International Federation of Automatic Control – IFAC

Institute for Handling Devices and Robotics - IHRT
Vienna University of Technology

1st IFAC Workshop on

Multi-Agent-Systems in Production

MAS'99

December 2nd – 4th, 1999
Vienna, Austria

Preliminary Program

Workshop organized by:

Institute for Handling Devices and Robotics
Vienna University of Technology

on behalf of OePWZ, Austria

APPLICATION OF SIMULATION IN MULTI-AGENT SYSTEMS

Attilio Brighenti [1]
Adrian Dore, Christofer JD Senior [2]
Thomas Laengle, Martin Albert [3]

[1] *S.A.T.E. - Systems and Advanced Technologies Engineering s.r.l., San Marco 3911 30124 Venezia (IT)*
[2] *Advanced Diagnostic Solutions (ADS), GenRad Ltd. Orion Business Park, Birdhall Lane - Stockport, Cheshire, SK 0XG (UK)*
[3] *Institute for Process Control & Robotics, Karlsruhe University, Kaiserstraße 12, D-76128, Karlsruhe (DE)*

Abstract: The **DIAMOND** project is developing a **DI**stributed Architecture for **MON**itoring and Diagnosis, by which a flexible integration of a variety of M&D systems will be possible. The optimisation towards real-time operation of this multi-agent system and of its network depends on a large number of parameters and functional options. The simulation of the agents' behaviour and of their communication network plays an important part in the project, aimed at fulfilling said objective. Virtual test bench simulators of two industrial applications will also be developed to provide the confidence on the system before its final demonstration by real test beds.

Keywords: Diagnosis, Monitoring, Multi-agents, Networks, Simulation, Modelling

1. INTRODUCTION

This paper overviews the role that mathematical modelling and simulations play in the development of an innovative architecture for Diagnostic Systems, which is undertaken within the **DIAMOND** project (**DI**stributed Architecture for **MON**itoring and Diagnosis), partly funded by the European Commission under the Contract No. EP 28735.

The main goal of the project is to provide an *infrastructure* for the *communication* and *knowledge integration of* Monitoring and Diagnosis (M&D) agents and a *development environment* for distributed multi-agent systems (Sanz Bobi *et al.*, 1999, Senior *et al.*, 1999).

The communication platform is tested and benchmarked by the help of a virtual model of a multi-agent network for a typical plant.

In order to prove and validate not only DIAMOND's functionality but also its flexibility, two M&D systems are developed in two completely different industrial application areas:

- a coal power plant
- an automated welding cell.

After testing these M&D systems with virtual models, they will be deployed to real systems, Union Fenosa's Coal Power Plant in Anllares (Spain) and an automated welding cell made by KUKA Roboter GmbH (Augsburg, Germany).

At the end of the project two demonstrators will be prepared for each test bed to show the functionality and the advantages of the M&D systems implemented by the DIAMOND architecture.

2. DIAMOND's AGENTS

The focus of the DIAMOND architecture lies on allowing the integration and co-operation of several single M&D systems into one global system to jointly handle the various tasks and problems. On the other hand the architecture allows control of all systems from a single platform with a master Graphical User Interface (GUI).

In order to enable the diagnostic agents to co-operate, the methods of semantically and spatially distributed diagnosis are used (Sanz Bobi *et al.*, 1999).

The following briefly describe the agent types that are being developed in DIAMOND, their role and their interactions.

2.1 Monitoring Agents (MA)

Each plant or machine component has one or more MAs that have to acquire and record data on specific component variables or parameters, exchange data with the component itself or other agents, when requested, under real-time conditions.

One MA may for example acquire temperature data on an engine cooling system, another on noise or vibration generated by gears.

MAs may also perform light pre-processing tasks if convenient. Other tasks include starting the diagnosis by requesting it from the Facilitator Agents (FA) or the Diagnostic Agents (DA) in the local domain, handling the FA registrations, interfacing to the global GUI, performing self tests (periodic, at start-up, on user request) and sending signals to process (e.g. setting parameters).

2.2 Facilitator Agents (FA).

Each component has also its own FA that performs networking and facilitation jobs and acts as mediator between the groups of the MAs and of the DAs.

More specifically the tasks foreseen for the FAs may include starting diagnostic process, keeping knowledge about functional and structural relationship of agents and components in the local domain and their references, registering MAs in local domain, handling DA and FA registrations from other domains, interfacing to the global GUI.

