A Multiagent-Based Architecture to Support Power System Recovery Decision Making

J. P. Barreto Neto, M. F. Q. Vieira and B. Espinasse

***Abstract* -- This paper presents the modelling of a multi-agent system at the basis of a computer tool for industrial operator's decision making support. The agents that compose the system implement a decentralized strategy to deal with contingency situations in industrial systems, based on the repair solutions approach. This paper discusses the agents' architecture for the context of electrical power systems' supervision and how to apply the proposed strategy to real situations giving directions for future work.**

***Index Terms* -- Multiagent Systems, Industrial Systems, Decision Support Systems, System Restoration**
***Index Terms* -- Multiagent Systems, Industrial Systems, Decision Support Systems, System Restoration**

I. INTRODUCTION

THE operation of industrial systems demands continuous supervision and control. This is usually performed by operators supported by SCADA (*Supervisory Control And Data Acquisition*) systems. Handling the industrial process data may represent a cognitive overload causing operator discomfort and error. The typically centralized approach to decision making on its turn can increase the time necessary to implement repair solutions after an operational fault. Among the reasons for the delay in finding the optimum solution is the time spent in the communication between the people in control centres and the affected areas of the system.

In the power system context all efforts are centred in preventing power supply interruptions since the user satisfaction is highly influenced by the frequency and duration of the power cuts [8]. Therefore in this context, solving the restoration problem is a critical activity. It involves the technical problem to be solved bound by the time restriction imposed both by technical and financial implications. From the technical viewpoint, there is a risk to equipment integrity when subjected operate overloaded for long periods of time. From the economic viewpoint, electricity companies pay fines proportional to the duration of the service interruption.

This paper proposes a multiagent based system to support operator decision making during system recovery after an operational malfunction. The multiagent system is based upon a decentralized control approach which aims to reduce recovery time after contingencies. The proposed strategy is distributed as opposed to the centralized one often adopted. Due to its distributed nature the decision process must benefit from the availability of alternative control elements and the possibility of localized solutions which could reduce the time spent in the decision making process. Since the time is a critical parameter it is essential to provide the operator with efficient tools. Therefore this paper focuses on the presentation of a multiagent architecture to implement the recovery strategy proposed in [23].

The decentralized strategy was conceived as a potential solution to minimize the time spent in communication during problem solving. It considers the existence of several decision centres distributed in the system instead of a centralizing one. Since the time is an important parameter in the restoration process, it is essential that tools and processes exist to assist operators during decision making to ensure efficacy and efficiency.

The agents' architecture is the first step to implement the decision making tool that will propose potential repair solutions based on the system evaluation and the negotiation between software agents.

This paper is structured as follows. In section 2 it is presented an overview of the literature related to decision making aid systems. In section 3 it is presented the proposed decentralized strategy and the corresponding software agents and is followed in section 4 by the software agent architecture. In section 5 it is presented the application of the agents' architecture to the operation context of electricity transmission system. In section 6 the strategy is evaluated in the context of the case study and section 7 presents the final considerations and future directions for the work.

II. RELATED WORK

The well known decision-making aid systems are largely used in the industry. The benefits include a more efficient operation and minimize the impact of system faults. Another important effect is to preserve the corporative knowledge relative to system operation normally detained by the specialist [17]. These systems are built on the basis of artificial intelligence concepts and techniques. Among the techniques employed in building decision making aid systems it is found Fuzzy

This research has been supported by the CNPq Brazilian agency, with a scholarship given to the first author during this project which resulted in his master thesis.

J P. Barreto Neto and M. F. Q. Vieira are with the DEE Department of the Federal University of Campina Grande, Brazil (josepbn@gmail.com, fatima@dee.ufcg.edu.br)

B. Espinasse is with the LSIS CNRS lab. of the University of Aix-Marseilles, France (e-mail: bernard.espinasse@univ-cezanne.fr).

logic, genetic algorithms. Examples of those systems are found in areas such as medical diagnosis [15] and oil extraction and processing [9].

