AQTR 2004 (THETA 14)
2004 IEEE-TTTC - International Conference on Automation, Quality and Testing, Robotics
May 13 – 15, 2004, Cluj-Napoca, Romania

DISTRIBUTED AGENT-BASED MONITORING AND DIAGNOSIS FOR LARGE-SCALE SYSTEMS

Honoriu Vălean, Tiberiu Leția, Adina Aştilean

Technical University of Cluj-Napoca
24-26 Barițiu st., Cluj-Napoca, Romania
Tel: +40-264-401822, Fax: +40-264-592055
{Honoriu.Valean, Tiberiu.Letia, Adina.Astilean}@aut.utcluj.ro

Abstract: Large-scale systems can be usually seen as highly connected network enabled systems. Such systems consist of embedded sensors in networked subsystem that can transmit data to remote monitoring stations. The paper deals with large-scale systems, using a distributed hardware for the monitoring and diagnosis system and implementing a modular multi-agent architecture for detecting the faults and for identifying the causes of failures. The multi-agent architecture is proposed to solve all the problems, starting with the sensor data access up to the diagnosis and visualization within graphical user interfaces.

Key words: Distributed agents, Monitoring, Diagnosis, Large-scale systems.

1. INTRODUCTION

The monitoring and diagnosis of industrial plants becomes very important due to the fast growth of the systems complexity and because of the need of production optimization, for competing within the industry. For reducing the failure occurrences and costs, a powerful monitoring and diagnosis system is indispensable. A continuous monitoring and diagnosis of the plant will provide a set of benefits, such as [2],[3]:

- the improvement of operational safety;
- the improvement of availability by reducing time to diagnose failures,
- the improvement of confidence in system serviceability,

For the most plants, the initial cost of the hardware and software of the monitoring and diagnosis system is easily offset by the savings in ground support and maintenance within a usually short period of time. Such savings in ground support costs are achieved by:

- reduced manual diagnostic costs, as consequence of the automatic identification of failures;
- detection of failures that manifest only under certain specific operational modes or stress conditions, e.g., at a certain resonant frequency. The embedded monitoring and diagnosis system will be able to detect and isolate a failure when it manifests itself in actual operation;

- the capacity of the system to identify failing components, thus, the unscheduled repairs will be reduced.

The complex industrial plant can be seen as set of different physical units, manufactured by different producers, having different behaviors. There are often different diagnostic methods and different knowledge about the unit behaviors in some particular situations. Furthermore, there are different diagnostic methods for the same type of unit.

For solving the previous problems, the paper presents the hardware infrastructure and the multi-agent framework of a monitoring a diagnosis system. The system is also extended with learn capabilities, for dealing with unknown failures.

2. THE HARDWARE INFRASTRUCTURE

The monitoring and diagnosis system is proposed to be built by interconnected cells. A cell structure is presented in Figure 1.

