JASSS

Minh Nguyen-Duc and Alexis Drogoul (2007)

Using Computational Agents to Design Participatory Social Simulations

Journal of Artificial Societies and Social Simulation vol. 10, no. 4 5
<http://jasss.soc.surrey.ac.uk/10/4/5.html>

For information about citing this article, click here

Received: 02-Dec-2006 Accepted: 01-Feb-2007 Published: 31-Oct-2007

Abstract

In social science, the role of stakeholders is increasing in the development and use of simulation models. Their participation in the design of agent-based models (ABMs) has widely been considered as an efficient solution to the validation of this particular type of model. Traditionally, "agents" (as basic model elements) have not been concerned with stakeholders directly but via designers or role-playing games (RPGs). In this paper, we intend to bridge this gap by introducing computational or software agents, implemented from an initial ABM, into a new kind of RPG, mediated by computers, so that these agents can interact with stakeholders. This interaction can help not only to elicit stakeholders' informal knowledge or unpredicted behaviours, but also to control stakeholders' focus during the games. We therefore formalize a general participatory design method using software agents, and illustrate it by describing our experience in a project aimed at developing agent-based social simulations in the field of air traffic management.

Keywords:
Participatory Social Simulations, Agent-Based Social Simulations, Computational Agents, Role-Playing Games, Artificial Maieutics, User-Centered Design

How to design participatory social simulations?

Participatory social simulations (PSSs)

Goals of participatory social simulations

1.1

One could recognize that PSSs generally aim:

- at improving social simulation models (by involving stakeholders in the simulations), and
- at allowing stakeholders to explore different scenarios of their collective activities.

1.2

Social scientists have recently adopted PSS methods (Reitsma et al. 1996; Bousquet et al. 1999; Purnomo et al. 2002; Rothman et al. 2003; d'Aquino et al. 2003; Barreteau 2003; Etienne et al. 2003; Ramanath and Gilbert 2004; Nguyen-Duc 2005; Dray et al. 2006), among which d'Aquino and colleagues' work, particularly their use of role-playing games (RPGs) to support a local decision-making process for land use management, has caught our interests. Another method using computer-mediated RPGs introduced by Guyot and Honiden (2006) has also inspired our work. The main difference is that we employ such games for a formalized participatory design process (see Section 4). General properties of these methods are discussed later, from 1.5.

Example of PSS application

1.3

The usefulness of PSS can be shown by the application to air traffic management (ATM), based on which we have built up and evaluated our design approach. The *European Organisation for the Safety of Air Navigation* (EuroControl) wants to change the working procedures used by air traffic controllers and managers, which currently lack reactivity. It is obviously impossible to test such procedures in real traffic conditions, the use of simulations is thus indispensable. The problem is that most of the behaviours exhibited by the controllers and managers are implicit and only used in situation, i.e., they are usually not explicitly described in interviews, and not easily detected by observation. For example, it is difficult for a controller to explain what she/he has to do to delay an aircraft, reduce its speed or lengthen its trajectory, because this decision depends on the traffic condition, even on the status of each aircraft flying near it. The only way to get feedback concerning the feasibility of these procedures and their impact on the efficiency of the controllers' work is to place the controllers in a simulated work environment, where they interact together in simulated scenarios.

1.4

However, it is often difficult to mobilize controllers or managers for a long time. This means that only a limited set of scenarios can be explored if we merely rely upon stakeholders. Therefore, there exists a need to replace humans with realistic and believable agents. But how to build them given that most of the knowledge is unformulated? This is the central point of PSS design, where we advocate the use of a participatory process like the one described above in order to capture this knowledge by making people interact with assistant agents (described in 3.11). More precisely, each controller or manager is assisted by a software agent when she/he plays her/his role in a simulated traffic management system. The power of the agent in this particular interaction process is that it can not only show the stakeholder how to manage air traffic in such an artificial system, but also propose her/him decisions to make according to traffic conditions and explain why these decisions should be made. In this way, the stakeholder understands better the underlying simulation model, and so has more chances to improve it. We present this specialized PSS in detail in Section 3.

PSS design specificities

1.5

PSS design is then described as a participatory approach to involving stakeholders, through the use of artificial intelligence techniques, in the development of agent-based social simulation models from the earliest specification stages. It should be noticed that a PSS design process is not necessarily separated from the use of the simulation models to be built. Due to the diversity of domains invoked in this approach, and the number of techniques that can be used, it naturally raises several challenges. We discuss them in this sub-section and show how they can be addressed.

How to involve stakeholders in a simulation?

