THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2012, VOL. 35, NO. 2, ISSN 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS

ABSTRACT MULTI-AGENTS USED IN INDUSTRIAL FAULT DIAGNOSIS

Marius-Cristian Costache, prof. dr. ing. Viorel Mînzu

Department of Automatic Control and Electrical Engineering, "Dunărea de Jos" University of Galați, Domnească Street, no. 47, Galați, ROMANIA

Abstract: Industrial diagnosis systems aim to anticipate the occurrence of failures or, if failures have occurred, to detect them and identify their cause, based on observable symptoms captured by sensors from the process. In case of large and complex industrial systems, characterized by dynamic situations and extensive data, such diagnosis system must satisfy requirements for modularity, flexibility and adaptability. Thus, it is difficult to tackle the problem through the strength and capability of a single intelligent entity. This paper introduces a multi-agent cooperative architecture customized for computer-supported fault diagnosis, as a solution to better solve the variety of problems in modern, real-time applications.

Keywords – Intelligent Agent, Complex systems, Cooperation, Fault Diagnosis, Coordination, Decentralized systems, System architectures.

I. INTRODUCTION

The overall objective of a monitoring and diagnostic system is to increase the reliability and the availability of complex applications. In order to build up such a system, a lot of generic tasks like data acquisition, knowledge management, communication between different units, performing diagnoses and displaying the diagnostic results have to be performed. The possibility to use different diagnostic engines in parallel is often a must for identifying faults. They may refer to different components of the industrial application or they may apply different diagnostic mechanisms.

The utilization of a multi agent paradigm to provide these features allows developing a flexible and reliable monitoring and diagnostic system. The flexibility enables the diagnostic system to react efficiently to modifications in the state of the industrial application. Complex plants are generally speaking not being modified by changing the process or the workflow itself. The flexibility is required as soon as the plant diverges from the normal, fault-free behavior into an unpredictable faulty state. In this case, a flexible diagnostic system is required to adapt automatically to the changes of the supervised application.

Together with a set of predefined ontologies and data processing mechanisms, different generic diagnostic algorithms and a set of reusable software agents, a standardized multi-agent architecture for diagnostic purposes is formed. The multi agent paradigm which warrants the flexibility, the predictability behavior, the extendibility and a cost effective development of a diagnostic system is most suitable to meet industrial requirements.

Historically, multi-agent systems technology was invented as a sub-field of distributed artificial intelligence, which itself is a sub-area of artificial intelligence. Today, the term 'multi-agent systems' is used to refer to all types of systems composed of multiple (semi) autonomous components (see [1]). This approach represents a new and promising solution to problems outlined above. It is assumed that each agent is capable of a range of useful problem-solving activities in its own right, has its own aims and objectives and can communicate with others (see [2]). As distributed systems, multi-agent architectures have the capacity to offer several desirable properties over centralized systems (see [3]):

An overview of distributed artificial intelligence applied in industry is given in [4]. This paper summarizes the industrial needs for distributed artificial intelligence, with particular attention to manufacturing systems, planning and control. Using the multi-agent approach is feasible because agents are most suitable for modular, decentralized applications, in constant change, poorly structured and complex. Regarding these characteristics, agent technology can obtain a more robust and adaptable solution than other software paradigms can (see [5]).

Although the industrial multi-agent systems have a great impact on complex processes, in terms of greater profitability and better management, developing such systems is sometimes difficult and poses many challenges (see papers [6], [7], [8], [9], [10], [11], [12]).

A platform for different diagnostic processes running in parallel used to handle large systems is described by Fabre (see

This paper was recommended for publication by Adrian Filipescu

[13]). This paper aims to model a plant system as a graph of interacting subsystems. Thus, theoretical analysis is provided in order to treat these subsystems and to bring them together for processing in the context of a large-scale system.

A multi-agent approach is presented by Letia for monitoring and diagnosis of spatially distributed technical systems (see [14]). The agents interact through concepts of beliefs, desires and intentions and it provides an example of diagnosis for a computer network while dealing with the symptoms spread throughout components.

The DIAMOND project presented in [15] uses a multi-agent system that is customized for specific requirements of monitoring and diagnosis. Communication between agents is based on CORBA, together with inter-agent communication language ACL, defined by the FIPA (Foundation for Intelligent Physical Agents). This system is not based on explicit diagnostic algorithms but on establishing a standardized multi-agent architecture.

In the paper [16] the power systems are regarded as autonomous systems and the defaults' diagnosis is treated using intelligent agents.

