Data Gardens: Agent Societies for visualization of complex system

V. Renault *,**
* France Telecom R&D
38-40, rue du Général-Leclerc
92 794 Issy Moulineaux Cedex 9
France

G. Hutzler **
** LIP6, case 169 (OASIS-MIRIAD)
Université Paris 6 - 4 place Jussieu
75 252 Paris Cedex 05
France

Abstract *Complex systems, and more generally complex data, entail significant problems of visualization, due to their dynamic and distributed nature, and the multiplicity of their relations and interactions. How should we represent, for instance, a biological system such as an ant colony or an economic system such as the Stock Market? However visualization of complex data is an efficient way to meet needs in both theoretical and practical frameworks. Indeed, in the framework of simulation of complex systems, it facilitates analysis and design. Secondly, in the framework of industrial systems, it enables strategies for control and decision-making. The solution suggested in this paper requires multi-agent techniques, used to allow a dynamic construction of an interface of visualization. Within the Data Gardens, each interface agent carries out a small portion of information. The agents are organized collectively to process input hierarchically according to the priorities given by the user to the pieces of data.*

Keywords: Complex systems, multi-agent systems, visualization and representation of information, hierarchy

1 Introduction

Various systems can be qualified of complex [11]. We can mention, higgledy-piggledy, biological systems such as an ant colony or an ecosystem, economic systems such as the Stock Market or computer systems such as the Internet network. These examples show that even if it is not possible to give a general and final definition of what is a complex system, it is at least useful to specify which one we use. A complex system is generally defined as a set of entities, linked together and organized in a structure. The goal of this paper is to add a dynamic component in this definition. Actually, complex systems are especially characterized by their parallel, distributed and opened evolution, which introduces dynamism at all levels, from the most elementary entity of the system to the highest level of organization, by the way of the interactions which ensures this micro/macro link [5].

Furthermore, in the contexts of on one hand analysis and design, and on the other hand control and decision, the visual and auditory representation of a system constitutes an essential means of comprehension of its functioning. In particular, it can make perceptible the processes by which the actions and interactions of all the entities result, at the global level, in emergent chaotic or evolutionary behaviors [1]. However the construction of these representations creates numerous problems, which derive not only from the characteristics of the systems themselves but also from the perceptive processes of the users for which they are intended.

On account of this statement, we approach the problem of visualization of complex systems as a dynamic construction of an interface rather than a design a priori of more or less static interfaces. In other words, the problem is to give to the interface the means of reorganizing itself, according to the changes which appear within the system, as well as those due to the user and his concerns. In this way, we present a solution based on the use of multi-agent systems [5], themselves complex systems. We show how to conceive a dynamic interface, first by giving the agents the necessary knowledge for describing the complexity of the visualized system, next by enabling them to dynamically access information about the evolution of the system. Symmetrically, the agents know how to obtain information concerning the user and they can react to his actions.

In section 2, we begin to introduce the problems of visualization of complex systems. Next, in section 3, we bring up the software architecture. Section 4 describes how simple knowledge and strategies of cooperation can be given to the agents, enabling the interface to visualization can evolve and adapt itself, in real time. This last set of methods is the core of the industrial project of the Data Gardens [9].

2 Visualization of complex systems

2.1 Problems

As we point out above, an essential aspect of the complex system is the dynamics. They occur on the level of the entities which could be not only active but also evolutionary, as well as in their behavior as in their shape. It is the case, for example, of a worker ant which proceeds through the stages of egg, larva and cocoon, before its mutation into a functional worker. This ant could be able to assume successively the task of care of eggs, the foraging, and finally the care of cocoons [1]. Dynamics occur, next, on the level of the interactions established between two entities: these interactions can appear at a given time, then change and altogether disappear. For example, in the case of the ants, these interactions are dynamically created and demolished, according to the tasks, the needs, the positions of the various protagonists, and also according to global changes in the environment or the structure of social relations [1]. Consequently, the social structure of the nest itself is modified, introducing dynamics at the organizational level.