Stakeholders' understanding of the simulation

1.6

An accurate model of stakeholders' behaviours cannot be obtained unless they understand (and can reason about) the behaviours played in the simulation. It is the only way they will be able to discuss issues with designers.

1.7

Important design difficulties result from the difference between the stakeholders' and designers' visions for the social phenomenon to be simulated. The misunderstanding between them is typically multiplied in PSS design when the stakeholders repeatedly communicate with the designers during participatory sessions. Indeed, they work daily in very different conceptual worlds specific to their own professions. Although the stakeholders and designers must have at least a common vision for the final simulation results, the major difficulty appears as soon as they try to communicate with each other in the design process, at the moment when each one possesses her/his own incomplete model of the problem to be simulated. Hence, an important mission not always achieved by the designers during participatory sessions is to make the stakeholders adequately understand partial simulation models that remain to be completed.

1.8

Agent-based models (ABMs) are a particular type of simulation model which explicitly represents social behaviours of individuals or small groups, and so gives stakeholders a possibility to access to their simulated activities. In this case, it is the designers who present the content of the ABMs to the stakeholders. However, the ABMs in their own formal forms are still difficult to understand. Barreteau and Bousquet (2001) justified the need for opening the black box of the ABMs, and in our opinion, the essential reason comes from the nature of the ABMs themselves. Our modelling work in the field of air traffic management has shown that the agent-based concepts can cause ignorant misunderstandings. The concept of "agent" is already in itself ambiguous: it is comprehensible for air traffic controllers and managers, but can lend itself to several interpretations (because many fields use this term). The computer science concepts it covers are not obvious to the stakeholders either: for example, an "action" of an agent implies a concept of "state change" of which the stakeholders are unaware.

1.9

Researchers therefore seek to provide tools to explain the content of ABMs, e.g., role-playing games (RPGs) (Barreteau and Bousquet 2001). We can consider an RPG as *group settings that determine the roles or behavioural patterns of players as well as an imaginary context* (Barreteau 2003), or more generally, such a game can be seen as *the performance of a roughly defined situation that involves people with given roles* (Mucchielli 1983).

Traditional role-playing games

1.10

In PSS design, many types of RPG have been considered. Barreteau and Bousquet (2001) indeed discussed the use of RPGs to help stakeholders understand ABMs and gave a general guide on how to convert objects, agents and rules included in an ABM into game supports and players. On the other hand, we would like to categorize RPGs encountered in science (not only in PSS design) by the purpose of their use: training, social learning or discussion support tools.

1.11

The first type of use, i.e., training, is predominant. It aims at placing players in simulated situations in order to train them to react to specific conditions or to become familiarized with the other players. The game described in (Burton 1989) is an example in the field of irrigation management. Players play the roles of persons in charge in the irrigation department, village water directors or farmers. The objective is to improve the awareness of the interpersonal communication's importance in irrigation management.

1.12

Research on collective decision support systems has recently been oriented towards social learning, with interactive methods, requiring specific tools and platforms. These tools enable learning by interactions and dialogues. In the Stratagène game (Aubert et al. 2002), aimed at facilitating negotiations between stakeholders in phytogenetic resource management, scientists acquire knowledge on management processes by playing games together with stakeholders, in other words, while interacting with stakeholders in simulated situations. Such games characterize group settings by means of real-world situations. For all the players, i.e., scientists as well as stakeholders, learning then concerns interactions with each other.

1.13

Certain RPGs have been used as discussion support tools, because they provide a representation or simulation of collective rules, e.g., those for natural resource management (Bousquet et al. 1999). Particularly, the most important requirement for negotiation support systems is the legitimacy of content in negotiation processes. A significant number of failures indeed come from the fact that the model used in the negotiation is not shared by all the negotiators (Reitsma et al. 1996). In this context, RPGs offer an additional communication channel between scientists and stakeholders, which is, furthermore, controllable and open. By bringing a common knowledge, they can thus prevent conflicts due to misunderstandings or misrepresentations.

1.14

The three ways to use RPGs presented above could all be applied to PSS design. While exploring scenarios modelled by using ABMs, the reliability of the models' evaluation depends on the stakeholders' familiarity with the models. Some RPGs dedicated to training stakeholders could thus be helpful to improve of their awareness of the models. Also, social learning could be significant when researchers seek to have new group procedures evaluated by stakeholders. In this case, these evaluators learn how to interact with each other in imagined situations, while the designers acquire more knowledge about stakeholders' social behaviours. Finally, thanks to the capacity to provide an intuitive representation of ABMs, comfortable for both designers and stakeholders, certain RPGs could play the role of "translator" between these two.