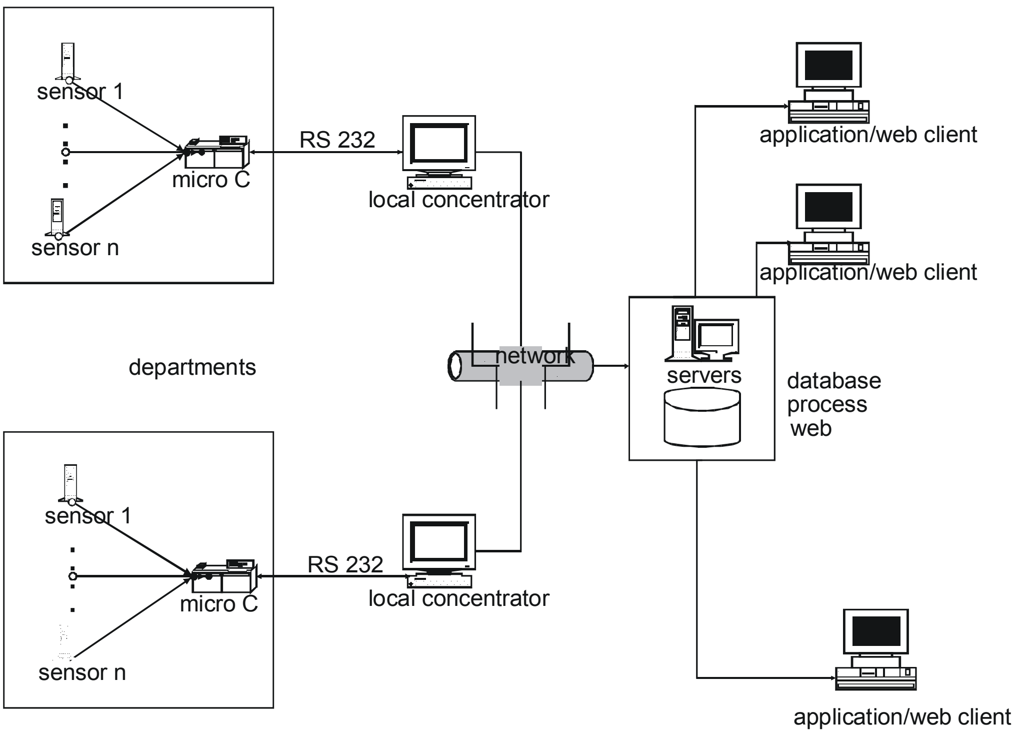

Figure 1. A cell structure for monitoring and diagnosis system

A cell can cover the entire plant, or some departments. Two or more cells can be interconnected over the TCP/IP protocol. Since the cell acts as Intranet, the cell interconnection acts in Internet.

For receiving data from the sensor, the infrastructure uses a set of micro controllers. The micro controller is a very cheap and powerful solution for distributing local process interfaces and controllers. They can be programmed to execute a set of data acquisition/distribution, control and communication tasks.

The result of data acquisition and control information is processed on more powerful computers. A process server software communicate with the micro controllers by reading/writing data on the serial interface, store data in a database and serve the client programs, located in its own LAN. The server doesn't communicate on Internet, because of the security problems.

The accessibility through the WWW to the information became a necessity in the last few years. Thus, a web server allows the web clients to interrogate the database, containing knowledge used by diagnosis algorithms. Using a general web server for interfacing the web, not a thread in the process server, provides more reliable security access. This is because the time execution purposes, the process server are designed as a service of the operating system.

More details about the hardware infrastructure can be found in [7].

3. THE DISTRIBUTED AGENTS ARCHITECTURE

The software architecture proposed in the paper uses software agents, for solving the monitoring and diagnosis problem. Each of the agents is responsible for a specific task. The entire production process aspects and functionalities are encapsulated into different agents, able to interact with the overall framework.

Because of their high generality, all the agents are designed and built not very different from the agents proposed in [1]. Two distinct software units compose each agent: the *Core*, responsible with the specific tasks performed by the agent and the *Wrapper*, responsible with the communication between agents.

The *Core* unit is different from an agent to another, depending on the specific task executed by the agent. Agents can perform different tasks, such as: monitoring, diagnostic, learning, conflict arbitration, etc. The monitoring tasks are usually similar for all industrial environments and the associate agents are not totally dependent on the application. Other tasks are specific to the application and have specific implementations.

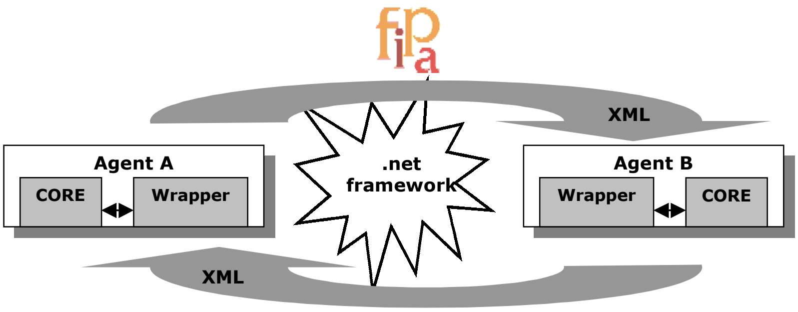

Figure 2 Agents structure and interconnection

The *Wrapper* unit is identical for all the agents, and allows the interaction over the .net framework. Other methods are available to implement the interaction between agents. More details about these methods can be found in [6]. The communication between agents is implemented with respect of FIPA specification [4]. FIPA specifications represent a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent.

All the information exchanged between agents, is generated-used by the *Core* and encoded-decoded by the *Wrapper* in a XML structure (Figure 2) [8].