Besides the application context, these tools also differ in respect to their focus. Whereas some are developed to support the planning stages process operation, others focus on the operation itself. In [17], for a given scenario, fuzzy logic is used to determine the best project alternative for an oil production system. In [16] a software agent strategy is used to optimize containers use in port terminals. In [4], genetic algorithms are used to support an electric distribution system, but the focus of this solution is in the protection of the system against one type of fault: earth-phase faults.

The operation of critical systems such as electric systems and oil production systems, also benefits from the adoption of decision-making aids. In [18] it is found a hybrid system to handle contingencies in oil plants that uses software agents to perform inferences bases on fuzzy logic. In [3] it is proposed the use of various systems to detect oil leakage in ducts and thus minimize its financial and environmental costs.

In the specific context of this work, which is to recompose an electric system after a contingency, there is the work presented in [13] that proposes fuzzy logic and heuristic search to help the operator during system restoration after a blackout. During the search for a solution, the objective is a real-time operational solution based only on "permanent regime" restrictions

Another application related to system recovery is found in [5] which proposes a specialist system to identify and diagnose faults filtering alarms and presenting the operator only relevant information. In [1] it is proposed a multiagent system to optimize the use of vehicles when treating contingencies in the electricity distribution network reducing the time spent in the recovery process.

Most of the work mentioned above shares the same general purpose and application context as the one presented in this paper, which is system recovery after contingency. In contrast, the recovery strategy presented here is cooperative, since it is based on agent interaction to propose solutions to minimize the impact of the contingency over the overall system.

III. RECOVERY STRATEGY

From the literature review, the major problems related to power distribution network supervision are: *network protection* in normal operation and in the event of disturbances; and *load sharing*. The methods to solve problems during incidents are based on diagnosis and repair [21][20][7][19].

As already introduced in [23], we proposed a recovery strategy based upon the negotiation between software agents which are directly related to the elements or groups of elements of the real system. These agents reason according to rules and the state of the system variables.

This strategy is based on the repair solution philosophy [22][24] that aims to minimize the propagation of the problem through the most local possible resolution. In other words, it aims to eliminate the problem through the cooperation between the elements present in the neighbourhood, without the need of propagation to all the elements of the system. The implementation of this strategy demands the specification of the agent architecture and of their behaviour.

It consists of four phases that are performed in a sequence until a solution is found or until all the phases have been performed:

- *Phase 1*: Attempt of internal resolution, it consists in an attempt to solve the problem within the system element which has been disturbed.
- *Phase 2,* Attempt of resolution in the same level of where occurred the disturbance. A negotiation exists between the elements located in the same hierarchical level in order to find a solution.
- *Phase 3,* propagation to upper levels. The problem is propagated to an upper hierarchic level. This phase is repeated until the highest level of the hierarchy is reached. It is performed in two steps. First the problem is propagated to the hierarchical level immediately above. In this step a request for cooperation is sent to an element directly connected to the disturbed one. In step 2 the receiver requests the cooperation of the other elements in the same hierarchical level. This phase can be repeated as many times as there are hierarchical levels not yet reached in the search for the solution.
- *Phase 4,* occurs when the highest hierarchical level is reached without a solution. In this context of system recovery it consists on a request for load shedding based on a scale of priorities. That is made to guarantee that the most critical consumers continue being supplied and the system does not collapse.

IV. THE AGENT'S ARCHITECTURE

As already mentioned, the decentralized strategy is based upon the software agents' interaction. The proposed agent's architecture is composed of two layers: one physical and another decisional, plus an interface connecting them, as shown in Figure 1. The connecting interface manages the exchange of messages between agents and can be implemented by the use of *sockets* or API's.

