An Agent-Based Approach to EMS in Open Environments

Gilberto P. de Azevedo
gilberto@cepel.br
CEPEL
Rio de Janeiro-RJ, Brazil

Bruno Feijó Mônica Costa
bruno@icad.puc-rio.br - monica@icad.puc-rio.br
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro-RJ, Brazil

Abstract - Considering the new power system scenario, this paper points to the necessity of a new generation of Energy Management Systems acting in open and uncertain environments. The changes in this new scenario are analyzed and the great challenges to EMS developers are discussed. Agent technology is pointed as a promising approach to support the construction of the new generation of EMS. Moreover, an agent-based framework to EMS is suggested and some results of the use of the proposed framework are presented.

Keywords: Supervision and Control, Control Centers, EMS, Agents, Artificial Intelligence.

I. INTRODUCTION

The importance of control centers in the operation of electric power systems has been rising steadily for decades. The evolution of control centers improves the safety of the whole system, thus allowing for operation closer to limits and the optimization of investment and operational costs.

Computer-based control centers appeared as soon as their cost-benefit relation became acceptable. Today, computerized control centers of very different sizes and importance coexist in power systems. The evolution of those centers is clearly influenced by two main factors: the power system scenario and the changes in hardware and software. The power system scenario, which remained stable for decades, is now facing revolutionary changes in most countries. The software and hardware industries, however, have long been evolving fast.

Under the traditional power system scenario, two generations of EMS have been developed. They were functionally similar, but based on quite distinct computational architectures. The overall architecture of the next generation of EMS will be strongly influenced by the transformations in the power systems industry and also by the continuous evolution of distributed processing.

In this work, Agent Technology is suggested as a promising approach to the architecture of the next generation of EMS. Agents are software entities that are specially suited to face the challenges of that generation, which include acting in uncertain and open environments.

Paper BPT99-351-21 accepted for presentation at the IEEE Power Tech'99 Conference, Budapest, Hungary, Aug 29 - Sept 2, 1999

II. THE EVOLUTION OF ENERGY MANAGEMENT SYSTEMS

Energy Management System architectures are deeply influenced not only by the power system scenario but also by the ever-changing improvements promoted by the software and hardware industries. While the computer industry has been evolving continuously, the power system scenario remained stable for decades. However, this scenario is now suffering revolutionary changes that require special attention of EMS developers.

Under the traditional scenario two generations of EMS have been developed. These generations mostly mirror changes in the computer industry.

The first generation of EMS appeared in the early seventies. Its computational architecture was based on redundant mainframes with very high costs and very low processing power for today's standards. Costs restricted the use of computers only to the most important power system control centers.

The limited processing capability demanded the development of a highly optimized code, which strongly used the specific features provided by each hardware and operational system. This included databases, user interfaces and even advanced applications. Therefore, the first generation of computerized power system control centers presented a deep interconnection among hardware and software. This approach allowed for the development of powerful systems running on very limited computers. It was very successful in enhancing the quality of power systems supervision and control.

However, the fast evolution of the computer industry brought serious problems to that generation of EMS. In a few years, most hardware, operational systems and support software became obsolete and disappeared from the market. Unfortunately, the dependency among hardware and software made very difficult, and often impossible, partial replacements or incremental evolution of the Energy Management Systems.

Companies soon discovered that their control centers used obsolete equipment, with low performance but high and ever rising maintenance costs - often with low reliability - and presented serious difficulties to adapt to the evolution of the requirements of the companies.

The second generation of EMS appeared in the beginning of the nineties and is still the solution currently available on the market. It reflects important transformations faced by the computer industry during the eighties. The major differences between the second and the first generation of EMS are related to the following characteristics: the widespread use of computer industry standards; the replacement of mainframes by a network of cheaper but powerful computers; and the rise of distributed processing. These characteristics motivated the use of the expression "open systems" to describe the second generation of EMS [8].

One of the main improvements exhibited by this generation of EMS as faced to its predecessor is the possibility of incremental growth. Equipment from different vendors can operate together and obsolete hardware can be replaced with minor impact. New functionality can be added easily, and the control center can evolve as needed by the company. These characteristics eliminate the problems of the previous generation and decrease the costs of acquisition and maintenance of equipment and of the evolution of the control center as a whole.