2.3 Diagnostic Agents (DA).

Each component has one DA or its own group of ***semantically distributed*** DAs that perform the actual diagnosis. Indeed critical components of a plant or machine may be diagnosed following independent, complementary approaches. For example a car drive train may be diagnosed by two different agents based on cooling oil data or on its vibration and noise, both having the scope of early identifying the onset of abnormal wear and inefficiency.
The two DAs exchange the data relevant to their diagnosis with the component via some MAs and FAs.

The specific tasks of a generic DA include registering at local FA (and local MA if needed), establishing direct communication with MAs by obtaining the MA references from the FA and storing the obtained references, generating the diagnosis, sending its results to the FAs and to the Diagnostic Blackboard Agent (BBA) whenever the result changes significantly, storing the inference path, namely the background logical path to the diagnosis, publishing it on request and interfacing to the global GUI.

2.4 Diagnosis Blackboard Agent (BBA).

All diagnoses from the several diagnostic agents are output to a Diagnosis Blackboard. The BBA is an agent itself too, in so far as it's in charge of performing interactive actions. It is three-dimensional because the system uses semantically (vs. various DAs) and spatially (vs. various components) distributed diagnoses and is time dependent.

The tasks of the BBA include recording of diagnostic result (only upon change), sending notification to the Conflict Resolution Agent (CRA), whenever diagnostic results have significantly changed or new results have arrived, interfacing to the global GUI.

2.5 Conflict Resolution Agent (CRA).

This agent, together with the BBA, is the core of the DIAMOND architecture. It checks the BBA for conflicting diagnoses and resolves those if some are found.

It is not domain specific and starts upon notification of the BBA, only when no other conflict resolution process is running. It checks the inference path of each DA involved, locks them during conflict resolution, outputs a set of potential diagnoses and rates them with the possibilities implied.

An acceptance by the user is required in order to leave it ready for new requests from the BBA.

3. SIMULATIONS IN DEVELOPING THE DIAMOND MAS

The challenging objective to prove the DIAMOND architecture and the variety of functions of real M&D agents onto real plants, requires a careful testing throughout the development of the system and its components.

Mathematical modelling and simulation play an important role to this regard, because they can anticipate situations that, although expectable in principle, are not easily reproducible in real plants, or even they are not wanted at all (e.g. major failures).

They also help understanding the dynamic interactions among the various agents, the network and the client systems, the impacts of queues and priorities of protocols, the resources allocation criteria, etc., in order to prevent excessive time losses, odd or faulty diagnoses.

Finally they allow a thorough testing and demonstration of the multi-agent M&D system behaviour and interface with the hardware, before its integration with a real plant.

For this reason the DIAMOND project includes:

- the development of the ***agents network simulator***, that will be used both as an agents design and architecture test bench and, once DIAMOND is commercial, also as a useful off-line test tool.
- two virtual test bed simulators (one of the Coal Power Plant water cycle chemistry and one of an Automated Welding Cell robot), interacting with the respective demonstration M&D system like the real plants, though for limited subsets of components.

Furthermore, ***simulators subsets*** will also be implemented, to act online or offline, as part of some DA expert systems based on model based reasoning.

All the models will be developed by MATLAB-SIMULINK (® by The Mathworks Inc.) in order to make the integration of discrete events and the dynamic models more easy and flexible, leaving the possibility to generate C++ coded models for the final implementation.

4. AGENTS NETWORK MODELLING

The scope of the network modelling will be primarily to test the performance of the multi-agent infrastructure to optimise the design of each agent.

Although real time constraints are not always a firm requirement, costs considerations (Senior *et al.*, 1999) push towards the availability of diagnoses as early as possible.

Therefore one critical success factor of the DIAMOND project is to prove by a general demonstration programme that it can be both reliable itself and fast enough.

The ***agents network model*** will include the following main elements:

- a model of a multipurpose and/or of a few specialised agents, featuring queues and priorities management,
- a model of the communication protocol (FIPA-ACL),
- a switchboard like model managing the requests of protocols between agents.

During the design and, later, during the implementation of a DIAMOND application, the user in charge of this task shall be able to verify in an interactive mode the performance of his specific multi-agent system, before making it interact with the physical system or, before this, with the virtual one.

4.1 Agent basic model

The preferred approach to the agents modelling according to the present progress of the work is to build up a basic model of a generic agent, in order to allow a flexible utilization of this model to the variety of agents above mentioned.

The scope of this phase of the activity is therefore to identify common functionalities that differ from each other agent only by quantitative performance figures or by activation settings.

These functionalities include:

- Conversation ports
- Handle queue
- Internal processing

An agent can initiate a ***conversation protocol*** with other agents, following a procedure stored in a basic sequences database or matrix.