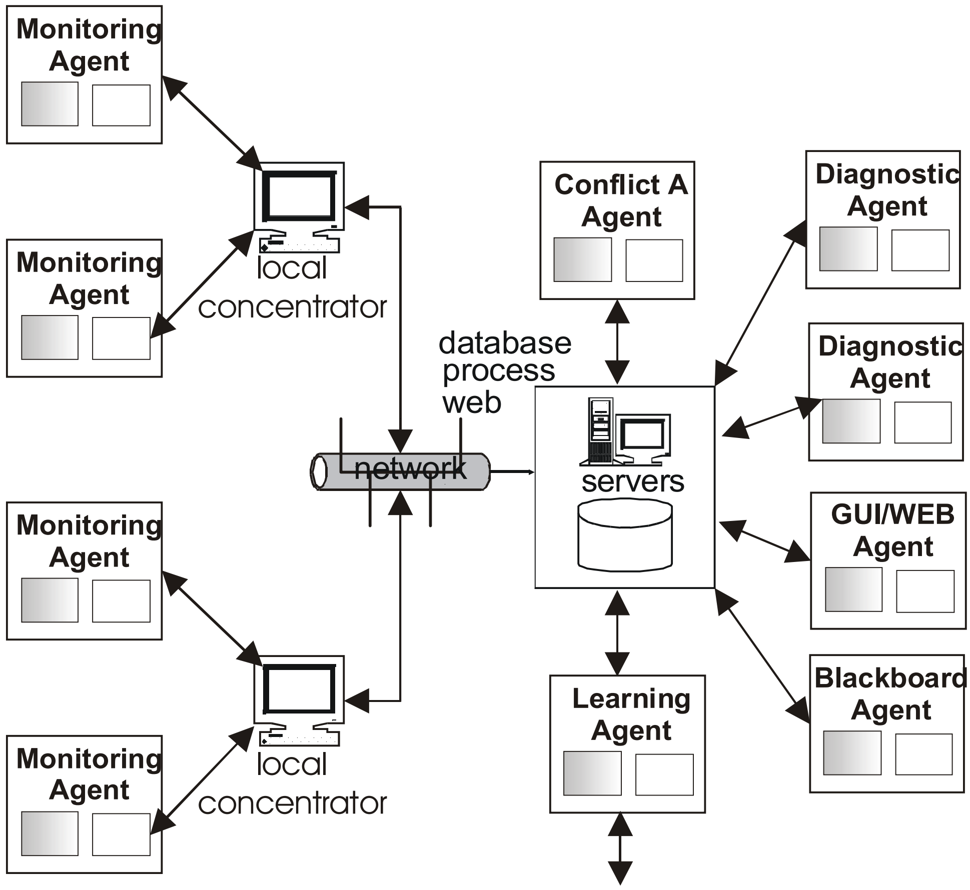

Figure 3 Agent types

The agent types and their localization are presented in fig. 3. As can be seen, 6 types of gents are present.

- *the monitoring agents* – implements the interface between the physical sensors and the software monitoring and diagnosis system. These agents receive the measurements from the sensors and make them compatible with the data format handled by the other agents. The monitoring agents are also able to initiate a diagnosis if a measurement is irregular;
- *the diagnostic agents* – receives the measurements provided by the monitoring agents and using monitoring algorithms and knowledge stored in database identifies the state of the plant. In case of failure, these agents provide information about the cause. In case of devices designed with BIST, these two agents can be merged and the wrapper will interact with the BIST result. For diagnostic of the overall process, the diagnostic results reported by different diagnostic agents are taken into account. Some of these results can to be in conflict [5];
- *the conflict arbiter agent* – receives the results of diagnostics provided by different diagnostic agents and resolve possible contradictions;
- *the blackboard agent* – stores the results of the diagnostic reported within a well defined timeframe.

- *the learning agents* – communicate with other learning agents, located in other monitoring and diagnosis cell, for improve the knowledge about the process,
- *the GUI/WEB agent* – provide a friendly man-machine interface. Information about the actual state of the process is available on Internet or provided by mobile telephony.

For implementing the software agents system, the .net framework was choose, because of its advantages: native XML Web Service Support, improved reliability, increased performance, powerful granular security, integration with existing systems, mobility support, support for more than 20 programming languages, flexible data access, etc.

4. INITIAL RESULTS

The architecture is at the moment in development. Some initial results, in laboratory environment were promising. A single cell was implemented, for a temperature monitoring and diagnosis.