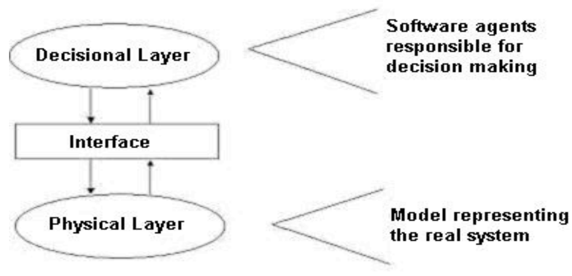

Fig. 1 – Proposed agent architecture.

The agents have only partial knowledge of the environment and have their behaviour determined by the rules defined in the inference engine JESS (Java Expert System Shell) [6] that is a Java implementation of the framework for expert systems CLIPS, developed by NASA. Each agent has as behaviour plan, either a local plan or role protocols that guide its behaviour. The behavioural rules derive from norms that prescribe how the system operation must be restored after a contingency. It follows the plan description.

The Local Plan describes an isolated behaviour which does not demand interactions to be fulfilled. The Role Protocol represents the behaviour of an agent in interaction with another one. These plans are stored in one of the four modules that compose the agent, as it is shown in Figure 2.

Each one of the four modules presented in the Figure has a specific function, as follows:

- *Communication Module*: its function is to manage the message exchange between the agents. It verifies the correctness of the message from the points of view: syntactic (in respect to the ACL language grammatical rules) and semantic (if the message context is known, through the name of the protocol being used);
- *Knowledge Module*: stores the information that the agent has about himself (individual knowledge) and about the other agents (social knowledge). Among other information, it also stores the physical and cognitive competences of the agent;
- *Expertise Module*: while the competences, present in the knowledge module describe what an agent is able to do, this module contains a detailed description on how the agent should do it. This detailed description is present in local plans and role protocols;
- *Decision Module*: controls the execution of the actions described in the expertise module by means of a behaviour plan manager.

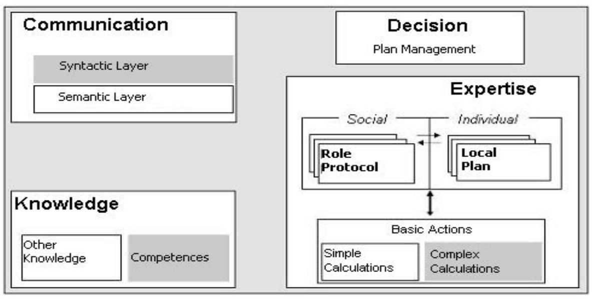

Fig. 2 – Software agent architecture.

Building the agents starts by building a graphical description of their behaviour based on the formalism ABD (*Agents Behavior Diagrams / RCA Représentation de Comportements d'Agents*) [22]. The ABD representation is composed by states and transitions instantiated to the context of multiagent. ABD, as the other graphical formalisms used to describe the agents' interaction protocol found in the literature, does not have a formal semantic [12], it offers only a semi-formal specification of the agent interaction [14].

The ABD graphic representation is in the XML format and is used as the entry to a translating tool into a JESS file with the agents' rule structure. The rules can then be edited to represent the agents' intended behaviour and the evaluation functions used by them before performing their tasks. The agents are then built using the JADE (Java Agent Development Framework) platform.

The set with all agents compose a decision layer (shown in Fig. 1) to be later used as the top layer of the proposed decision-making tool. The JADE platform simplifies the development of multiagent systems supplying a complete set of services in compliance with the FIPA specifications [2].

V. Applying the Strategy to an Electric Power Sytem

The application domain chosen to validate the agents' architecture and related recovery strategy was in the electric supply restoration, more specifically in the transmission system. The choice was based on the hierarchical characteristics of the transmission system which allow dealing with individual subsystems. Currently in Brazil, in most electric companies the system has a central decision structure based on regional control centres which in turn are subordinate to a national centre in charge of the national grid. To operate the grid, it is necessary to ensure communication between all levels.

