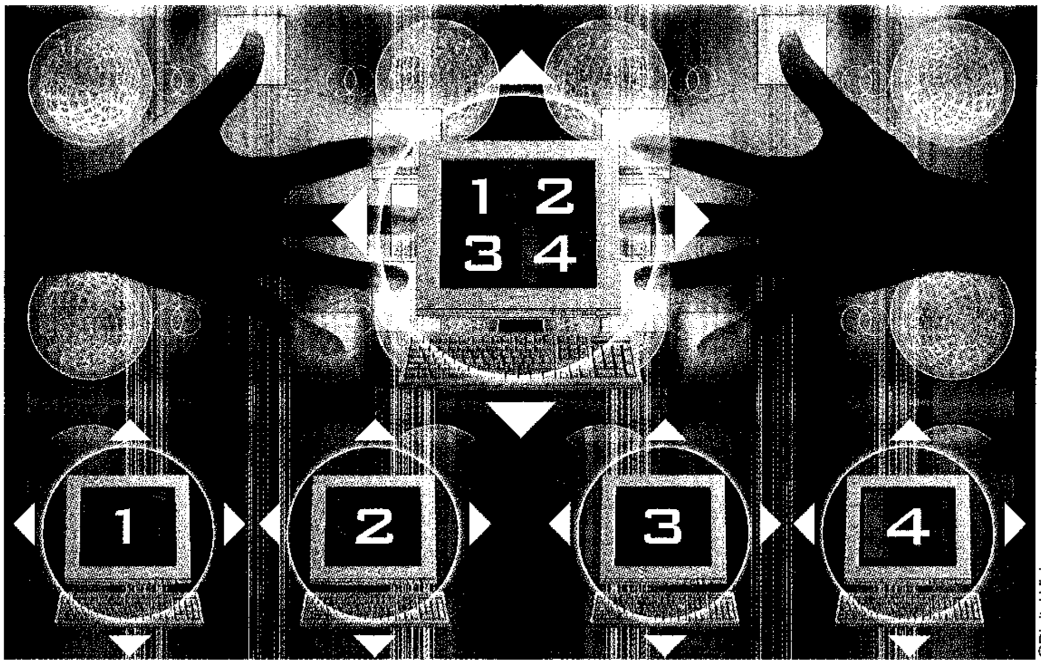

©Digital Vision

Control Centers Evolve with Agent Technology

Gilberto P. Azevedo,[1] Bruno Feijó,[2] Mônica Costa[2]

Energy management system (EMS) architectures are deeply influenced by power system and information system scenarios. The computer industry has been evolving continuously, and the power industry, which remained relatively stable for decades, is now undergoing revolutionary changes that require the special attention of EMS developers.

The introduction of new players, the decentralization of production and processing of information, and competition will change the way control centers operate and, consequently, their architecture. Distinct areas of computer science have suggested many different approaches to tackle the problems that arise in the new decentralized scenario. Of particular interest, *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.

The controlled, organized, and relatively isolated architecture of today's control centers will become open and dynamic, and the connection with other players will be of utmost importance

Power System Scenarios

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 1970s. Its computational architecture was based on redundant mainframes (Figure 1), which had very high costs and very low processing power by today's standards. Costs restricted the use of computers to only 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

[1] CEPEI.
[2] Pontifícia Universidade Católica do Rio de Janeiro

ISSN 0895-0156/00/$10.00©2000 IEEE

and operational system. This included databases, user interfaces, and even advanced applications. Therefore, the first generation of computerized power system control centers presented an intimate interconnection between 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 system 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 partial replacements or incremental evolution of those systems very difficult and often impossible. Companies soon discovered that their control centers used obsolete equipment, with low performance, high and ever-rising maintenance costs, often with low reliability, and this presented serious difficulties to adapt to the evolution of the requirements of the companies.