The agents' model will include three conversation ports. This number may be increased later in the project in order to test the sensitivity of the system performance to this parameter, once the modelling approach has been confirmed.

Each port will be able to handle one conversation at a time, thus each agent shall communicate with no more than three other agents in parallel.

Each port is occupied by a conversation, until its protocol has reached its last step. Afterwards, the conversation port is no longer occupied and can be used by a new conversation, following the queue handling criteria.

When an agent initiates a conversation protocol its port assigned to that conversation is occupied even if the called agent is unable to respond because it has no free ports available. This conversation is then continued as soon as the responding agent assigns a port to the incoming protocol, following its own priority based queue management.

The agent's ***handle queue*** stores all incoming protocols (in the real world the first message of the protocol), if there are no free communication ports at this agent. The size of this queue is unlimited.

The protocol in the handle queue with the highest priority will use the first available communication port to start a new conversation.

The simulation model therefore implements message priority handling.

Each agent needs a specific time to process incoming data. The ***processing time***, defined specifically for each agent, depends on the type of agent, protocol and process complexity.

A monitoring agent usually requires a shorter processing time than a diagnostic agent. This delay factor will be derived experimentally and parameterised, where possible.

Some DAs then do need data from a number of other agents to perform a diagnosis. This situation is simulated according to the protocols sequence set, as per the definition of each type of protocol, and by the queuing of cascade protocols.

The agent's protocol processing time depends on the conversations actually ongoing at the same time. The agent model includes a processing resource allocation block that manages the velocity at which each protocol is handled by the agent processor, based on each protocol priority and intrinsic characteristic defined in a suitable table. In this way, for example, it could be defined that protocols with higher priority are granted more processing resources than lower priority ones, so that, under similar protocol complexity conditions the former are completed in a shorter time.

4.2 The agents network model

The multi-agent network model will be represented by a GUI screen (figure 1) displaying the state of the agents interacting to each other and with input ports that represent the events generating an operational diagnostic cycle.

A map of coloured boxes each representing an agent will show whether they are idle, conversing with or waiting answer from another agent. In each box the ID of the responding agent and the type of protocol in use will also be displayed (in figure 1 the ID is exemplified by the row/column code). The three cases exemplified in the zoomed in box, i.e. for the conversations involving agent E3, are:

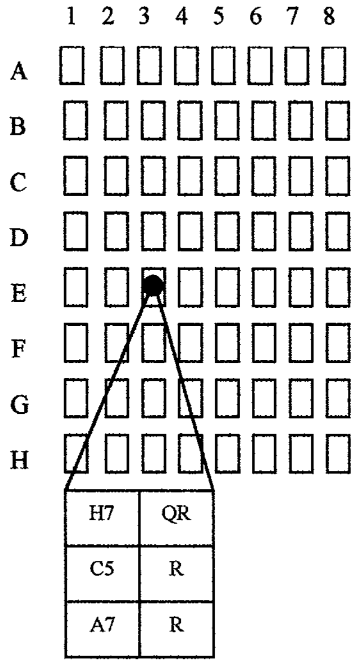

Fig. 1 - Representation of the multi-agent network model display

Port 1. Ongoing conversation with agent H7, Query-ref protocol (No. is omitted).

Port 2. Ongoing conversation with agent C5, Request protocol

Port 3. Ongoing conversation with agent A7, Request protocol

The input events, e.g. low level failures and/or symptoms starting the diagnostic process or routine monitoring and diagnostic cycles, will be generated either manually or automatically through an I/O interface compatible also with the monitoring interfaces on the virtual models of the physical systems. This will allow the final virtual demonstration of the DIAMOND architecture integrated with the two planned test beds models.

4.3 The FIPA-ACL protocols modelled

For the purpose of this demonstration the following protocols are foreseen among those belonging to the FIPA specification (1997, Part.2):

- ***Subscribe***: In this protocol, the recipient is invited to inform the sender of the value of a reference and to notify again whenever the object identified by the reference changes or it must send a *failure* communicative act, if a failure has occurred. This procedure will recur until the recipient receives a *Cancel* act to terminate the *Subscribe* protocol exclusively by the same agent that sent the originating *Subscribe* act.

- ***Query-ref***: In this protocol, the receiving agent is requested to produce some kind of information. If initiated by *query-ref*, an *inform-ref* is planned as reply. If the recipient does agree with the *query-ref*, it must send directly an *inform-ref* or a *failure* communicative act.

- ***Request***: This protocol simply allows one agent to request another to perform some action. This is the most common protocol, usually implying the responding agent to initiate one or more other protocols with other agents, in a cascade pattern. The initiating *Request* protocol ends when the whole agreed action is finished or failed.