From the operational point of view, the electric systems are subdivided into generation, transmission and distribution systems. In the event of a contingency there is the need for communication between hierarchical levels which can be very high and can contribute to longer periods of interruption with the related costs.

To exemplify the use of the decision-making aid tool in the described context it follows the description of a simplified model of the sub transmission part of a power system, as shown in Figure 3.

A. Case study model

The following model represents a simplified vision of the real system and only takes into consideration information relative to: system loads, node tensions, and the transmission line parameters. Those parameters are the minimum necessary for analyzing the system dynamics through the Newton-Raphson flow calculation method. Although the transmission line length is not considered in the original method, this is going to be used by the agents to evaluate a cost function when choosing the best solution, in case there are many.

The oversimplification imposed to the real system limits the scope of the results during validation, to a qualitative analysis of the possible solutions. In order to obtain a quantitative analysis it would be necessary to extend the system model to consider other variable. As a result the search for solutions would demand a higher processing power. Besides, the agents' behaviour must be considerably more complex in other to deal with a more complex model of the system. In turn, the raise in agents' behaviour complexity demands more processing power and time to evaluate the rules which govern their behaviour. It remains to be investigated if this extra processing power will be compensated by raise in quality of the solution. Among the criteria to be adopted when evaluating the agents' proposed solution there is the time spent to reach it. If the real-time deadline is reached or surpassed, the strategy will be considered useless from the practical point of view. It must be pointed out that the software agents developed for this study have all the information relative to the operational limits and about the entire system, so as to guarantee that system operation is according to the programmed standards.

The agent types that model this system are the following (Figure 3):

- SE – Step-up substations (level 1);
- SS2 – Sub transmission substations 2, directly connected to the step-up substations (level 2);
- SS3 – Sub transmission substations 3 indirectly connected to the step-up substations (level 3);
- SD – Distribution substations, establish the system demand (level 4);
- LT – Transmission lines (links between substations).

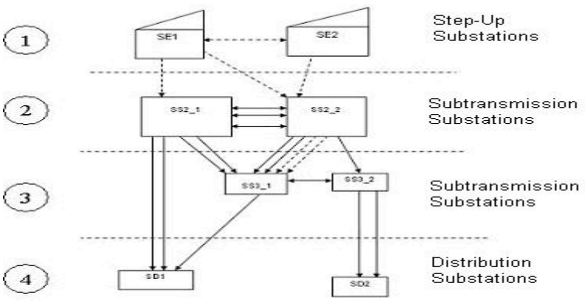

Fig. 3 – Simplified representation of an electric sub transmission system.

B. Contingency scenario

The goal of the hierarchical division of this system is to establish clearly the boundaries of each stage of the recovery strategy, based on the repair solution philosophy. For the scenario of a transmission line loss, more specifically the one connecting levels 2 and 3 as shown in Figure 3, one must establish the stages that compose the agents' solution process. The stages description follows:

- *Phase 1* corresponds to a search for a solution internal to the disturbed element. In this specific case, it evaluates if there is another transmission line connecting the two elements and if so, if it would be able to transmit the interrupted flow, respecting the pre-established operational limits. If this is not feasible, the second phase is initiated.
- *Phase 2*, a negotiation is established between the elements located in the same hierarchical level, as the disturbed one, in search for a solution. The interactions which represent the process of obtaining the solution in phase 2 are illustrated in Figure 4.
- *Phase 3* consists of propagation to upper hierarchical levels. In the specific case it means that a power surplus is requested to elements in higher levels in order to keep supplying all the loads attached to the disturbed element. The negotiation starts by element SS3_1 sending a request to all the elements directly connected to it, in the higher level. If a solution is not achieved, SS2_2 sends a request for cooperation to SS2_1. This phase is repeated until the highest level of the hierarchy is reached. In case the problem is not solved on this phase a process of load shedding is initiated in phase 4.
- *Phase 4* consists of a Load Reduction Process- That is made to guarantee that critical consumers continue being supplied and the system does not collapse. Initially SS3_1 sends a request to reduce the demand from its direct consumer SD1. In case this reduction is not possible begins the process of load shedding.