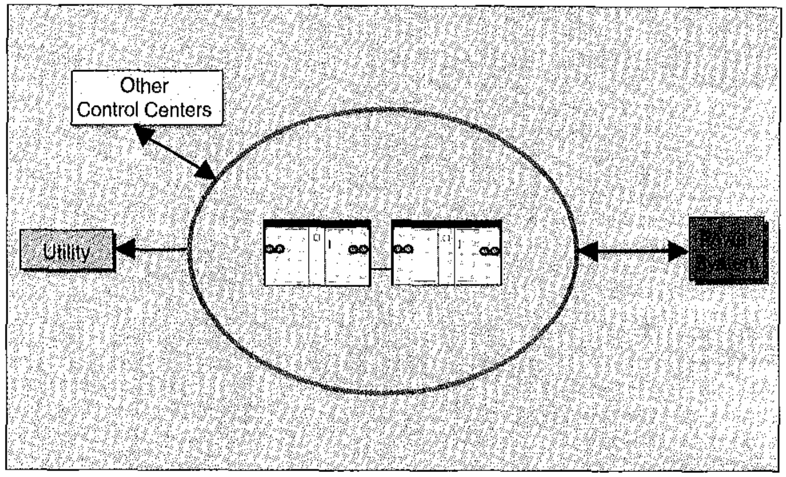

Figure 1. First generation control center: autonomous, based on mainframes, with weak links to other players

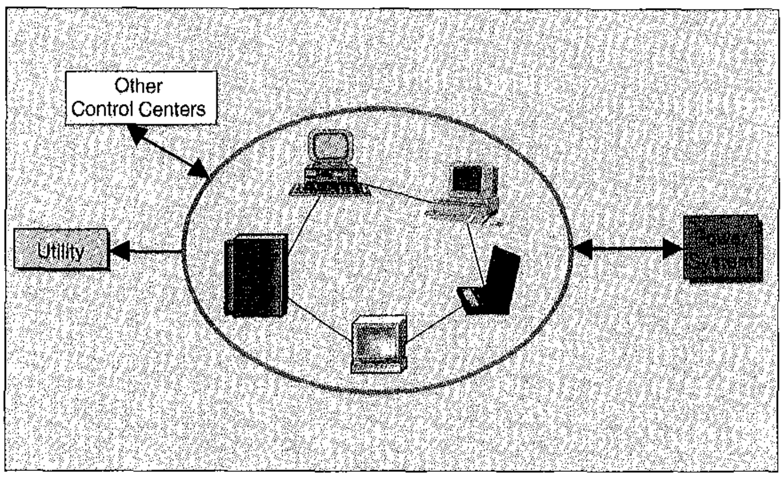

Figure 2. Second generation control center: distributed and heterogeneous hardware, autonomous, with weak links to other players

The second generation of EMS appeared in the beginning of the 1990s and is still the solution currently available on the market (Figure 2). It reflects important transformations faced by the computer industry during the 1980s. The major differences between the first and second generation of EMS are related to the following characteristics:

- Widespread use of computer industry standards
- Replacement of mainframes by a network of less costly yet powerful computers
- Rise of distributed processing.

These characteristics motivated the use of the expression "open systems" to describe the second generation of EMS.

One of the main improvements exhibited by this generation of EMS, in contrast 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 their functionality. One of the most important similarities is the concentration of computer processing inside the limits of the control centers. The operation is driven mostly by technical criteria, and 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 with regard 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. We can say that the traditional scenario of the power system market, with its rigid organization, has been mirrored in the structure of both generations.

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 (such as regulatory agencies, energy markets, independent energy producers, large customers and suppliers, companies dedicated to sell services to the control centers, and others) are starting to play an important role that must be taken into account by the EMS. In many parts of the world, large utilities are being split into smaller and more specialized ones, and they will often have their own control centers with specific goals. At the same time, strong integration with other areas of the same company will also be of the utmost importance in the competitive scenario. The final result is that the operation of control centers will have to be directly attached to the other interests of the company, will require a large amount of interaction with external partners, 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 activities that are considered today as naturally belonging to the control center (and also activities that still don't exist in today's centers) 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 such as 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. One consequence is that the *production* of information will lose some of its importance to the *negotiation* of information produced outside.

The centers will have little (if any) control over their partners; actually, many 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. If we remember that the traditional power system scenario influenced the structure of the previous generations of EMS, it can be expected that this new and uncertain environment will gradually reach the internal architecture of the control centers, where different software may be competing or cooperating to achieve their goals. The expression *open system* must be redefined to refer to these open and uncertain environments in which 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 systems (Figure 3). Software components must be able to behave autonomously and appropriately, 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.