Those two generations of EMS, however, still have many similar characteristics concerning to their functionality. One of the most important similarities is the concentration of computer processing *inside* the limits of the control centers. These centers are essentially autonomous. Their connections with other partners, or even with other areas of the same company, are remarkably small. The processing, distributed or not, is strongly controlled and organized. In spite of the use of the expression "open systems" to characterize the second generation of EMS, these centers are almost closed in what regards to their interaction with other companies or with other sections of the same company. In this aspect, these first two generations of EMS are quite similar systems.

The new power system scenario, together with the expected evolution of telecommunications and the improvement of computer networks, will force substantial changes in EMS architectures. A number of new actors will start playing an important role which will have to be taken into account by the EMS, such as: regulatory agencies; energy markets; independent energy producers, large customers and suppliers; and companies dedicated to sell services (or even processing power) to the control centers. Strong integration with other areas of the same company will also be of the utmost importance in the competitive scenario. The operation of the EMS will have to consider other interests of the company and will not be only a technical issue anymore.

One of the results of this new scenario is that a significant amount of the computer processing required to operate the power system will take place *outside* the borders of the control centers. It is possible that some activities that are considered today as naturally belonging to the control center will be gradually transferred to other partners. The connection with different kinds of centers owned by companies dedicated to sell services may become necessary. Some examples are due to activities like emergency maintenance, weather forecast, patrimonial surveillance and others. The centers' borders, clearly defined today, will become fuzzy. They will not be the autonomous and almost independent entities that they presently are. The production of information will lose some of its importance to the negotiation of information produced outside.

The centers will not have control over their partners; actually, most of them will be able to join or leave the network, or even refuse to supply services or information, according to their current interests or availability. It can be expected that this uncertain environment will gradually reach the internal architecture of the control centers, where different software may be competing or cooperating to get the best possible results. The expression "open system" must be redefined to refer to these open and uncertain environments, where no central or complete control exists over the participants.

This new scenario poses great challenges to EMS developers. The whole architecture of the EMS must be adapted to this new definition of open system. Software components must be able to behave autonomously and opportunistically, while engaging in communication and negotiation using some common language. They will have to monitor the environment to try to accomplish their objectives. They will also have to cope with unreliable partners (some of them potentially hostile competitors) and lack of information, and still try to reach their goals. These challenges are found in many other information systems and should be understood from a more general point of view.

III. THE NEW SCENARIO BEHIND INFORMATION SYSTEMS

The dissemination of the UNIX operational system in the mid-seventies and the later diffusion of personal computers in the early eighties started a new trend which has affected information processing environments ever since, that is: decentralization. Many companies replaced their large mainframes by smaller PCs and workstations in a movement that was called downsizing. In order to allow different computer systems to work together, a great amount of effort has been dedicated to standardization. The creation of standards has contributed to the emergence of numerous LANs (Local Area Networks) and WANs (Wide Area Networks).

By the mid-eighties, standardization efforts start addressing problems that go beyond the interoperability of different platforms and operational systems and try to develop standard interfaces that are independent of programming languages and even network protocols. These interfaces shall allow the software industry to take a further step towards a decentralized world.

Instead of building large applications that model a whole enterprise, smaller software components with special capabilities must be devised to interact in a flexible and dynamic way in order to solve problems more efficiently. These components not only need to be able to inter-operate with other components within the limits of the enterprise, but shall also interact with other customers' and suppliers' systems dispersed across a network. Briefly, this new scenario where information systems must act has the following main characteristics [1]:

(i) heterogeneity: components on computers of different platforms, from different vendors, running different operational systems and using different protocols must interact;

(ii) distribution: interacting components may be geographically dispersed;

(iii) openness: components may become available or disappear at any time, according to their own interests;

(iv) dynamism: the environment where the components must act is constantly changing.

Distinct areas of computer science have suggested many different approaches to tackle, from their own perspectives, the problems that arise in this new decentralized scenario. In particular, the authors believe that Agent Technology, proposed in the context of Distributed Artificial Intelligence, is a very promising approach to support the construction of a new generation of EMS in an open environment.