Computer-mediated role-playing games

1.15

The majority of RPGs used until now in science are not in the form of software tools, i.e., charts, tables, boards and the like have mainly been employed as game supports. A special case we have found in the literature is a project in which Rothman and colleagues (2003) applied computer games to sustainable development research. On the other hand, in the field of video games, people are already familiar with distributed games, either on local networks or on the Internet. We argue that distributed computer-mediated RPGs can not only give players much more game configurations, but also allow the use of software agents that, as we will show later, can bring significant advantages to PSS design. Moreover, with small adjustments, agent-based simulation models can be reused in such games. Very recently, Guyot and Honiden (2006) introduced a similar idea by considering computer-mediated RPGs as the result of merging multi-agent systems and traditional RPGs. Our approach is not totally separated from this work, but adds to RPGs powerful tools which help efficiently elicit stakeholders' knowledge.

How to explore life-like scenarios?

Stakeholders' focus on scenario exploration

1.16

Any participatory project repeatedly explores operational scenarios built from typical real-world situations. A question raised here is whether simulation model's designers can obtain as much knowledge as they want to extract from all the information provided by stakeholders.

1.17

We distinguish two traditional purposes for exploring scenarios: validating the model itself, and exploring and validating different types of organizational structure or collective decision-making. Certain projects (Costanza and Ruth 1998; Van Asselt et al. 2001; Doran 2001; Ramanath and Gilbert 2004) explicitly use scenarios to test simulation models. Even in those whose objective is to test decision-making procedures (Abel 1999; Barreteau and Bousquet 2001; Etienne et al. 2003; Nguyen-Duc et *al*. 2004), an implied process of model validation by stakeholders always exists. In d'Aquino et al. (2003), for instance, stakeholders in the land use planning *imagine together solutions that do not somehow punish the "bad conduct" of some stakeholders, but instead increase everybody's satisfaction*. These solutions are transformed into two types of scenarios: one that is playable in RPGs, and the other which can be executed by computer simulations. The stakeholders can then not only be immersed in future situations imagined by themselves, but also observe simulation results of the same solutions. This transformation is, however, carried out by designers; the stakeholders are thus responsible for validating its faithfulness, or in other words, for confirming that the scenarios correctly represent the solutions. It should be noticed that the model validation is normally done on the same platform and in the same environment as the solution testing. The only difference is the purpose on which stakeholders focus while playing games or observing simulation results.

1.18

However, these purposes are not always achieved because a simulated world is far different from the real one. Indeed, accompanying problems, concerned with the artificial environment where stakeholders are immersed, strongly affect the success of participatory processes. In such an environment, stakeholders and designers discuss scenarios while either observing a computer simulation (Costanza and Ruth 1998; Abel 1999; Doran 2001), or playing RPGs which intuitively represent an underlying ABM (Barreteau and Bousquet 2001; d'Aquino et al. 2003; Etienne et al. 2003). The main problem is that of the stakeholders' focus on purposes for exploring scenarios. It can cause the ineffectiveness of working sessions. For example, during an RPG, a new stakeholder, who is not yet familiar with the game rules, has the potential to discover the game, neglect the professional task related to the scenario in question and play without using her/his expert knowledge. The same phenomenon occurred in our project (presented in Section 3) when we presented our computer games to the air traffic managers who participated in the design process for the first time. We have also seen many other unexpected effects arising when stakeholders work with scenarios during participatory sessions.

1.19

We have found methodological solutions to such accompanying problems in another field, i.e., *User-Centered Design* or *Participatory Design* (Schuler and Namioka 1993), a sub-field of software engineering.

User-Centered Design (UCD)

1.20

It is obvious that any computer social simulation is software running on a computer or computer network, whose users are both researchers and stakeholders. Hence, the way simulations are designed can be inspired by methods used in the field of software (or systems) engineering, or more particularly in UCD (Schuler and Namioka 1993).

1.21 UCD is a research field that has evolved since the late 80s. Its principle is to involve users in the design process from the earliest stage, but the objective is not to design social simulations but software systems. One of its important premises was the *Trade Union* perspective (Nygaard and Bergo 1975; Ehn and Sandberg 1983) concerned with developing strategies that enable workers and their unions to influence the development of new technologies in their workplace. We recognize that, in both UCD and PSS design, it is important for designers to ensure the quality of knowledge elicited from software users or social stakeholders during each participatory session.