Information System Scenarios

The dissemination of the UNIX operating system in the mid-1970s and the later diffusion of personal computers in the early 1980s started a new trend that 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 local area networks (LAN) and wide area networks (WAN).

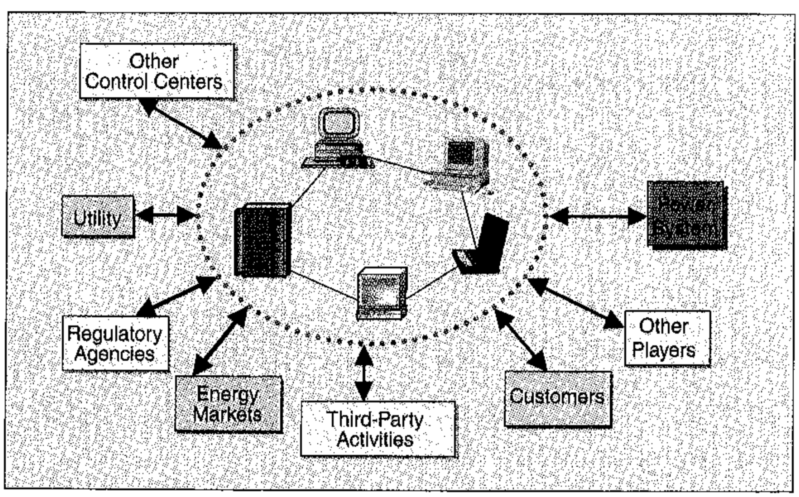

Figure 3. The next generation: distributed and heterogeneous hardware, with strong external links and fuzzy borders

By the mid-1980s, standardization efforts started addressing problems that go beyond the interoperability of different platforms and operational systems and tried to develop standard interfaces that are independent of programming languages and even network protocols. These interfaces should 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 interoperate with other components within the limits of the enterprise, but also interact with other customers' and suppliers' systems dispersed across a network. Briefly, this new scenario in which information systems must act has the following main characteristics:

- Heterogeneity: components on computers of different platforms, from different vendors, running different operational systems and using different protocols must interact
- Distribution: interacting components may be geographically dispersed
- Openness: components may become available or disappear at any time, according to their own interests
- Dynamism: the environment in which the components must act is constantly changing.

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.

- 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.
- Proactivity: Agents have their own goals and do not just act in response to changes that have occurred in their environments. They also initiate action to try to achieve their goals.

There are many different approaches to constructing multi-agent systems. Detailing all of the items 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 preestablished 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 preestablished architecture and no predefined global goal. Agents have their particular goals and interact while pursuing them. 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.

It is worth pointing out that agent technology is not intended to overcome the broadly accepted object-oriented paradigm; rather, 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 that should be conceived considering particular goals and whose internal state represent some "mental" state. Agents are not a new revolutionary solution to a large range of problems, but they provide an appropriate abstraction level when one devises information systems to operate in distributed, dynamic, open environments.

Agent-Based Control Centers

The next generation of control centers will be required to deal with a number of new practical problems.

- Constant evolution of goals, methods, and requirements: The slow-changing scenario for which the previous generations were designed is being transformed into a dynamic one. The design of the next generation will have to allow for the constant evolution of goals, methods, and requirements. These will require a very modular design of its software for easy replacement, introduction, or removal of components; otherwise, the control centers will be short-lived.
- Varying computational load: The rigid organization of the computational environment of the previous generations led to relatively stable computational loads. This will change in an environment in which independent external partners will play an important role. The structure of the control center must be able to accommodate large changes in computational load.
- Operation closer to power system limits: Economic concerns will require the operation to be as fine tuned as possible. Systems that are better controlled can work closer to their limits without increasing the risks involved, thus increasing profits and reducing the need for new investments. Control center operators must have easy access to the information they need (but avoiding information overload), without having to deal with multiple programs and user interfaces. User interfaces must be designed taking into account the actions the operators need to perform instead of specific application programs.
- Fast obsolescence of computational technologies: The same challenges faced by the previous generations will still exist, and probably with increasing speed. The structure of the software must be as open and modular as possible to extend the life of the control center design.
- Geographical distribution: The concept of a control center as a delimited physical environment will be gradually replaced by a more distributed one, including different sectors of the company and external partners.