IV. AGENT TECHNOLOGY

Although an increasing number of researchers have been discussing issues related to different aspects of agent technology since the last decade, the concept of an agent still lacks a formal definition. Generally speaking, agents are software entities that have the following main features [2]:

- *autonomy*: agents act without any direct external intervention and have some kind of control over their actions and internal state;
- *social ability*: agents interact with other agents via some agent communication language;
- *reactivity*: agents are able to perceive changes in the environment in which they are immersed and respond to those changes whenever necessary;
- *pro-activeness*: agents have their own goals and act not just responding to changes occurred in their environments but also trying to achieve their goals.

There are many different approaches to constructing multi-agent systems. Stressing out all the matters related to these different approaches is not in the scope of the present work. However, it is worth distinguishing between two general approaches regarding the overall architecture of multi-agent systems: the top-down and the bottom-up approaches.

In the top-down approach, the system has a pre-established organizational structure and the agents cooperate in order to achieve a common global predefined goal. On the other hand, in the bottom-up approach, there is no pre-established architecture and no predefined global goal. Agents have their particular goals and interact while pursuing them [3]. Considering that the emergent power system scenario is characterized by open environments in which partners may enter or leave at any time, the bottom-up approach reveals itself as the appropriate one in modeling the new EMS.

Still regarding agent technology, it is also worth pointing out that it is not intended to overcome the broadly accepted object oriented paradigm. Rather than that, the agent paradigm should be an extension of the object-oriented one. In this extension, a system is a collection of autonomous and interacting entities which should be conceived considering particular goals and whose internal state represent some "mental" state. Agents provide an appropriate abstraction level when one devises information systems to operate in distributed dynamic open environments.

V. AN AGENT-BASED FRAMEWORK TO EMS

The framework proposed in this paper is inspired in previous works by the two last authors in a different context [4, 5, 6]. Some of the ideas proposed in these works were further extended and adapted to the context of EMS [7].

In the present work, agents are defined as *objects* that:

- have their own goals;
- are capable of perceiving facts about their environment and act accordingly;
- communicate by means of a language which is independent of the specific context.

The definition of agents as a special kind of objects makes possible the transference of the available experience on object-oriented software development to the domain of agents.

A. Internal Structure of Agents

The internal structure of the proposed agents consists of the following elements:

(i) *Communicative Center*: responsible for message exchanges between the agent itself and the other agents in the environment.

(ii) *Sensors*: responsible for monitoring the physical and virtual environment in which the agent is immersed, allowing it to respond to changes whenever necessary.

(iii) *Memory*: represents the declarative knowledge of the agent or its beliefs.

(iv) *Behavior*: responsible for the achievement of the agent's goals; controls the other components of the agent and the agents that are directly subordinated to it.

(v) *Actuator*: responsible for effectively changing the environment through actions that result from the decisions taken by the Behavior.

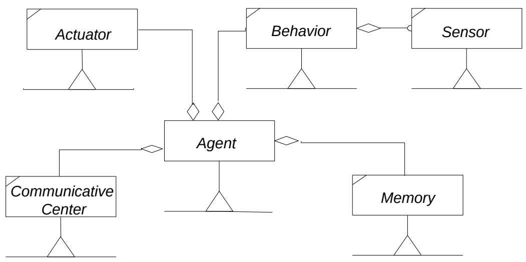

Fig. 1. Some abstract classes of the framework.

All these components are implemented as instances of specific object classes, which are aggregated into one object of the Agent class (Fig. 1). Characteristics derived from Object Orientation, such as inheritance and encapsulation of internal states and implementations, are fully used.

B. Communitary Aspects

A major distinction between traditional EMS and open systems EMS is that, in the last one, the existence of any agents cannot be presumed because they are free (at least in principle) to join or leave the network, and even to reject requests, according to their own interests or availability. In contrast, traditional EMS have a strictly controlled composition; the availability (or not) of software components is usually well known.

This suggests the use of some type of broadcast to start negotiations when a certain agent needs a service provided by another agent or agents. The existence and availability of a partner can be assumed only after a direct negotiation is started; this hypothesis is valid only while the negotiation is occurring or until the conclusion of the service requested. After that, it is not possible to presume that the partner is still available.

Another major difference is that, in open systems, there will often exist the possibility of multiple agents offering similar services. This does not occur in traditional EMS. That is, a request for a service may receive from 0 to n proposals, in some cases even from agents situated outside the control center. The requesting agent must select among those proposals; this will ideally be done considering costs, time, reliability and previous experiences, among other factors.