In addition to the dynamics, the conception of a means of visualization adapted to complex systems brings forth many other problems. Among them, two kinds of problems seem important to us: firstly, the potentially distributed and opened nature of these systems imposes a significant flexibility and modularity. Secondly, the multiplicity of possible levels of analysis leads to confusingly versatile means of representation. For example, the Internet network is composed of a great number of computers, linked together, with which many users can exchange and access vast stores of information. All of this is executed in a totally distributed way. The visualization of this system requires collecting data describing the network from different sources, and integrating them into one only system of representation. In other respects, the network is opened: computers, users or information can dynamically come in and come out. This makes it necessary to have the ability to dynamically add and delete data sources in the representation system.

Moreover, complex systems are characterized by the fact that their levels of complexity are organized hierarchically [11]. A biological ecosystem, for example, consists in animal and vegetal organizations, themselves compounds of organs, cells, molecules, atoms, etc. According to the required level of analysis, the interface has to offer various levels of abstraction in the presentation of information [10]. This should be achieved through continuous filtering and organizing information in a hierarchy. The aim is to obtain a mode of representation which does not overload the user with unnecessary information. Consequently, the agents have to adapt their mode of representation to the users task at a given time.

2.2 Existing solutions

These various constraints of dynamism, modularity, and hierarchically structured processes are not new. Nevertheless, until recently they had a limited impact in the conception of visualization interfaces. Existing solutions stem from various fields, in an array ranging from the interfaces for complex industrial systems [10] to the simulation of complex systems, and including data visualization and mining [7]. Within the framework of complex industrial systems, the basic tool is the synoptic [10], which provides a synthetic view of a process, according to a layout close to the physical layout of the process. The various indicators associated with the operation of the system are, as for them, synthesized in numbers, charts or graphs. This solution is suitable for systems which have fixed structure, but does not permit the analysis of indicators on a hierarchical basis, for example according to their importance from the viewpoint of the security of the operation.

Data visualization and data mining frameworks have developed numerous techniques for representing, usually in synthetic forms, huge sets of data [7]. The user manipulates maps, trees or detail and context representations; he travels around 3D landscapes, and many other metaphors allowing him to be immediately familiar with the structure of the visualized information. All these techniques offer a huge choice in the way of representing complex data. Unfortunately, these techniques often restrict themselves to a visualization, at a given time, of a fixed set of data which summarizes the last evolution of the system, over a more or less lengthy period. This precludes their use in a dynamic context. In contrast, the framework of simulation offered by complex systems, and particularly multi-agent systems, provides all the techniques necessary for taking into account these dynamic aspects, while often excluding the problematic of the visualization of simulated systems.

3 Data Gardens: towards multi-agent interfaces of visualization

As we show, the synthesis between the techniques of multi-agent simulation and of visualization provides new solutions for the problem of representation in real time of complex systems. The Data Gardens project is the fruit of this synthesis, using virtual ecosystems to filter and organize hierarchically a flow of complex data [9].

3.1 General principles

First, it can be useful to specify what multi-agent systems are before studying how they can constitute a fitting answer for the above problems. We define a multi-agent system (MAS) as a set of agents which interact together in a shared environment [4]. The agent is regarded as a real or virtual autonomous entity which evolves in an environment, can perceive this environment, can act in this environment, can communicate with other agents, and presents an autonomous behavior, which is the consequence of its knowledge, of its interactions with others agents, and of its goals [4]. MAS research is based on a double aim: firstly, the realization of distributed artifacts able to achieve complex tasks by cooperation and interaction, and secondly, theoretical and experimental analysis of the mechanisms of self-organization, mechanisms which take place when autonomous entities interact [5].