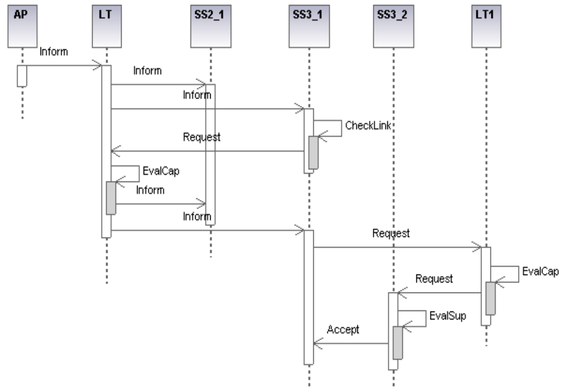

Fig. 4 – MSC illustrating the agents' interaction during the recovery strategy.

To validate the agents' solution, which is based on the proposed strategy it is necessary to follow the system parameters' behaviour along the various phases. In this particular case it was chosen the scenario where a transmission line is lost. Conceptually the strategy and modelled agents can be applied to other power system situations such as the loss of transforms, circuit breakers and other elements. Similarly the solution can be applied to other industrial contexts where operators must make decisions based on the state of the system's variables. The software agents must interact and negotiate in the search for solutions which can help operator's decision making.

In the simulation level, the interactions between agents depend on the complexity level of the system model, since this determines the volume of data that must be accounted for during the agents' reasoning process. The decision on which data will be considered during reasoning is made in the stage before defining the agents' behaviour rules, and is the result of the application context and the intended level of realism. On the other hand, the interactions determine the complexity of the interface layer between the system model and the supervisory software (e.g. SCADA).

To minimize the addressing load in the data packages exchanged between agents, it is proposed to create an interface agent in charge of routing the messages to the correct addressees. One could argue that such solution would leave a weak link in the communication system since a failure in such agent would interrupt communications. To avoid such problems it is proposed to introduce redundancy in this agent's structure. The authors acknowledge the potential weakness of this proposal, since the imposition that each message must contain the addressee's information might raise the time taken to find an acceptable solution. To come to a more definitely conclusion it is necessary to experiment with the interface layer analyzing from a cost/benefit point of view, and based upon a set of quality criteria such as the time, mentioned above.

VI. RESULTS DISCUSSION

Having modelled the system and specified the agents' architecture allowed anticipating questions and boundaries for this work which are now discussed.

The first difficulty consisted in modelling the electric system with a reasonable degree of realism. The choice of a software tool to build the model posed some difficulties which are discussed next.

The Anylogic™ tool offers support for modelling various types of system, but does not make available the mathematical support for performing partial derivates which is essential for the calculations of the simplest flow. Along with these calculations it is also necessary to convert each message exchanged between agents since those messages are treated as strings in the physical layer and as facts, by the JESS in the decision layer.

Given the need for the mathematical support, the software MATLAB® was chosen to implement the physical layer of the strategy. For the case of a simplified model dynamics which can be evaluated through the calculation of a single flow, the MATPOWER package can be employed, since it offers a set of files ***.m*** to solve the flow problem and the optimum flow problem [11].

Having solved the need for mathematical support, there are still two problems to be solved. One consists in the integration between the system model, built using MATLAB® and the agent layer. The second problem that remains to be solved is to reduce the time taken to update system information in the physical layer.

The first problem can be solved by using *sockets* and JmatLink, which is an API which allows using the MATLAB® resources in applications Java, such as *applets* and *servlets* from end users [10].