Finally, it is important to notice that there is no central representation of the system of agents. This improves the *reactivity*, a characteristic often considered necessary to deal appropriately with the uncertainties of the real world [4-7,9,10].

C. Types of Agents

So far we have described the general characteristics of the agents used in the framework. However, it is necessary to develop more specialized agent classes that are actually capable of useful actuation. This is done by the specialization of the abstract class Agent into other (still abstract) classes: Interface Agent, Application Agent, Data Agent, Mediator Agent and External Communication Agent. Those abstract classes will have to be derived again to originate concrete classes that will implement specific behaviors.

(i) *Interface Agents*: They control the interaction of the user with the system of agents. It is highly recommended that those agents are task-centered, that is, they should be organized around the tasks routinely performed by operators and not centered on specific computational applications.

(ii) *Application Agents*: These agents control the execution of applications, extract the relevant results and send them to the requesting agent or to a Data Agent. They can be used to encapsulate legacy applications.

(iii) *Data Agents*: These agents control the access to data of general interest to the system of agents. In open systems, that control may represent an important safety concern. Data Agents can encapsulate commercial databases.

(iv) *External Communication Agents*: They control the communication with agents situated outside the local computational system. Their tasks are related to delivery of messages and access control.

(v) *Mediator Agents*: They mediate the execution of activities that can be requested multiple times in short time intervals. They know the activities but don't have the means to process them. After receiving a request for a certain service, a mediator controls the instantiation of other agents (usually application agents) that are capable of performing the service requested, or at least parts of it. These agents disappear immediately after completion of their activities, while the mediator remains active and processing other incoming requests.

Mediators avoid problems that could occur if only one agent responsible for a certain service existed in the system. If a certain application agent, that takes about 5 seconds to perform its service, receives 15 simultaneous solicitations from different agents, the overall performance could be negatively affected. A mediator, however, would process those requests by instantiating 15 application agents across the network, on machines that presented the most favorable load/performance conditions. Those agents disappear after completing their services. In some situations, the time necessary to process the 15 requests could be similar to that necessary to process only one request in an overloaded computer. By using opportunistically the available computational resources, mediators can improve the overall performance.

D. Evaluation and Perspectives

Developing a large-scale evaluation of the proposed framework on a realistic system would take years of hard work, because of the size and complexity of EMS software. Even if this could be done, the resulting evaluation would not be totally conclusive, because the scenario for which it was conceived is not yet a current reality.

For these reasons it was decided to develop simplified prototypes that simulated some interactions among agents that could be considered typical of the future scenario. In spite of these simplifications, the prototype has demonstrated to be very important to the consolidation of some concepts and to the development of others. This suggests that tests and prototypes will play a major role on the evolution of agent-based EMS.

However, during the development of this work – as is common in many projects – it was sometimes necessary to make choices that could not be tested and compared against other alternatives. The adoption of behavioral decomposition, some aspects of the internal structure of the agents, the use of task-orientation for Interface Agents and the selected communication strategy are some examples. In these cases, choices were made based on the careful analysis of the alternatives, but no comparative tests were carried out. As these and other aspects become more intensively evaluated, it can be expected that the overall design and performance will greatly improve.

EMS are, for obvious reasons, systems where innovations have to be introduced carefully. Ideally, changes should be incorporated and tested incrementally. This is not an easy task, however, when the innovations derive from major technologic changes.

For example, reliable legacy applications should not be discarded; instead, they would better be adapted to the new architecture. In the proposed framework, those applications can be "wrapped" by application agents.

Other aspects, however, will continuously challenge EMS developers. Perhaps the intensive adoption of object-oriented techniques in the current EMS architecture will prove to be a satisfactory compromise, since objects are compatible with the traditional organization of control centers and can be extended to support the agent paradigm when it becomes necessary.

Changes in the power system scenario are fast, but gradualist. It can be expected that developers will try to adapt the current EMS architecture as long as possible, before it proves to be incapable of handling the complexities of the future scenario.

VI. CONCLUSIONS

The differences between the architectures of the first two generations of EMS have been determined mostly by the evolution of the computer industry, since the power system scenario remained stable. However, the next generation of EMS will be deeply influenced by the radical transformations that are taking place in the power system industry.