1.22 As shown in Ramanath and Gilbert (2004), UCD-related techniques can ease the design of a large number of scenarios by reducing the biases introduced in their input data (e.g., adjusting, anchoring, availability, data saturation, spurious cues, etc.) while maintaining the participants' motivation (because stakeholders can gradually lose interest during repeated sessions). The authors propose to apply some UCD basic steps, i.e., *Scenario Analysis*, *Future Workshop* (FW) and *Prototyping*, to the modelling of the *European Digital Content Market*. Their use of FW as a brainstorming technique to uncover problems and formulate visions is really a novelty in PSS design. This technique is particularly helpful in the first design stages when simulation models are not yet available.

1.23 We intend to go further than that by applying concrete methodological solutions proposed in Grønbæk (1991) to the accompanying problems described above. This approach called *Cooperative Prototyping* (CP) is centred on the use of prototyping techniques in cooperative activities among users and designers ("prototyping" here has its usual meaning, i.e., using software prototypes). Grønbæk (1991) revealed a central element of the prototyping sessions' installation: the selection of professional tasks to be performed in each session. These tasks, which frame the users' evaluations in an open way, are called "frame tasks." They are selected by users and designers in cooperation and are formulated in the users' terminology. By analyzing the frame tasks used in the studies described in Bødker and Grønbæk (1991b) and Trigg et al. (1990), Grønbæk indicates that certain tasks require facilities very different from those envisaged by the designers.

1.24 Prototyping sessions need to be carefully prepared. It is particularly important to clarify the purpose of each session. According to Grønbæk, if the evaluations are arbitrary, the prototypes will not improve the interaction between designers and users sufficiently to avoid false ideas about users' needs (Bødker and Grønbæk 1991a). He distinguishes the purpose of generating ideas about desirable facilities and that of evaluating the adequacy of a suggested solution. Designers are advised to prepare all session purpose specifications before each work day with users or at the time of organising a series of sessions.

1.25 Especially, software agents can help introduce such methodological solutions in each participating session if computer-mediated RPGs are employed (see 4.9).

Eliciting stakeholders' knowledge

Stakeholders' self-modelling

1.26 Modelling stakeholders' social behaviours is in fact a process of eliciting human beings' knowledge. In real-world situations, even while working within well-defined procedures (e.g., in air traffic management), stakeholders frequently use informal knowledge based on their personal experience and behave in unpredicted ways. This kind of behaviour is not easy for stakeholders to describe and present to designers. Philosophers and psychologists (Kihlstrom 1993; Cleeremans 2001) have also shown that people are not always conscious of their behaviours, and that it is often difficult for them to formalize the knowledge they use to take actions. Skilled reading provides an example: *we recognize certain patterns of marks on the printed page as letters, and certain patterns of letters as words, and decode the meanings of words in light of the words around them, but we rarely have any conscious awareness of the rules by which we do so*. In this context, the rediscovery of the unconscious is necessary.

1.27 We distinguish three phases of knowledge elicitation in PSS design. At first, when simulation models do not exist yet, the designers' mission is to build some pertinent model of stakeholders' behaviours. They try to acquire knowledge about basic behaviours, particularly those related to the problem to be simulated. Because of the absence of models, some brainstorming exercises (Kensing and Madsen 1991; Bødker et al. 1993; Ihlström 2002), can be applied in order to organize designer and stakeholder groups working together on how stakeholders behave in supposed situations. Secondly, once the designers obtain a model executable by computer simulations, they execute multiple scenarios in participatory sessions in such a way that stakeholders can use their knowledge about their own behaviours to improve this model. Finally, when the models become pertinent as much as necessary, they will be used to test new group behaviours or procedures. In this third step, new theories of social behaviours can be examined and refined by stakeholders during participatory sessions. The stakeholders' self-modelling is involved in all three steps because they are repeatedly asked to describe and explain their actions in particular situations.

1.28

We thus identify a need for tools to elicit knowledge usable in PSS design. According to Hamel (2005), Dray (2006) and colleagues, research on knowledge engineering (Schreiber et al. 1999) provides efficient elicitation techniques, e.g., *Laddering* (Reynolds and Gutman 1984) and *Thinking Aloud* (Conrad et al. 1999). Alternatively, we are interested in maieutic processes (Sempé el al. 2005), which directly support the stakeholders' self-modelling.