Agent technology provides abstraction levels appropriate to handle these problems in an easier way than other traditional software development approaches, because agents are naturally suited to operate in open systems. There are many different alternatives to develop agent-based control centers. But, given the problems discussed previously, we believe that such agent architectures must include the following features.

- User interfaces and application programs must be decoupled. This is essential to meet the requirements of evolution and modularity.
- User interfaces must be task-oriented. Instead of being developed around application programs, interfaces should encompass the tasks actually performed by the operators.
- Intelligent distribution of computational load is necessary given the characteristics of the computational load in open systems.

In the next generation of power system control centers, agents can be defined as objects that:

- Have their own goals
- Are capable of perceiving facts about their environment and acting accordingly
- Communicate by means of a language that is independent of the specific context.

The research being carried on at CEPEL and Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), in Rio de Janeiro, has revealed the need for the following types of agents:

- Interface agents that control the interaction of the user with the system of agents. Those agents should be task-oriented, i.e., they should be organized around tasks routinely performed by operators and not centered on specific computational applications. Figure 4 compares a user-oriented model, an application-oriented model, and a task-oriented model. In the first model, there is a single agent taking care of several applications. In the second one, similarly to conventional interfaces, each application has an interface agent. In the last one, the agents are defined in terms of the tasks to be performed. In the task-oriented model, the interface must be designed taking in account the actions the operator needs to perform instead of specific application programs.
- Application agents control the execution of applications, extract the relevant results, and send them to the requesting agent. They can be used to encapsulate legacy applications.
- Mediator agents mediate the execution of activities that can be requested multiple times in short time intervals. They know these activities but don't have the means to process them. After receiving a request for a certain service, a mediator controls initiating the performance of other agents (usually application agents) that are capable of performing the service requested (Figure 5), 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.

Agents are software entities that feature autonomy, social ability, reactivity, and proactivity

Other types of agents not described here might include data, translator, and external communication agents.

Mediators avoid problems that could arise 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 initiating up to 15 application agents across the network on machines that presented the most favorable load/performance conditions. Those agents would disappear after completing their services. In extreme situations, the total time to process the 15 requests could be similar to that necessary to process only one request in an overloaded computer. By using the available computational resources at the proper time, mediators can improve the overall performance.

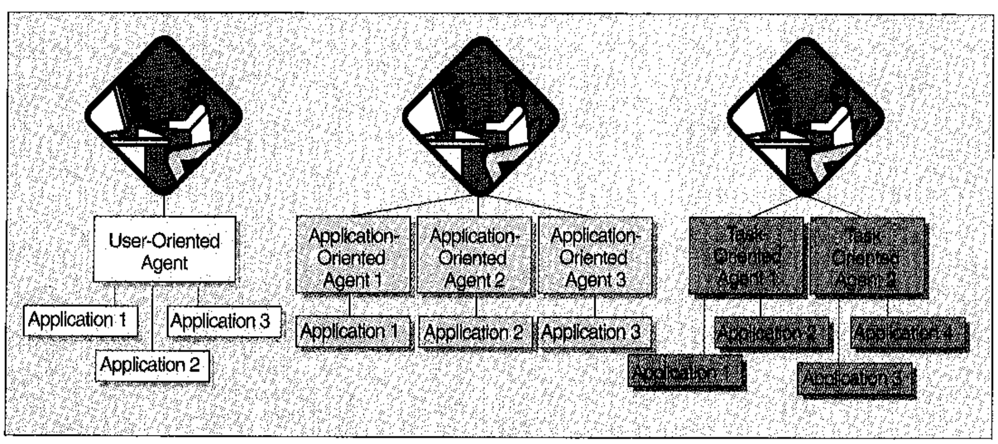

***Figure 4.** User-oriented interface agents are difficult to develop in control centers. Application-oriented interface agents require the knowledge of different applications. Task-oriented interface agents are an efficient approach for control center operators.*

A major distinction exists between traditional

EMS and open systems EMS. In open systems, 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 from other agents, 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 an 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 taking place or until the conclusion of the service requested. After that, it is not possible to presume that the partner is still available.