As for the second problem it is dependent on system requirements and can lead into situations where the proposed strategy will not be suitable for real time applications. In spite of the limitations mentioned above, the proposed strategy would still be applicable to operation management applications where the time constraints are not strict.

VII. FINAL CONSIDERATIONS

Due to the economical and technical impact of the elapsed time for system restoration, the use of decision aid tools combined with a decentralized repair approach presents itself as an interesting choice for the industry. The multiagent architecture presented on this paper implements the decentralized strategy proposed to treat system contingencies.

However, the proposal presented on this paper was only validated in simulation terms and based on a simplified system model as the physical layer. The full evaluation of the decisional layer is still to be performed and in particular the time taken by the interaction between agents. As a criterion to evaluate this approach it is suggested to compare its results with the ones issued by others found in the literature. Among the criteria to evaluate the efficacy of the strategy are the response time and the processing load associated to it.

Another aspect that must also be explored is the creation of a *Log agent* to register all the proposed solutions and the actions taken during the restoration process. This record would be useful for auditory purposes or for training purposes.

As future work on this research it was considered the following actions: (1) to refine the system model to evaluate the agents interaction efficiency; (2) to include the *log agent* mentioned above; (3) to build the proposed decision-making aid tool based on the agent architecture proposed on this paper; and then (4) to compare the results of the use of this tool with the results of the current centralized approach, both practical and found in the literature. Finally it is suggested to employ the strategy and related agents architecture to contexts other than electric systems.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank the Brazilian agency: CNPq for the scholarship given to the first author during this project which resulted in his master thesis.

IX. REFERENCES

[1] A. R. Aoki, G. Lambert-Torres, C. H. V. Moraes, B. R. Costa and J. A. Barbosa, "Sistema multi-agente para o atendimento de ocorrências na rede de distribuição de energia elétrica". *In Proc. 2003 Simpósio Brasileiro de Automação Inteligente*, pp. 1049-1054.

[2] F. Bellifemine, A. Poggi and G. Rimassa, "JADE – A FIPA-compliant agent framework". *Proc. PAAM'99*, pp. 97-108, London, UK, 1999.

[3] C. H. M. Bomfim, B. R. Menezes and W. M. Caminhas, "Detecção de vazamentos em dutos de transferência de derivados de petróleo: abordagem utilizando comitê de especialistas". *Proc. 2003 Simpósio Brasileiro de Automação Inteligente,* pp. 1055-1059.

[4] E. M. Campitelli, L. G. W. Da Silva, R. A. F. Pereira and J. R. S. Mantovani, "Sistema inteligente para alocação e coordenação de proteção contra sobrecorrentes em redes radiais de distribui-

ção de energia elétrica". *Proc. 2006 Congresso Brasileiro de Automática,* pp- 821-826.

[5] A. N. Duarte, J. P. Sauvé, W. Cirne, J. C. A. Figueiredo, M. C. Sampaio, E. R. Neto, M. A. H. Da Silva, A. S. Araújo, and M. S. C. Melo, "Tratamento de eventos em rededs elétrticas: uma ferramenta". *Proc. 2004 Congresso Brasileiro de Automática.*

[6] E. Friedman-Hill, "Jess in Action – Rule-Based Systems in Java", Manning, Greenwich, 2003.

[7] G. Friedrich and W. Nedjl, "Choosing observations and actions in model-based diagnosis-repair systems", *Proc. DX-92*, pp. 76-85. 1992.

[8] Y. Fukuyama, H. Endo and Y. Nakanishi, "A hybrid system for service restoration using expert system and genetic algorithm". *Proc. of 1996 IEEE ISAP*, pp. 394-398.

[9] F. A. Guerra, F. T. R. Tovar, M. P. Ramos, L. A. D. Corrêa, L. S. Coelho, C. Andrade and W. Baptista, "Sistema especialista para monitoramento e controle de processos: a experiência do projeto Injex". *Proc. 2003 Simpósio Brasileiro de Automação Inteligente*, pp. 450-455.