The ***Cancel*** protocol, although essential in the real communications implementation, is not within the scope of the simulations, which aim at highlighting maximum load situations, when a certain number of subscribe protocols are occupying some agents ports.

4.4 Agents characterization

In order to run the network model with realistic conditions, the characteristic parameters of the agents developed for the two demonstration applications will be derived by testing them under controlled conditions after they have been implemented. The processing time delay will be characterised in an "agents' test shop".

The network model will then treat these inherent agent delays as dynamic parameters, quantitatively function of the agent load, of the type of data processing and of the protocol.

The delay added by queuing effects due to cascade type communications with other agents, as foreseen by a *request* protocol, will be taken into account thanks to the priority driven queues managers in each agent model. The resources allocation options for protocols processing vs. their priority will be tested to see their influence on the overall performance.

5. MODELS OF PHYSICAL SYSTEMS

The models for the demonstration of the two DIAMOND applications by the virtual test beds, that precede the real ones, are being developed following two different approaches that are briefly described in the following.

5.1 Coal Power Plant (CPP)

The application envisaged addresses the M&D of the water chemistry in the cycle of a power plant, in charge of the prevention of abnormal impurities increase that would cause damage to the plant's components (Sanz-Bobi *et al.*, 1994). In the project the approach previously followed and successfully demonstrated on a small scale set of data based on the neural networks (NN) modelling techniques, (Sanz-Bobi and Munoz., 1998) will be extended to large scale, to include all the relevant parts of the cycle. In this extension work this technique will be also compared with other approaches, such as the neuro-fuzzy, in order to achieve fast yet accurate training.

The training data set will include actual time histories recorded from the Union Fenosa's Anllares power plant. These data will be purged from possible abnormal data sets, so that the trained NN model will best fit the normal behaviour.

The generation, either by a manual exogenous input to the model or by interfacing the real plant, of abnormal values of some variables monitored, will be detected by the reaction of the model, which will cause some related dependent variables to exit from expected trends and values (symptoms of anomaly).

In this phase the DIAMOND M&D system will receive these data records, according to its specified operation to see if it is able to identify and return the originating cause of the anomaly.

If it succeeds in identifying the same cause inputted to the model the integrated M&D system implemented with the DIAMOND technology will be demonstrated at the virtual test bed level and be ready for integration with the real plant, where not all the tested exogenous inputs can be planned or generated conveniently.

5.2 Automated Welding Cell (AWC)

The method in using the model in the AWC will be similar to the one above described for the CPP.

The model however will mostly reproduce a fault tree structure, from which exogenous inputs representing both normal events and abnormal primary causes of malfunctioning will cause symptoms outputted towards the M&D system (figure 2).

The exogenous events may be of the following nature, which the virtual AWC model will deal all with, in order to test the M&D system effectiveness:

- Analogue events or "slow evolution" events, e.g. backlash onset.

- Binary events or "fast evolution" events, e.g. breaking of an electrical cable.

- On-call events, i.e. purpose tests made by pre-programmed schedule, as described in the following section.

The model will have a hybrid nature, i.e. be based on discrete events and dynamic processes. The dynamics may be limited to parts of the plant or machine, to describe phenomena that could help the M&D tasks, featuring condition based strategies, which may be affected by some dynamic patterns.

For instance, an increase of a robot joint backlash might be suspected on-line by looking at special patterns of the current signals in the joints motor drivers, before a more reliable "on-call test" with a dedicated tool can confirm this. These patterns will be studied and made part of the whole AWC model.

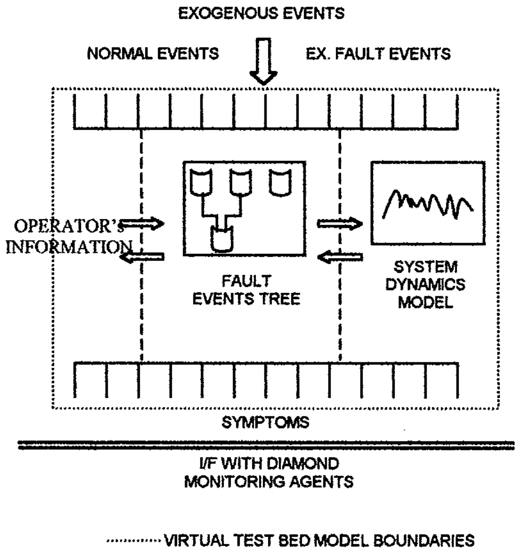

Fig. 2. Scheme of the AWC model

6. MODELS FOR DIAGNOSTIC AGENTS

The above-mentioned models, both the NN based and the deterministic dynamic models that are part of the virtual test bench, can be useful in the real M&D system, if integrated in a model based diagnostic agent. Therefore it is planned that, at some stage of the DIAMOND models development, these sub models are extracted and integrated in the DA to work either on-line or off-line. For the CPP the following tasks will be performed:

- Searching of patterns at sub-models levels to enhance "real time" processing
- Synthesize and extract from the whole NN model subsystems behaviour patterns to be used by the DAs in a more time-effective way
- Embedding sub-models as modules of the DIAMOND system DAs.