Multi-agent systems can themselves be regarded as complex systems. Dynamics is made possible by the autonomy of the agents in relation with their environment (we include in the concept of environment the physical or simulated environment, as well as the other agents and the user of the system). This autonomy introduces dynamics in the interactions of the agents and also in the organization. As for modularity, it is an inherent property in these systems, linked to their division in agents. Finally, the concept of agent is sufficiently gen-

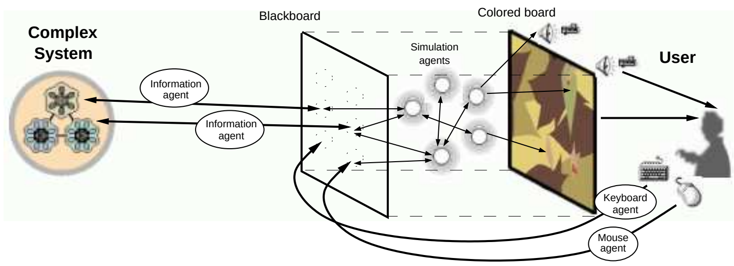

Figure 1: General architecture of the representation system.

eral to authorize various levels of decomposition: the choice of the entities and of their interactions and the level of division can be adapted at the desired level of abstraction of the interface.

3.2 Information - Filtering Visualization

The architecture used for building the multi-agent interface of representation (see Figure 1.) rests on the use of two kinds of agents: first, information agents have to gather data characterizing the complex system being represented, secondly the representation agents have to filter and treat on a hierarchical basis this data before representing it graphically. Each information agents is associated with a data source. They function together as an information system whose task is to gather in real time the data required for the description of the systems functioning. They convey this data to the system of interface itself through a structure analogous to a blackboard [5]. Then, the information is controlled by the representation agents. It should be noticed that in this scheme of operations, the user of the interface is also assimilated to a system, and is observed as a data, in particular through his mouse and keyboard actions. Two information agents are thus specifically responsible for gathering the mouse and keyboard data; causing specific reactions at the level of the representation agents, in the same way as they will react to the data of the complex system.

By way of the blackboard, the representation agents receive a huge flow of data, data which they have to represent as a colored board. In the simplest case, one representation agent is associated with one entity of the complex system, and the agents aspect matches the kind of entity it represents. Nevertheless, the approach yields more interesting results when the agents have knowledge enabling them to adapt their visual representation according to the current state of the entities, to their kind of functioning, to the interactions between all kind of entities, and to the groups which are dynamically assembled and dissolved. In return, it is possible to equip these same agents with the capacity to adjust their representation according to the actions of the user. From a user viewpoint, this means that one has the power of directly acting in real time on the interface of visualization, and therefore of orienting the final appearance of the representation.

4 Application to the design of an interface for monitoring road traffic

The above architecture is merely a framework for the conception of various multi-agent interface systems. Implemented as a software design core for multi-agent systems, it does not require any knowledge nor behavioral model for the representation agents. All knowledge or behaviors are specified through a little scripting language. With this language, the interface

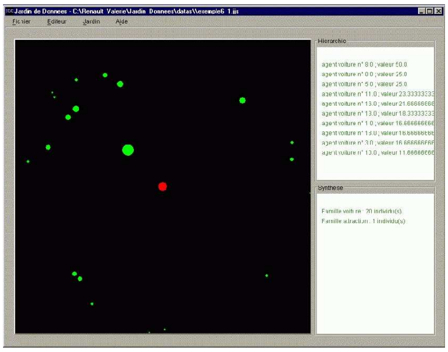

Figure 2: The agents whose the dangerousness is most significant are attracted by the center of the screen, pushing back the others towards the periphery.

designer also describes the strategies of representation that the agents shall use. In the simplest case, an agent can be assigned to track a particular variable, and adjusts its size or its color according to the value of the variable. The drawback of this first approach is that the display may end up drawing a multitude of unrelated graphic indicators. In this situation, a global and synthetic view would be difficult to obtain. It is thus necessary that the various agents take into account not only the information that they have to represent, but also of the context in which they are, i.e. the information which the other agents are presenting.