The paper [17] presents a theoretical approach of the use of Multi Agent Systems in the case of Model Based Diagnosis. In a large dynamical system, it is often impossible to maintain a model of the whole system. Instead, several incomplete models of the system have to be used to detect possible faults. These models may also be physically distributed. A Multi Agent System of diagnostic agents may offer solutions for establishing a global diagnosis. If a separate agent for each incomplete model of the system is used, establishing a global diagnosis becomes a problem of cooperation and negotiation between the diagnostic agents.

This paper proposes a fault diagnosis system implemented by a multi-agent system (MAS). The particularity of this system consists in the fact that there is only one intelligent agent responsible with the detection of a particular fault in the process' running. The implementation of the MAS is exemplified using a case study: the wastewater treatment process.

II. FAULT DIAGNOSIS ABSTRACT MULTI-AGENTS

We propose a fault diagnosis abstract multi-agent system (FDAMA) which consists of a variable number of agents, each responsible for a particular fault that may occur in the system under diagnosis. Such an agent is called Fault-Agent (FA). FDAMA includes a Supervisor-Agent (SA) which starts and coordinates the diagnostic process and a Rule-Agent (RA) which infers certain application-specific rules in order to discriminate between multiple faults and determine which fault is the initial cause of others.

Each FA is taught to identify the footprint of a single fault in the input data set collected from the system under diagnosis. The Supervisor Agent (SA) is designed to fulfill the position of manager in the decision process. The interaction between different Fault-Agents creates a dynamic environment, allowing finding a solution in an efficient manner.

When several fault-agents announce the supervisor agent that each of them identified the corresponding fault, the supervisor agent assumes the task to apply meta-heuristics in order to refine the final result. For example, multiple faults can coexist in the system and only one (original) fault has caused the others. FDAMA presents the results based on interaction and cooperation within the team of agents. Below is presented the structure of the proposed system:

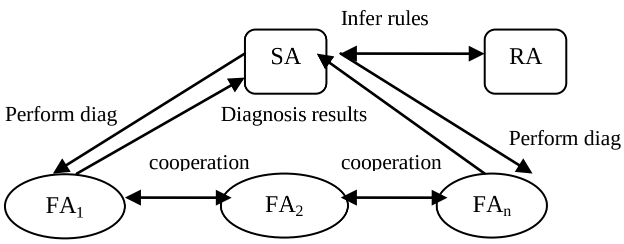

Figure 1. Structure of FDAMA

III. CASE STUDY: A SIMULATED ABSTRACT PROCESS

In the following section, we will describe the structure of FDAMA and the way it perform the diagnosis in the case of a simulated, abstract industrial application described below.

This abstract process is designed and modeled using a general Abstract Systems Simulator, implemented by the authors. Here is the general scheme of the abstract system:

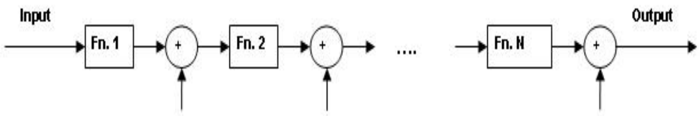

Figure 2. Abstract system scheme

Each block from the scheme models a functional component of the industrial system, with its input, state function and output. External inputs (from other sub-systems) are also taken into consideration by the accumulators near each component.

All abstract components can introduce response delays and can be set to operate either in the "normal" mode or in a specific fault mode (one of the many possible faults later to be identified by FDAMA).

A possible set of functions for the abstract system components is the following:

$$\begin{aligned} F_1(x) &= a_1x^2 + b_1x + c_1 \\ F_2(x) &= a_2x^2 + b_2x + c_2 \\ &\ldots\ldots\ldots\ldots \\ F_n(x) &= a_nx^2 + b_nx + c_n \end{aligned} \tag{1}$$

where: $a_1 \ldots. a_n$, $b_1 \ldots. b_n$, $c_1 \ldots. c_n$ are the coeficients of the functions which describe the system's components.

All outputs of the individual components become inputs for the Fault Diagnosis Abstract Multi-Agents system proposed in this paper, as can be seen in the figure:

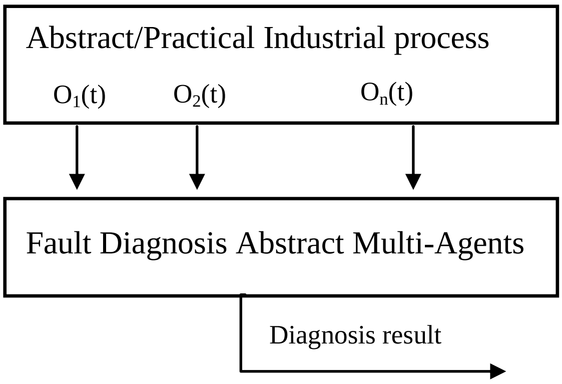

Figure 3. FDAMA and process coupling scheme

In the chosen application, we modeled the faults set of each component *i* as $\{F_{i1}, F_{i2}, \ldots F_{im}\}$, each representing the component's output blocked at a certain value. These theoretical faults gain specific practical meaning in a real industrial process, i.e. a pump blocked at a certain flow level, a sensor indicating a constant 0 value, etc.