Artificial maieutics

1.30

Socrates, a Greek philosopher, introduced a method for eliciting knowledge called "maieutic method" based on the in-situ self-questioning, which, by definition, has five main characteristics (according to Samples 1998):

- It is skeptical. *It begins with Socrates' real or professed ignorance of the truth of the matter under discussion.*
- It is conversational. *It employs the dialogue not only as a didactic device, but as a technique for the actual discovery of opinions amongst men (there are truths upon which all men can agree). Socrates proceeds to unfold such truths by discussion or by question and answer.*
- It is conceptual or definitional *in that it sets the acquisition of concepts as the goal of knowledge. Socrates tacitly assumes that truth is embodied in correct definition.*
- It is empirical or inductive *in that the proposed definitions are criticized by reference to particular instances. Socrates always tested definitions by recourse to common experience and general usages.*
- It is deductive *in that a given definition is tested by drawing out its implications, by deducing its consequences.*

1.31

This method is an intellectual foundation of what scientists call "critical thinking" (CT) (Bowell and Kemp 2001). When people self-consciously reflect on what they are thinking about, deliberately examine the issues involved and ask why and discuss reasons with each other, they are thinking critically. In other words, CT is a way of thinking that enables people to reason with studied deliberation. They concentrate and reflect on their own reasoning. They intentionally focus their mind on a subject.

1.32

In fact, the main objective of the maieutic or Socratic method is to use question-answer dialogues, called "maieutic dialogues," to help people think for themselves. This kind of dialogue has largely been applied in education. An existent *Socratic Method Support System* for teaching law (Sakurai and Yoshino 2004) provides software tools allowing teachers to prepare maieutic dialogues and graduate students to exercise these dialogues in class as well as on their own.

1.33

Another architecture for implementing maieutic dialogues (Weusijana et al. 2002) aims at fostering deeper student reflections on well-defined tasks. The authors of this architecture, named SASK (*Socratic Ask*), have built a program that employs Socratic questioning to improve students' performance and learning in an undergraduate biomedical engineering school. A provided conversation panel is the key interface of the program. By using this panel, students can answer questions such as "*Do you think you will see a relationship between Ultra-filtration (U) and flow rates?*" or "*What controls U or flow rates?*" The interface does not intend to constrain what students say, but a free-text answer is too difficult to process. Therefore, some buttons and templates are used. For example, simple answers with no variable content, e.g., "*We're still thinking,*" are buttons. More complex answers with variable content are buttons that bring up structured templates similar to web forms.

1.34

Sempé and colleagues (2005) recently introduced a new concept of "artificial maieutics" that uses *Artificial Intelligence* tools, or more particularly software agents, to support a maieutic process of knowledge elicitation. The idea is that an agent (assistant agent) interacts with a stakeholder through a computer screen, questions her/him and tests her/his reactions, either directly, e.g., "*Why such an* *action?*," or indirectly, e.g., through a modification of the perceptions available, in order to explore her/his informal knowledge. This technique, because it appears able to capture this knowledge in situation, and because it can easily be implemented within a computer-mediated RPG, seems to be appropriate for PSS design.

Our design approach

2.1

We have shown that important challenges in PSS design can be overcome by using techniques from RPGs, UCD and artificial maieutics. We propose in the next section a solution based on the concurrent use of these techniques to the design of a participatory simulation for the problem of ATM previously mentioned. Then, in Section 4, we generalize this solution to a PSS design method. In comparison to Guyot and Honiden's work (2006) using a similar type of RPG (mediated by computers), our approach stresses more design aspects by applying off-line and one-line maieutic processes, and by employing accompanying agents (see 4.9) which help guide stakeholders during participatory sessions.

Application to air traffic management

3.1

In this project, we initially design a multi-agent simulation in the traditional way, i.e., based on interviews, and then transform it into a participatory design environment. Concerning the terminology, we employ from now on a new concept, i.e., "social actor" or "actor," equivalent to "stakeholder."

The initial simulation

Air traffic management (ATM)

3.2

Current ATM system is airspace-based. The airspace is divided into several sectors, the size of which depends on the number of aircraft in the region and the geometry of air routes. There are usually two air traffic controllers to handle the traffic in each air sector: a planning controller and an executing controller. The planning controller works at a strategic level to minimize the number of conflicts or their complexities. The executing controller works at a tactical level to ensure that there are no conflicts, i.e., infringements of standard separation between aircraft, by giving instructions to the pilots.

3.3

ATM additionally comprises a higher level of management: traffic flow management. The mission of flow managers is to (re-)plan flights at the multi-sector level with the main objective to avoid traffic congestion and the controllers' overwork load (due to a large number of aircraft to be controlled).

Goals of the simulation

3.4

Research undertaken by EuroControl (*European Organisation for the Safety of Air Navigation*) on collaborative working procedures needs simulation tools able to validate new team-working methods. We thus implement a multi-agent simulator, named MadFam, which can simultaneously complete the three following objectives:

- evaluate new procedures of collective work defined within the framework of the FAM project (Stoltz and Ky 2001) by using basic team-work criteria (optimization time, message transmission number, collective failure rate, etc.),
- perform a set of demonstrations of these new procedures bound for ATM actors (air traffic controllers and flow managers), and
- provide an experimental framework that allows integrating human participants (air traffic controllers and flow managers) into the simulator's execution loop in order to validate the new procedures.