[10] JMATLINK "*JmatLink*". Acceded at 28/02/07, http://www.held-mueller.de/JMatLink/

[11] MATPOWER "*MATPOWER*". Acceded at 14/03/07, http://www.pserc.cornell.edu/matpower/

[12] F. Mokhati, M. Badri and L. Badri, "A formal framework supporting the specification of the interactions between agents". *Informatica*, Vol. 30, Issue 1, pp. 97-110, 2006.

[13] A. A. Mota, L. T. M. Mota and A. L. M. França, "Metodologia de recomposição coordenada de sistemas elétricos com base em técnicas de lógica nebulosa e busca heurística". *Proc. of 2004 Congresso Brasileiro de Automática.*

[14] Paurobally, S.; Cinnigham, J.; Jennings, N. R. “Developing agent interaction protocols using graphical and logical methodologies”. Proceedings of 1st Workshop on Programming Multi-Agent Systems, pp. 149-168, 2003.

[15] P. Ramnarayan and J. Britto, "Paediatric clinical decision support systems". *Archives of Disease in Childhood*, 87, pp. 361-362, 2002.

[16] R.A. Rosa and H. J. A. Schneebeli, "Um sistema baseado em agentes para alocação de recursos aplicado a um terminal marítimo de contêineres". *Proc. of 2003 Simpósio Brasileiro de Automação Inteligente*, pp. 1031-1036.

[17] A. B. S. Serapião, I. R. Guilherme, C. K. Morooka, J. R. P. Mendes and K. P. M. Franco, "Um sistema inteligente para escolha dentre alternativas de sistemas marítimos de produção". *Proc. of 2003 Simpósio Brasileiro de Automação Inteligente*, pp. 272-277.

[18] L. B. Sheremetov, M. Contreras and C. Valencia, "Intelligent multi-agent support for contingency management system". *Expert Systems with Applications*, Vol. 26, pp. 57-71, 2004.

[19] Y. Sun Y. and D. Weld, "Beyond simple observation: Planing to diagnose", *Proc. of DX-92*, pp. 67-75, 1992.

[20] S. Thiébaux, M. O. Cordier, O. Jehl and J. P. Krivine, "Génération de plans de reprise pour les réseaux de distribution EDF", *Rapport de recherche INRIA n° 3006*, Oct. 1996.

[21] S. Thiébaux, M. O. Cordier, O. Jehl and J. P. Krivine, "Supply Restoration in Power Distribution System: A Case Study in Integrating Model-Based Diagnosis an Repair Planning", *Proc. of 12th Conference on Uncertainty in Artificial Intelligence*, pp. 225-232. 1996.

[22] E. Tranvouez and B. Espinasse, "Protocoles de coopération pour le réordonnancement d’atelier". *Proc. Journées Francophones d’Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, pp. 295-312, 1999.

[23] M. F. Q. V. Turnell, B. Espinasse and G. Aventini, "Supply restoration in electric distribution networks: a multi-agent approach". *Proc. of I3M-Conceptual Modeling and Simulation SCS Conference*, pp. 129-138, 2005.

[24] M. Zweben, E. Davis, B. Daun and M. J. Deale, "Scheduling and rescheduling with iterative repair". *Proc. of 1993 IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, pp. 1588-1596.

BIOGRAPHIES

José P. Barreto Neto is Master student at the DEE Department of the Federal University of Campina Grande, Brazil (josepbn@gmail.com).

Maria F. Q. Vieira is Professor in electrical engineering at the DEE Department of the Federal University of Campina Grande, Brazil. She is researcher at the LIHM laboratory of this University (fatima@dee.ufcg.edu.br).

Bernard Espinasse is Professor in computer sciences at the University of Aix-Marseilles, France. He is researcher and team-leader at the LSIS UMR CNRS 6168 research center (bernard.espinasse@univ-cezanne.fr).