The introduction of new players, the decentralization of production and processing of information and the competition will change the way control centers operate and, consequently, their architecture. The controlled, organized and relatively isolated architecture of today's centers will become open and dynamic, and the connection with other players will be of utmost importance. This paper suggests that the uncertainties of that new scenario could be treated by the use of agents.

Agent technology has been suggested as a promising approach to power system control centers elsewhere [9]. However, by the time those proposals were developed it was not possible to anticipate the characteristics of openness and uncertainty of the new generation of Energy Management Systems.

This paper presents an agent-based framework that is closely aligned with the needs of this new generation. The experiments have shown that Agent Technology is an effective alternative to the next generation of Energy Management Systems.

VII. REFERENCES

[1] E. Cardozo, J. S. Sichman, and Y. Demazeau, "Using the Active Object Model to Implement Multi-Agent Systems", *Proceedings of the 5th. IEEE International Conference on Tools with Artificial Intelligence*, pp. 70-77, 1993, Boston, USA.

[2] M. J. Wooldridge, and N. R. Jennings, "Intelligent Agents: Theory and Practice", *Knowledge Engineering Review*, 1995: 10(2).

[3] J. S. Sichman, "DEPINT: Dependence-Based Coalition Formation in an Open Multi-Agent Scenario", *Journal of Artificial Societies and Social Simulation*, 1998: 1(2).

[4] M. Costa, B. Feijó, and D. Schwabe, "Reactive agents in behavioral animation", *Proceedings of the VIII SIBGRAPI*, pp. 159-165, 1995, São Carlos-SP, Brazil.

[5] M. Costa, and B. Feijó, "Agents with emotions in behavioral animation", *Computer & Graphics*, 1996: 20(3).

[6] M. Costa, and B. Feijó, "An architecture for concurrent reactive agents in real-time animation", *Proceedings of the IX SIBGRAPI*, pp. 281-288, 1996, Caxambu-MG, Brazil.

[7] G. P. Azevedo, "Reactive Agents for Power System Control Centers", *D.Sc. Thesis* (in Portuguese), PUC-Rio, August 1998, Rio de Janeiro, Brazil.

[8] L. C. Lima, P. A. Machado, A. L. Oliveira Filho, L. A. C. Pereira, and G. P. Azevedo, "Design and Development of an Open EMS", *Proceedings of IEEE Athens PowerTech'93*, 1993.

[9] S. Taludkar, V. C. Ramesh, R. Quadrel, R. Christie, "Multiagent Organizations for Real-Time Operations", *Proceedings of the IEEE*, vol. 80 n.5, May 1992.

[10] F. C. Cheong, "OASIS: An Agent-Oriented Programming Language for Heterogeneous Distributed Environments", *D.Sc. Thesis*, Michigan University, 1992.

[11] E. V. Berard, "Essays on Object-Oriented Software Engineering", Volume 1, Prentice-Hall, 1993.

VIII. BIOGRAPHIES

Gilberto Pires de Azevedo was born in Rio de Janeiro in 1960. He received his B.Sc. in Electrical Engineering from PUC-Rio in 1984, his M.Sc. degree from COPPE-UFRJ (also in Electrical Engineering) in 1989, and his D.Sc. degree in Computer Science from PUC-Rio in 1998, all in Rio de Janeiro, Brazil. He has been with CEPEL (a Brazilian Power Systems Research Center) since 1982. His current interest areas include Power System Operation and Control, Human-Computer Interaction and Agent Technology.

Bruno Feijó received his B.Sc. degree in Aeronautical Engineering from ITA in 1975, M.Sc. degree from PUC-Rio in 1980 and PhD degree from Imperial College in Intelligent CAD in 1988. He is an Associate Professor at the Computer Science Department of PUC-Rio and Director of the Intelligent CAD Laboratory (ICAD). His areas of interest are CAD, Computer Graphics, Animation and Agent Technology.

Mônica Costa received her B.Sc. degree in Mathematics, her M.Sc. degree in Computer Science and her Ph.D. degree in Computer Science from PUC-Rio in 1989, 1993, and 1997, respectively. She joined PUC-Rio's Computer Science Department in 1998, as an Assistant Professor. Her research activities currently focus on Human Modeling and Simulation. She is currently supported by the Brazilian Council for the Development of Research and Technology (CNPq), as a post-doctoral visiting scholar at the Computer and Information Science Department of University of Pennsylvania.