Air traffic simulation - eDEP

3.5

Our simulator is based on the EuroControl's early demonstration and evaluation platform (eDEP), implemented in Java (eDEP 2006), that provides a set of standard ATM components, e.g.:

- *airspace*: provides a database of static airspace information;

- *integrated air surveillance* (IAS): provides a database of surveillance radar tracks;
- *flight path monitor*: uses track data produced by the IAS to monitor the progress of a flight according to its planned path through the airspace;
- *initial flight plan*: reads a traffic sample from an input file and generates an initial plan that defines route constraint points and altitude limits;
- *trajectory predictor*: provides a trajectory prediction algorithm which uses aircrafts' kinematic models to predict the real motion of a particular aircraft.

ATM simulation - MadFam

3.6

A major concern in leaving some loose ends in ATM rules is the occurrence of unpredicted traffic peaks at the entry of a congested area. A way to solve the problem is to structure and organise the arrival flows in real-time. The FAM (*Future Air Traffic Flow Management Measures*) project (Stoltz and Ky 2001) carried out by EuroControl defines a procedure of collective work aimed at synchronizing air traffic in real-time. Initially, the project only stresses the first sub-procedure, i.e., the establishment of a common synchronization measure, which only concerns flow managers (Guerreau 2002):

- the local flow manager (LFM), responsible for the zone containing the congested sector (called the requestor LFM) after identifying a risk of traffic overload, claims a traffic synchronization measure in order to remove the risk;
- he sends a request for synchronization to the central flow manager (CFM) and to all the LFMs (called the supplier LFMs) responsible for the zones crossed by the arrival flows that reach the congested sector;
- the establishment of the synchronization measure between the actors in contact then begins; different synchronization measures can be proposed and are jointly examined; their feasibility and effectiveness are evaluated from the different points of view of the actors; the requestor LFM estimates whether they solve her/his congestion problem; the supplier LFMs evaluate the difficulty in implementing the proposed measures (e.g., in terms of controller workload); the CFM examines the side effects (network effects, conflicts with a measure previously applied, etc.);
- the process finishes with a collective agreement on the synchronization measure to be applied; a validation is required for all actors.

3.7

To assess the applicability of such procedures of collective work, we develop MadFam (*Multi-Agent Demonstrator for FAM*), in which social actors (air traffic controllers and flow managers) and software agents work together like team-mates. We model the interaction between actors and agents by using STEAM (*Shell for Teamwork*), a generic teamwork model described in Tambe (1997) and Pynadath and Tambe (2003). To enable the integration of social actors, the simulator is implemented in the form of distributed software running on several computers. Each human participant is provided with a user interface that simulates her/his real working position, i.e., an air traffic controller's or flow manager's working position.

Observations and problems

3.8

Although the human-in-the-loop experiments are not new in ATM research, MadFam represents a particular kind of simulation, in which social actors and software agents play similar roles. The initial purpose of the actors' participation is to validate the feasibility of new collective working methods. On the other hand, the use of agents aims at reducing the required number of professional actors. Indeed, at EuroControl, it is always difficult to have enough participants because there are only four or five permanent flow managers for all human-in-the-loop experiments.

3.9

First of all, however, the agents' behaviours must be realistic enough because they affect the validation of new working procedures. While discussing these behaviours with ATM specialists, we faced many difficulties related to the explanation of their content formalized in agent-based terms. These difficulties justify a need for new methods of behaviour design, and also suggest the use of RPGs, which can help open the back box of ABMs. In addition, the actors' participation gives us the idea to implement the games on MadFam itself. Consequently, we apply and test a particular PSS design approach, as presented from 3.10.

Participation of traffic flow managers

Properties of the participatory approach

3.10
From the PSS designers' point of view, we use MadFam:

- to simulate a new traffic management approach by using agent-based simulation models and software agents,
- to develop a simulation environment which allows the participation of social actors,
- to add functionalities to this simulator in order to transform it into a tool for the participatory design of the agents' behaviours, and
- to formalize methodological solutions to guide social actors in this process.

3.11
Actually, the actors who participate in human-in-the-loop experiments can help design the behaviours of the software agents used in the experiments. MadFam is thus employed to set up an environment in which social actors play RPGs (more detailed from 3.14 to 3.16). We apply maieutic processes by using a particular type of software agents, called assistant agents, each of which assists an actor.