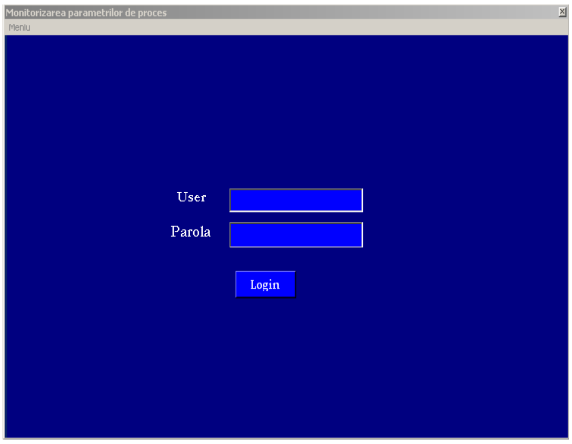

Figure 4. Example of GUI. Login panel

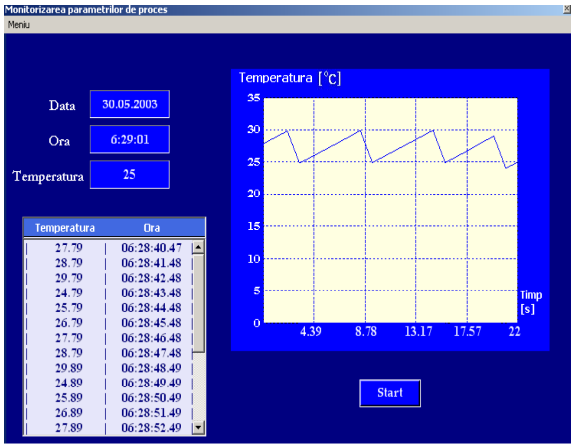

Figure 5. Example of GUI. Monitoring panel

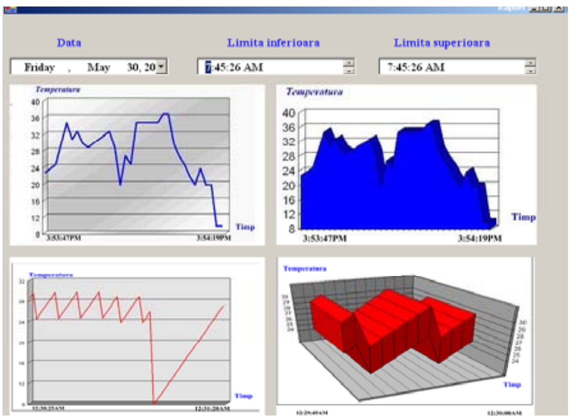

Figure 6. Example of GUI. Charts and reports panel

The application is implemented in C++ at the micro-controller level and in C# up from local concentrators. GUI/WEB agent allows users to interact with the application on the local network or on the web, using ASPX pages. Knowledge database is implemented using a SQL Sever 2000. An example of GUI panels is presented in Figures 4, 5 and 6.

5. CONCLUSIONS

In the paper, a monitoring and diagnosis system based on distributed agents is presented. The system is designed to act in a general manner, and can be particularized for a specific plant. The hardware architecture uses common components, and the software architecture is based on .net technology. In addition to common monitoring and diagnosis tasks, the system allows the web information access and the special situation alarm using mobile devices.

Due to the complexity of the problem, the paper deals only with the hardware and software architecture of the system. The problem of core and wrapper implementation will be presented in further works.

The system is under development, but a set of laboratory tests are already done, for a temperature monitoring and diagnosis.

6. REFERENCES

1. Albert, M., Längle, T., Wőrn, W. (2002), *Development Tool for Distributed Monitoring and Diagnosis System*, DIAMOND project, EUSPRIT EP 28735.
2. Deb, S., Ghoshal, S., Malepati, V.N., Kleinman, D.L. (2000), Tele-diagnosis: Remote monitoring of large-scale systems, *IEEE International Conference on Systems, Man and Cybernetics*.
3. Deb, S., Mathur, A., Willet, P.K., Pattipati, K.R., (1998), De-Centralized Real-time Monitoring and Diagnosis, IEEE
4. http://www.fipa.org, (2001) FIPA 98 Specification.
5. Froelich, P., Nejdl, W. (1996), Resolving Conflicts in Distributed Diagnosis, *12th European Conference on Artificial Intelligence ECAI'96.*
6. Vălean, H., Leția, T., Aştilean, A., Hulea, M., Avram, C. (2002), A Comparison of Distributed Object-Programming Techniques, *2002 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics AQTR 2002 (Theta 13)*, Tome II, pp. 480-485.
7. Vălean, H., Leția, T., Hulea, M., Aştilean, A., Avram, C. (2002), Distributed Control Architecture Using DCOM Technologies, *International Scientific Conference UNITECH '02*, pp. 154-156.
8. http://www.xml.org, Extensible Markup Language, (2001).