In the following sections, we present some aspects concerning agents' implementation in FDAMA.

IV. FDAMA IMPLEMENTATION

IV.1 Fault-Agents (FA)

The goal of each Fault-Agent (FA) is to identify the existence of a particular fault from the ones the system is trained to recognize. In the current implementation, it is based on a neural network trained with a lot of data that is characteristic to the considered fault, which the FA can later recognize in the online data.

In the example chosen (the abstract industrial process), the training data is composed of *n* vectors corresponding to the *n* output parameters of the system's components (O_1, O_2,…O_n), which are inputs for the FDAMA. The online data batch contains the n-tuples whose elements belong to the n parameters at a given moment of time. So, each parameter determines a vector of real values providing the value of the corresponding parameter at different moments of time. Therefore, FDAMA will have a sequential running mode; the data is delivered with a constant time period. This period is predefined according to the system that is subject to diagnosis.

Taking into account the data at present time and a configured number of previous n-tuples, the FA can ascertain the existence of a fault. When it recognizes the fault it was trained for, it sends to the SA an informative message specific to the multi-agent framework chosen for implementation.

IV.2 Supervisor-Agent (SA)

The Supervisor-Agent (SA) has a specific goal: it facilitates the diagnosis process of the multi-agent system. First, the SA is responsible for collecting input data, processing them in the format recognized by FAs and providing a batch of data to all FAs. Once this phase is completed, the SA starts the diagnosis process and expects to receive results from FAs. This shows that parallel processing is used to achieve diagnosis (the FA team of agents together with a unique SA).

After receiving individual results, the role of SA is to send a cooperation request to ADs who identified their faults. Thus, the ADs form a team (coalition), aimed at refining the initially recognized results. To avoid reporting "false" faults, the SA has a procedure to reconfirm the faults based on specificities of the system under diagnosis.

The SA gives the final result in a form comprehensible to the human user, stating which faults were diagnosed at the current moment of time, having the confirmation-pending status. After the confirmation procedure is completed, the AS eliminates the unconfirmed faults from the report and raises alarms for the confirmed ones. It also offers a list of solutions in order to fix them.

IV.3 Rule-Agent (RA)

The Rule-Agent (RA) has the role to conclude which of the identified faults is the cause of perturbations (in case one single fault generates a waterfall of faults in the next system components), by implementing some meta-heuristics specific to the system subjected to diagnosis (in this example, the abstract industrial process). The RA is initially trained with the system's causal relations between multiple possible faults, in order to "know" later which fault is the real cause of others. Thus, the RA increases the priority of such major faults (and decreases the priorities of the induced ones) which will apear in the final Diagnosys report generated by the SA.

IV.4 FDAMA Implementation

Each neural network (from ADs and RA) must first be trained to recognize a fault and after this phase, it can become operational and identify those faults in the online batch data. In the current implementation, the Matlab Neural Network Toolbox offers various learning algorithms: descent or conjugate gradient methods, Levenberg - Marquardt algorithm, resilient back-propagation algorithm (chosen option).

The training is performed with data batches specific to every fault, under SA coordination. During the training phase of FDAMA, the SA designates which FA will cover the current fault and will train it to recognize that fault. The training batches contain the n-tuples (O_1, O_2,…O_n), corresponding to a single fault which was deliberately induced using the designed Abstract Systems Simulator. For example:

The training stage, as well as the diagnosis process, is performed based on a memorized sequence of data batches and not on instant values.

Training the FDAMA with a single fault is performed therefore by forcing the output of the FA responsible for current fault on 1 and all the other FAs on 0. The training process requires a batch of data for normal behavior (when all agents respond with 0) and a batch when multiple faults are present (and several FAs respond with 1).

In the operational phase, each FA receives the data batch from the SA, consisting of the n time vectors (O_1, O_2,…O_n) collected by the sensors of the system under diagnosis (the abstract industrial process). The goal of every neural network from FAs is to detect the presence of the fault it was trained for, and when this occurs, to inform the SA through a message.