3.12
For experimental validation, we stress significant properties to be verified. These properties relate not only to the MadFam's software components but also to the interest of social actors for RPGs and to the way in which they use MadFam to play the games. We distinguish three categories: user interface properties, actor interest properties, and properties of impact on the agents.

3.13
With the aim to refine the agents' behaviours, we collect "logs of actor behaviours" generated by assistant agents during their dialogues with social actors (see an example shown in Figure 1). Since we (the designers) will use these logs to update behavioural rules for the agents, a question arises here: which are the qualities of a set of logs that can make a true improvement? The results described in 3.24 and 3.25 give an answer.

Participatory design

3.14
We have already implemented MadFam in the form of a distributed human-in-the-loop simulation, which allows several social actors (flow managers) to participate together in new working procedures. With small changes, we transform it into a platform for playing distributed RPGs (as described in 4.10 and 4.11), in which the actors not only strictly follow fixed scenarios but also play games, i.e., make choices according to their perception of the traffic condition. Based on the knowledge of the flow managers' behaviours that we acquired during interviews, we create software agents to be introduced into the simulation. These agents (artificial actors) play games together with social actors. The purpose is to have the former's behaviours validated/corrected by the latter through the repeated exploration of game scenarios.

3.15
The ability of actors and agents to directly interact with each other during participatory sessions enables us to perform maieutic design processes. To extract knowledge used in situation by actors, we introduce a type of maieutic dialogue (illustrated in 4.21 and 4.22) between an actor and her/his assistant agent. In fact, this particular kind of agent has the same decision-making mechanism as any artificial actor, but it is never an independent player in games and is always dedicated to a social actor (or "tutor"). During dialogues with its tutor, each assistant agent produces "logs of actor behaviours," based on which we update decision-making rules for the agents.

3.16
Concerning accompanying problems discussed from 1.16 to 1.19, we recognize that while playing games social actors have the tendency not to focus on the purpose for exploring scenarios, but on the software itself by trying out play possibilities without using their expert knowledge to validate the underlying agent model. We then apply two methodological solutions from *Cooperative Prototyping*, i.e., "session purpose" and "frame task" (illustrated from 4.3 to 4.9), in order to concentrate their attention on what we want to obtain in each design session.

Experimental setup

3.17
The RPG used in the experiments is based on the procedure of collective work described in 3.6, which only concerns flow managers. The number of players is fixed: four local flow managers (actors or agents) and a central flow manager (an agent).

3.18

We always use four connected computers:

- A presentation computer (PC) that displays aircraft positions, and message exchanges between social actors and software agents;
- A control computer (CC) used by the monitor (among the designers) to supervise the execution of the simulator;
- Two game terminals (GT), each of which provides an actor with a graphic user interface equipped with the tools necessary to play the game.

3.19

An experiment day (or half-day) normally starts with two demonstration sessions in which the actors involved (flow managers) are only observers. In the first session, only the artificial players (agents) play the game, and in the second, the monitor plays with the agents. The actors can subsequently participate in some trial sessions in order to get used to the game and user interface. Afterwards they are really integrated into a predetermined series of sessions with a time interval envisaged between two consecutive sessions. These inter-session periods allow the actors to discuss with the monitor or to quickly answer specific questionnaires. The experiment day ends with a conclusion meeting.

```
#SatOct18135824ICT2003

--Predicted Situation:
sector-LFESE-load(11)-capacity(15);
sector-LFPMOP-load(7)-capacity(15);
sector-LFFUJ1-load(10)-capacity(15);
sector-LFLLTMA-load(6)-capacity(15);
sector-LFEUF1-load(13)-capacity(15);
sector-LSAGFOW-load(9)-capacity(15);
sector-LFBTHU-load(12)-capacity(15);
sector-LIRRMIE-load(10)-capacity(15);

avion-MSK20N-delay(60);...

LFM-LFMM-availability(YES);...

--Suggested action type: Modify

--Suggested measure: LFM_A_LSAZ
flight: MSK20N
replicate from RBT to LFMN
way points: RBT PTV NEV *2MTL MTL GIROL *MEDO *5MTL AMFOU ALBET ARMUS SINRA DRAMO LFMN

--Because of:
controller-LFEUF1-load(13)-capacity(15);
controller-LFFUJ1-load(10)-capacity(15)

--Reason: The controllers of LFEUF1 will be very loaded and the controllers of LFFUJ1
will be less loaded

--Updated action type: Modify

--Updated measure: LFM_A_LSAZ
flight: MSK20N
replicate from RBT to LFMN
way points: RBT PTV NEV *2MTL MTL GIROL *MEDO *5MTL AMFOU ARMUS DRAMO LFMN

--Because of: aircraft-MSK20N

--Reason: The MSK20N aircraft is delayed too much (for an hour) compared to its initial
flight plan and its itinerary cannot be lengthened
```