The advantage of using multi-agent systems appears clearly in this framework. These systems enable agents to interact, cooperate or negotiate in order to dynamically determine the most significant information at a given time. Drawing inspiration from processes of social hierarchical organization in animal groups [2], we examine how similar processes can be introduced in agent societies to hierarchically organize the flow of data according to various criteria of priority. We associate an agent with a hierarchical level based on the priority given to the data the agent must represent. It is then easy to integrate the user in the mechanism of the construction of the interface: for example, the user can define his own priorities, and the system reorganizes itself dynamically each time these priorities are altered. Finally, the hierarchical level of an agent is a function depending on three kinds of information: first, the data values describing the road traffic, next the interactions between the agents and the history of these interactions, and finally the modifications caused by the user.

4.1 From data to agents

We present here an example of the visualization of cars on a freeway. The data known about these vehicles are their speed and their position. Our aim is to achieve to visualize situations known as abnormal and considered dangerous in this context. Dangerous situations are for example a car driving faster or slower than the others.

To keep an intuitive relationship with the road system and its representation, we preserved a correspondence between the representation agents and the vehicles. However, the position of an agent on the screen does not reflect the matching vehicles position on the freeway section but matches instead its degree of abnormality. The degree of abnormality of a car is computed as the difference between the speed of a vehicle and the average speed of the vehicles which surround it. The more abnormal an agent is, the more it will be attracted towards the center of the screen, pushing back the normal agents towards the periphery (see Figure 2.). Therefore, the environment in which the agents evolve no longer matches the physical environment of the vehicles, but constitutes a spatially structured environment in which the agents will be able to interact and communicate. The agents communicate locally via this environment by stimuli. By this communication, they hierarchically organize themselves in a completely distributed way, according to their degree of abnormality.

4.2 Hierarchical processes and local interactions

To implement these dynamical hierarchical processes of agent societies, we use, as often

```
car.law("calcul_index", 1, 0, "if ( norm($car_dominance_s) > $thresholds) {
    $fight = random();
    if ($fight <= $index_dominance) {$issue_K = 1;};
    if ($fight  >  $index_dominance) {$issue_K = 0;};
    $car_dominance = $car_dominance + ($issue_K - $fight)*0.01;
    $thresholds = $thresholds - ($issue_K - 1 + $fight) * 0.01;
    $index_dominance = $car_dominance / (norm ($car_dominance_s) + car_dominance);
};");
```

Figure 3: Extract of the scripting language translating local interaction of the agents.

in the multi-agent field, ethological models [1]. These models describe the processes of organization at some definite stage of the life of a group: during the sociogenesis process, when the leader dies, or when a new member enters the group. The model of organization developed in [8] is our starting point to define interactions between our agents. This model defines the organization of a society of bees and the distribution of the bees in the nest according to their hierarchical level. This rank is established by a series of fights between the bees. The extract of the scripting language introduced on Figure 3 shows the translation of this ethological model of behavior to a behavior for the representation agents. The rank of an agent is computed, in addition to its degree of abnormality, according to the result of the agents fight and to the history of these flights.

These interactions can be interpreted thus: an agent diffuses a stimulus of predominance in its environment (car_dominance), and detects the stimuli emitted by the other agents in its immediate environment (car_dominance_s). If the stimuli that it detects exceed its tolerance level (thresholds), a fight takes place between the agents. The result of the fight, although random, is influenced by the index of predominance (an agent initially strong stands a better chance of being the winner than an initially weak agent). Finally, the outcome of the fight modifies the agents tolerance level and its predominance index.