For real-time online diagnosis, the system creates a "sliding window" on a time horizon $k \cdot \Delta t$, which encompasses the last k data n-tuples that are accessible to all FAs for recognition. At the next moment of time (t +1), the window moves forward on the temporal axis.

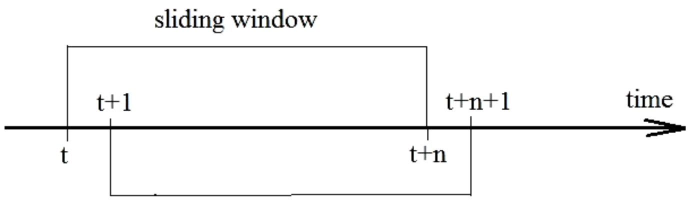

Figure 4. The sliding window

In fig. 5 is presented the functional diagram of the diagnosis process:

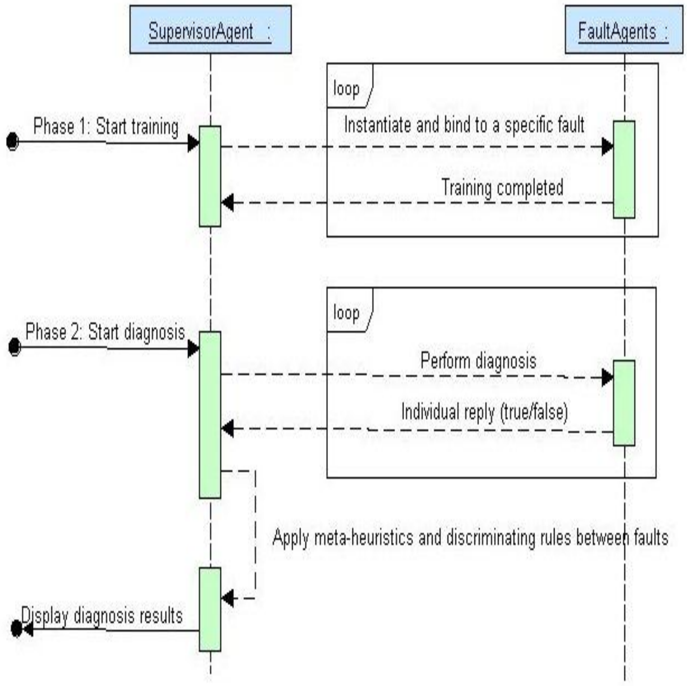

Figure 5. FDAMA implementation

The multi-agent platform chosen for the implementation of FDAMA is JACK™ Intelligent Agents[1], which is an agent-oriented software development environment, built upon Java™ programming language.

The chosen MAS platform extends Java to support Agent Oriented programming in the following ways: it defines new base classes, interfaces and methods; it provides extensions to the Java syntax to support new agent-oriented classes, definitions and statements; it provides semantic extensions (runtime differences) to support the execution model required by an agent-oriented software system.

JACK is built on top of the BDI model (Belief-Desire-Intention), and every agent from the proposed diagnosis system exhibits reasoning behavior under both proactive (goal directed) and reactive (event driven) stimuli. Each agent has: a set of beliefs about the world (its data set); a set of events that it will respond to; a set of goals that it may desire to achieve (either at the request of an external agent, as a consequence of an event, or when one or more of its beliefs change; a set of plans that describe how it can handle the goals or events that may arise.

The main reasons why we chose the above mentioned framework are: ability to act autonomously; high-level representation of behavior – a level of abstraction above object-oriented constructs; flexibility, combining pro-active and reactive behavioral characteristics; real-time performance;

[1] A temporary evaluation license was used

suitability for distributed applications; ability to work co-operatively in teams.

V. PRACTICAL RESULTS

In order to test the FDAMA, the Abstract Systems Simulator was used, which can provide either "nominal" data or altered data by one of the faults that FDAMA was trained to recognize. In all simulated fault scenarios subjected to testing (each containing a single fault from the list at Chapter III), the proposed system was able to identify and isolate the fault present in the input data batch in less time than using a traditional expert system approach, due to parallel processing and distributed computing superiority brought by the intelligent agents' software paradigm.

The validity and effectiveness of the proposed multi-agent system was demonstrated by applying it to a practical size model. It should be noted that the proposed system is able to determine the existence of a fault using only local information, making it a promising approach for large complex models.

The future work concerning FDAMA is to improve the performance and knowledge of the multi-agent system in order to deal with the problem of multiple faults, occurring simultaneously. Depending on the system under diagnosis, information from human experts can be collected and then implemented in the automated system, enabling it to decide which fault is causing the other, what solution to offer in order to correct that fault, which fault can be considered "noise" induced in the system, or if all identified faults can be valid at a given moment of time.