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, often 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.

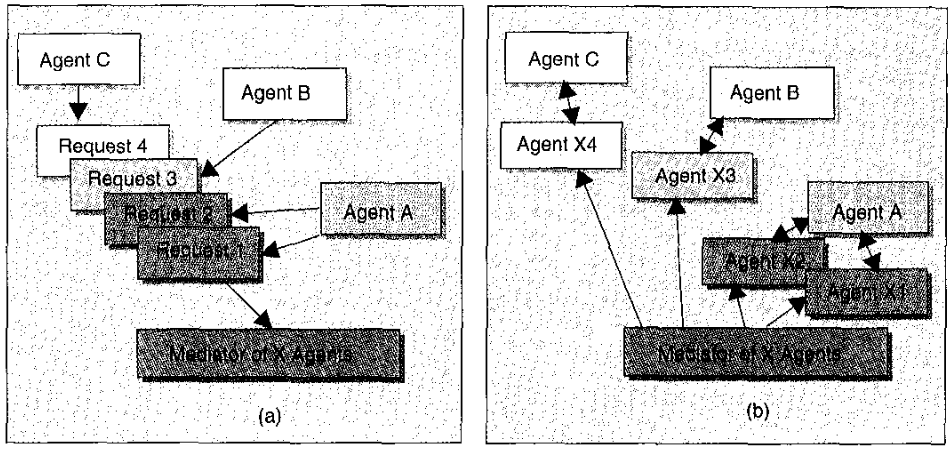

Figure 5. (a) Three agents send four almost simultaneous requests to the mediator of agents of type X. (b) The mediator creates four instances of X agents in the network, which disappear after completing their tasks. The mediator remains alive and answers new incoming requests.

Evaluation and Perspectives

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.

EMSs are, for obvious reasons, systems in which innovations have to be introduced carefully. Ideally, changes should be incorporated and tested incrementally. Changes in the power system scenario are fast, but progressive. 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 ever-increasing complexities of the new scenario.

Agent technology has been suggested as a promising approach to power system control centers elsewhere. 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 EMS. The introduction of new players, the decentralization of production and processing of information, and 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. Agents actually are a promising alternative to deal with the uncertainties and challenges of that new scenario.

For Further Reading

M.J. Wooldridge, N.R. Jennings, "Intelligent agents: Theory and practice," *Knowledge Engineering Review*, vol. 10, no. 2, 1995.

S. Taludkar, V.C. Ramesh, R. Quadrel, R. Christie, "Multiagent organizations for real-time operations," *Proceedings of the IEEE*, vol. 80, no. 5, May 1992.

G.P. Azevedo, B. Feijó, M. Costa, "An agent-based approach to EMS in open environments," in *Proceedings of 1999 IEEE PowerTech*, Budapest, August 1999.

L. Jamnicsky, "EMS network security applications of the future," *IEEE Computer Applications in Power*, April 1996.

Biographies

Gilberto Pires de Azevedo received his BSEE from PUC-Rio in 1984, MSEE from COPPE-UFRJ in 1989, and DSc degree in computer science from PUC-Rio in 1998, all in Rio de Janeiro. 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 artificial intelligence, with a special interest in agent technology. He may be reached by E-mail at gilberto@cepel.br.

Bruno Feijó received his BS in aeronautical engineering from Instituto Tecnológico da Aeronáutica in 1975, MS in dynamic analysis from PUC-Rio in 1980, and DSc in intelligent CAD from Imperial College in 1988. He is currently a professor in the Computer Science department of PUC-Rio and director of the Intelligent CAD Laboratory. His interest areas include CAD, computer graphics, and animation and agent technology. He may be reached by E-mail at bruno@inf.puc-rio.br.

Mônica Costa received her BS in mathematics, MS in computer science, and PhD 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/simulation and agent technology. She may be reached by E-mail atmonica@inf.puc-rio.br.