also needed in our diagnostic system. G2 supports this facility with the help of its built-in inference engine.

In our multi-agent system there is a central *supervisor agent* who is responsible for the connection of all of the other agents by providing communication interface and supervising their interaction. This supervisor agent is a special subclass of the agent object definition with special methods realizing the necessary properties of this agent. The other agents are responsible for doing subtasks of the diagnostics in cooperation with each other.

2) Communication: The communication among the agents is performed through message passing. A message, the necessary element of the agent communication, is realized by the instance of the general message object definition. For standardization of the agent communication the message structure implements a standard agent communication language, the FIPA ACL [12]. The supervisor agent is responsible for the communication of the agents.

# C. System Architecture in G2: agents and their tasks

The main architecture of the agent system can be seen on Fig. 4. The main agents and their behaviours are seen in Fig. 4.



Fig. 4. The agents in G2

1) The Supervisor Agent: The main task of the Supervisor agent is to monitor and moderate the operation of the agent system, supervise the communication, and it provides a graphical console for agent communication.

2) The Monitoring Agents: are real-time agents, that receive and handle real-time data based on theirs ticker behaviours and handle messages based on their cyclic behaviours.

3) The Diagnostic Agents: cooperate with each other to realize the diagnostic procedures. There are stand-alone agents for generating symptoms and isolating faults. These agents work using their one-shot behaviours initialized by



Fig. 3. The class hierarchy of the Plant and Analysis ontologies in G2

messages, and may perform logical reasoning. The diagnostic agents and their main tasks are as follows:

- The *Symptom generator* uses the set of non-permissible deviations, and checks whether a symptom is present or not.
- The *Fault isolators* work in the case of the occurrence of a symptom to isolate the fault based on HAZOP and/or FMEA knowledge.
- The *Completeness coordinator* resolves the conflicts and checks the completeness of the result (fault detection, fault isolation) and calls additional Fault isolator agents if necessary.

# V. CASE STUDY

The aim of the case study is to illustrate the diagnostic procedures and the operation of the agent based diagnostic system implemented in G2.

## A. A Solvent Delivery Process of a Printing Ink System

The simplified flowsheet of the process system that is used for delivering solvent to a printing ink plant can be seen on Fig. 5. The process system consists of tanks, pipe lines, valves, motor and pumps.

Solvent is delivered to the bulk tank TA via delivery pump PB, valves VD and VE. When required, solvent is pumped from the bulk tank TA via pump PA driven by motor MA, valve VA and line L to the systems feed tank TB. The flow of solvent into the feed tank TB is controlled by float valve VB.



Fig. 5. Solvent delivery process system

From the feed tank TB, solvent is supplied to the printing press system via valve VC.

## B. Simulation Results

For applying the proposed agent-based diagnostic system in G2, a process model for the solvent delivery system was developed using the built-in simulator facility of G2. The elaborated model can be used for modelling both the normal and the faulty modes. The flowsheet of the system implemented in G2 is illustrated on Fig. 6.

The real-time simulator is connected to the agent-based diagnostic system with the help of *Monitoring Agents* that monitore and store the necessary variable-values. Based on the value of the state-variables, the *Symptom Generator Agent* determines the deviances in the system and in case of a deviance checks the presence of symptoms and informs

the *Completeness Coordinator Agent*. This calls the *HAZOP Fault Isolator Agents* and/or the *FMEA Fault Isolator Agents* to determine the possible faults and suggest preventive actions if available.

The diagnostic process performed by the above agents is illustrated on the example of a deviation, which is the high level of bulk tank TA. One of the possible causes of this fault is the more solvent delivery as can be seen on the right-hand side bottom of Fig. 6.



Fig. 6. Printing ink example implemented in G2

# VI. CONCLUSIONS

A diagnostic expert system implemented in G2 is described in this paper that is able to integrate conventional engineering dynamic models and heuristic operational experiences. The proposed model-based fault detection, diagnosis and loss prevention system is able to

- handle uncertain and/or heuristic knowledge together with partial dynamical information,
- automatically focus on the faulty part of the system,
- handle multiple fault hypotheses and improve the diagnosis,
- give advice for recovery strategies.

The real-time expert system, G2 is selected as the implementation tool of the proposed agent-based diagnostic system. We are in the early part of this work, but we have some promising initial results on the field of ontology and agent development and implementation for this diagnostic system.

## VII. ACKNOWLEDGMENTS

The authors thankful to Professor Katalin Hangos for her guidance and suggestions. This research partially supported by the Hungarian Scientific Research Fund and the National Office for Research and Technology through grant No. K67625, which is gratefully acknowledged. The authors acknowledge the support from the Australian Research Council under Linkage Grant LP0776636.

#### REFERENCES

- K.M. Hangos, I.T. Cameron, Process modelling and model analysis, Academic Press, 2001
- [2] W. Jordan, "Failure modes, effects and criticality analyses", In: Proceedings of the Annual Reliability and Maintainability Symposium, IEEE Press, 30-37, 1972
- [3] R.E. Knowlton, *Hazard and operability studies: the guide word approach*, Vancouver: Chematics International Company, 1989
- [4] N.R. Jennings and M.J. Wooldridge, Agent Technology, Springer-Verlag, Berlin, 1998
- [5] J. Ferber, Multi-Agent Systems An introduction to distributed artifical intelligence Addison-Wesley, Oxford, 1999
- [6] T.R. Gruber, "A Translation Approach to Portable Ontology Specification" *Knowledge Acquisition* 5: 199-220, 1993
- [7] V. Venkatasubramanian, R. Rengaswamy, and S.N. Kavuri, "A review of process fault detection and diagnosis Part II: Qualitative models and search strategies", *Computers and Chemical Engineering*, Vol. 27, pp. 313-326, 2003.
- [8] I.T. Cameron, B. Seligmann, K.M. Hangos, R. Lakner and E. Németh, "The P3 Formalism: A Basis for Improved Diagnosis in Complex Systems" In: *Chemeca Conference*, 2007, Melbourne, Australia
- [9] R. Lakner, E. Németh, K.M. Hangos and I.T. Cameron, "Agent-based diagnosis for granulation processes" In: 16th European Symposium on Computer Aided Process Engineering (ESCAPE-16) and 9th International Symposium on Process Systems Engineering, Elsevier (Eds.: W. Marquardt, C. Pantelides) pp. 1443-1448, Garmisch-Partenkirchen, Germany, 2006
- [10] R. Lakner, E. Németh, K.M. Hangos and I.T. Cameron, "Multiagent realization of prediction-based diagnosis and loss prevention", In: 19th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE'06), Annecy, France, LNAI 4031, pp. 70-80, 2006
- [11] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, NJ: Prentice Hall, 2003
- [12] "FIPA ACL Message Structure Specification", Doc. Number: SC00061G 2002,
  - URL: http://www.fipa.org/specs/fipa00061/SC00061G.pdf
- [13] "The Protégé Ontology Editor and Knowledge Acquisition System", 2008
- URL: http://protege.stanford.edu [14] "JADE - Java Agent DEvelopment Framework", 2008
- URL: http://jade.tilab.com [15] "JESS, the Rule Engine for the Java platform", 2008
- URL: http://herzberg.ca.sandia.gov/jess/
- [16] "Gensym Corporation, G2 Real-Time Rule Engine", 2008 URL: http://www.gensym.com