Similarly, for the AWC some preliminary results have been achieved regarding the backlash symptoms identification, by a simple model of each arm joint and inertia. The signal processing of the response to a motor oscillatory torque signal, e.g. of the chirp type, has been analysed by various methods, including the time dependent Transfer Function Analysis and the cross-correlation analysis of the output signals.

The best results so far achieved regard the time delay that maximises the cross-correlation between the acceleration signals at the arm shaft and at the motor shaft; this time delay proved well correlated with the amplitude of the backlash (figure 3).

This type of analysis is being further studied, to improve and extend the correlation to other signals and bring it to a practical applicability into a suitable DA for the AWC.

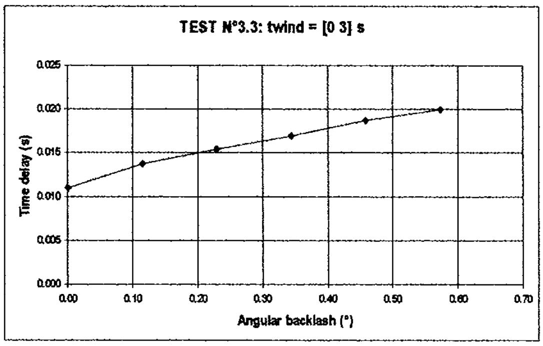

Fig. 3. Sensitivity of the cross-correlation time delay for two response signals to the backlash in a robot arm joint.

7. CONCLUSIONS

The work performed so far in the DIAMOND project regarding the system definition and the simulation activities brought to the main agents types and functionalities that mostly affect their behaviour in a network.

In the next period of this research an extensive cooperative work among the partners is necessary to develop and validate the models outlined both as regards the infrastructure and for the very diverse fields of application identified as test bed.

The simulators developed in this project will be useful not only to identifying possible bottlenecks or faults at the design stage of this innovative agents architecture, very promising for effective and flexible M&D systems, but also during the application phase, as help for the user in his/her implementation activity. Simulators by-products are also expected to become important parts of the diagnostic processes and commercial strategies of the partners involved.

ACKNOWLEDGMENT

This research work has been performed at S.A.T.E. s.r.l. (IT) in collaboration with Union Electrica Fenosa (ES), KUKA Roboter GmbH, Genrad Ltd (UK), Instituto de Investigación Tecnológica - Universidad Pontificia Comillas (ES) and the Institute for Control and Robotics, Prof. Dr.-Ing. H. Woern and Prof. Dr.-Ing. R. Dillmann, Department of Computer Science, University of Karlsruhe. The research is partly funded by the European Commission in the DIAMOND project under the Contract No. EP 28735.

ACRONYMS

AWC	Automated Welding Cell
CPP	Coal Power Plant
DA	Diagnostic Agent
ES	Expert System
FA	Facilitator Agent
MA	Monitoring Agent
MAS	Multi Agent System
M&D	Monitoring and Diagnosis
NN	Neural Network
TF	Transfer Function

REFERENCES

FIPA 97 Specification, Part 2, http://www.fipa.org

Sanz-Bobi, M.A., et al. (1994). Control and diagnosis of water chemistry in the water-steam and water make-up in a fossil fuelled power plant. *International Journal Electrical Power Energy Systems*, **Vol. 16, No. 4**.

Sanz-Bobi, Munoz A., (1998). An incipient fault detection system based on the probabilistic radial basis function network: application to the diagnosis of the condenser of a local power plant. Neurocomputing, Vol. 23, 177-194.

Sanz-Bobi, M.A., Sanchez-Ubeda, E.F., Villar J. Revuelta S. Kazi A. (1999) Multi-agent diagnosis systems in industry. 1st International IFAC Workshop on Multi-Agent Systems in Production (MAS'99), Vienna - Austria.

Senior JD, C., Dore, A., Laengle, T. and Albert, M. (1999), An architecture for distributed diagnosis systems. 1st International IFAC Workshop on Multi-Agent Systems in Production (MAS'99), Vienna - Austria.