4.3 Interaction with the user

Since we describe a method for visualizing a hierarchy of the level of priority of a set of information, the interest is to allow the user to inform the system of what he wants to visualize at one precise time. The platform architecture gives the possibility to take into account the user as a particular data source, so the agents can be made sensitive to the actions of the user with the mouse. This is done by giving each representation agent an internal parameter (user_action) which will modulate the interaction or the repulsion compared to the center of the screen: if the user moves an agent away from the center of the screen, the degree of attraction of the agent towards the center will be decreased proportionally with the action of the user. If the functions defined in the scripting language make it possible to keep an internal coherence in the system with the disturbances induced by the data, the introduction of the user as an agent which can interact with the others, also makes it possible to introduce in the system an external coherence: by interacting and by modifying the weight of some data, the user can modify the representation which he has of the systSem, and thus keep a representation which is coherent for him.

5 Conclusion

We present in this paper a new model for building interfaces of visualization for complex data flows. This model is established in a platform of multi-agent design. It take into account the dynamic aspect of these data, which is, in our opinion, a very significant aspect although often neglected. At the same time, this model permits a modular, adaptive and opened design of interfaces. The example, although relatively simple, shows the validity of our approach considering that the building of an interface to visualize dynamical information can be drawn in parallel with the building of a society of agents of representation. Moreover, the presented model does not make any assumption on the nature of the information represented, which makes it possible to consider applications in various fields. Within our work in progress, we are interested in particular in the filtering of electronic mail. From the point of view of classification, the mails will be grouped according to various criteria of origin, destination, date, or subject. From the point of view of processing, the criteria selected will be associated with the priority with which each mail must be treated. It will be necessary, within this framework, to develop the model presented so as to allow the synthesis of information by the regrouping of various agents. By introducing the concept of groups, we will allow the dynamics to change of level of representation. Also the role of the user will have to be reinforced, giving to the system of interface the means of learning the preferences and the practices from the user, so as to adapt the process of filtering and regrouping. Thus, we wish to set up a real co-evolution between the user and the system of interface: by adapting the interface to the user, the user reduces his effort to adapt himself to the interface.

References

[1] G. Bonabeau & E. Theraulaz. *L'intelligence Collective.* Hermès, Paris, 1996

[2] R. Chauvin. *Les Sociétés Animales.* P.U.F., Paris, 1982

[3] E.H. Chi & all. Visualizing the Evolution of Web Ecologies. *Proceedings of Computer Human Interaction'98 (Los Angeles), ACM Press*, 400–407, 1998

[4] Y. Demazeau & J. Ferber. Introduction to Multiagent Systems. *ICMAS'98 Tutorial*, 1998

[5] J. Ferber. *Multi-Agent Systems.* Addison-Wesley, 1999

[6] D. Gelernter. *Mirror Worlds - The Day Software Puts the Universe in a Shoebox...How it will Happen and What it Will Mean.* Oxford University Press, Oxford, 1992

[7] N. Gershon & all. Information Visualization. *CHI'98 Tutorial Notes 3, ACM Press, New York*, 1998

[8] P. Hogeweg & all. The ontogeny of the interaction structure in bumble bee colonies: a MIRROR Model. *Behav. Ecol. Sociobio*, 12: 271–283, 1983

[9] G. Hutzler. Du Jardin des Hasards aux Jardins de Donnes: une approche artistique et multi-agent des interfaces homme/systèmes complexes. *thèse de doctorat, Université Paris 6*, 2000

[10] C. Kolski. *Interfaces homme-machine - application aux systèmes industriels complexes.* Hermès, Paris, 1997

[11] J.-L. Le Moigne. *La modélisation des systèmes complexes.* Dunod, Paris, 1990

[12] A. Moukas. Amalthaea: Information Discovery and Filterinf Using a Multiagent Evolving Ecosystem. *Proceedings of PAAM'96, London*, 1996

[13] C. M. Mitchell. Human interaction with complex systems: Design issues and research principle. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 27:3, 265–273, may 1997

[14] H. Proton & all. Un outil pour observer l'organisation d'une société d'agents; le cas d'une société d'agents chasseurs agriculteurs. *JFIADSMA'97, J. Quiqueton, M.C. Thomas et B. Trousse Eds, Hermés*, 1997