The results are conclusive to state that the proposed multi-agent approach is effective for designing a fault diagnosis system.

VI. CONCLUSION

This paper proposed a multi agent diagnosis system as a solution for detecting faults that occur in industrial processes. The presented architecture encompasses a set of agents each responsible for one of the faults known by the system, together with a supervisor agent which facilitates the cooperation process in the team of agents and to refine the initial results. Each agent responsible for a fault contains, in the chosen implementation solution, a neural network capable of identifying the fault in the input data batch.

The practical results have shown the efficiency of the proposed system. We are currently comparing the results obtained with similar diagnosis systems. Each FA becomes after the training stage a local expert on its field, recognizing a single fault.

As future work directions, we are studying the case of multiple faults being recognized at the same time, discerning what fault caused the others, determining the relations between multiple faults initially discovered in the system subjected to diagnosis (in order to determine if they all coexist in the system or one implies the others).

ACKNOWLEDGEMENT

This work was partially supported by the strategic grant POSDRU/88/1.5/S/61445, Project ID61445, cofinanced by the European Social Fund – Investing in People, within the Sectorial Operational Programme Human Resources Development 2009-2012.

REFERENCES

[1] N.R. Jennings, M.R. Wooldridge, "Agent Technology Foundations, Applications and Markets", Springer-Verlag, ISBN: 3540635912, 1998;

[2] N.R. Jennings, L.Z. Varga, R.P. Aarnts, J. Fuchs, and P. Skarek, "Transforming standalone expert systems into a community of cooperating agents", Int. Journal of Engineering Applications of Artificial Intelligence, 6(4), p317-p331, 1993.

[3] P. Stone, and M. Veloso, "Multi-agent systems: a survey from a machine learning perspective", Autonomous Robots, p345-p383, 2000.

[4] Parunak, H. Van Dyke, "What can Agents do in Industry, and Why? An Overview of Industrially Oriented R&D", In *CEC, CIA98*, 1998.

[5] Parunak, H. Van Dyke, "Practical and Industrial Applications of Agent-Based Systems", 1998.

[6] W. Mark, J. Dukes-Schlossberg, and R. Kerber "Towards very large knowledge bases", IOS Press/Ohmsha, msterdam/Tokyo, chapter Ontological commitment and domain specific architectures, 1995

[7] D. Mylaraswamy, "DKIT: a blackboard-based, distributed, multi-expert environment for abnormal situation management", PhD thesis, Purdue University, 1996.

[8] D. Mylaraswamy, V. Venkatasubramanian, "A hybrid framework for large scale process fault diagnosis", Computers and Chemical Engineering 21, S935–S940, 1997.

[9] H. Vedam, "OP-AIDE: an intelligent operator decision support system for diagnosis and assessment of abnormal situations in process plants", PhD thesis, Purdue University, 1999.

[10] H. Vedam, S. Dash, and V. Venkatasubramanian, "An intelligent operator decisionsupport system for abnormal situation management", Computers and Chemical Engineering 23: S577–S580, 1999.

[11] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, K. Yin, "A review of process fault detection and diagnosis", part 1, 2, 3, Computer & Chemical Engineering 27(3): 293–346, 2003.

[12] V. Venkatasubramanian, "Prognostic and diagnostic monitoring of complex systems for product lifecycle management: Challenges and opportunities", Computers and Chemical Engineering 29: p1253–p1263, 2005.

[13] E. Fabre, A. Benveniste, and C. Jard, "Distributed Diagnosis for large discrete events in dynamic systems", In: *P*roceeding of the IFAC world congress, July 2002.

[14] I. A.Letia, F. Craciun, Z. Kope, and A Netin "Distributed Diagnosis by BDI Agents", In: IASTED International Conference: Applied informatics, 2000.

[15] M. Albert, T. Längle, H. Wörn, A. Kazi, and all, Distributed Architecture for Monitoring and Diagnosis. EU Esprit Project No. 28735, 2001.

[16] C. Rehtanz, Autonomous Systems and Intelligent Agents in Power System Control and Operation, Springer-Verlag, 2003

[17] M. Barbu, S. Caraman, G. Ifrim, "Identification of a wastewater treatment process using sensitivity method", International Conference on Automation Quality Testing and Robotics, Cluj Napoca 2010.

[18] N. Roos, A. Teije, A. Bos, C. Witteveen, "Multi Agent Diagnosis: an analysis", Maastricht Economic Research Institute on Innovation and Technology, Infonomics Research Memorandum series, June 2001