Figure 1. Log of a dialogue between an assistant agent and a flow manager

3.20

We organized the first two experiments at EuroControl to test not only the MadFam's user interfaces but also the interest of actors. The participant group was composed of a monitor using the PC and two actor pairs, each of which used a GT. The three following experiments aimed to collect the "logs of actor behaviours." This time, it was not a pair but only one actor who used a GT. Each session was divided into sub-sessions, each of which corresponded to a concrete activity defined by a frame task and a sub-session purpose (see 4.3-4.9). The number of sessions depended on the number of prepared traffic scenarios and also on the number of flow managers involved. In order to vary the game in different sessions, we prepared many traffic scenarios, exchanged the actors' roles and changed the actors themselves.

Experiment examples

3.21

We parameterized 7 out of 9 elements of a log, i.e., "`predicted situation`," "`suggested action type`," "`suggested measure`," "`because of`," "`updated action type`," "`updated measure`" and "`because of`" (see

Figure 1), in order to quantitatively evaluate a set of logs. This enabled us to test the following assumptions:

- Im1: the entities perceived by the actors ("`because of`") in the same "`predicted situation`" were not too varied;
- Im2: the diversity of the entities perceived by the actors in the same "`predicted situation`" decreased each time the designers modified the behavioural rule base for the agents;
- Im3: the actions decided by the actors ("`updated measure`") in the same "`predicted situation`" were not too varied;
- Im4: the diversity of the actions decided by the actors in the same "`predicted situation`" decreased each time the designers modified the behavioural rule base for the agents.

Table 1: Experiment sessions for collecting "logs of actor behaviours."

Session	Peak hour	Aircraft to handle	Predicted situation (corresponding to a combined synchronization solution)	Actor group
1	09:41:40 – 25/06/2003	DLH4260, BER2576, MSK20N	FAMC_YELLOW_1, FAMC_ORANGE_2_1, FAMC_RED_1_10	No. 1
2	09:41:40 – 25/06/2003	DLH4260, BER2576, MSK20N	FAMC_YELLOW_1, FAMC_ORANGE_2_1, FAMC_RED_1_10	No. 2
3	09:41:40 – 25/06/2003	DLH4260, BER2576, MSK20N	FAMC_YELLOW_1, FAMC_ORANGE_2_1, FAMC_RED_1_10	No. 3
4	09:41:40 – 25/06/2003	AFR1205, AZA365, TRA1493	FAMC_BLUE_1_1, FAMC_RED_1_8, FAMC_ORANGE_2_2	No. 1
5	09:41:40 – 25/06/2003	AFR1205, AZA365, TRA1493	FAMC_BLUE_1_1, FAMC_RED_1_8, FAMC_ORANGE_2_2	No. 2
6	09:41:40 – 25/06/2003	AFR1205, AZA365, TRA1493	FAMC_BLUE_1_1, FAMC_RED_1_8, FAMC_ORANGE_2_2	No. 3
7	09:41:40 – 25/06/2003	HHI2252, CRL523, AZA365	FAMC_YELLOW_2, FAMC_BLUE_1_2, FAMC_RED_1_12	No. 1
8	09:41:40 – 25/06/2003	HHI2252, CRL523, AZA365	FAMC_YELLOW_2, FAMC_BLUE_1_2, FAMC_RED_1_12	No. 2
9	09:41:40 – 25/06/2003	HHI2252, CRL523, AZA365	FAMC_YELLOW_2, FAMC_BLUE_1_2, FAMC_RED_1_12	No. 3

3.22

We intended to obtain as many "logs of actor behaviours" as possible. By varying the game scenarios and actor groups, we could carry out up to 45 sessions organized within three 15-session days. During the two inter-day periods, we were based on the logs already obtained to modify the agents' knowledge base. In any session, we implemented for each actor three or four actor-agent dialogues (a particular type of maieutic dialogue), corresponding to the proposition of three or four solutions to handle aircraft, and thus saved three or four "logs of actor behaviours." The size of an actor group that participates in a session was always two (a "requestor LFM" and a "provider LFM"). We thus obtained more than 90 logs.

3.23

Table 1 summarizes the 9 first sessions of the first 15-session day, in which each of three actor groups confronted three "`predicted situations`." The only imposed condition was that different "`predicted situations`" must be distinct even if the same aircraft could be present in several sessions.

Results

Improvements in the agents' behaviours

3.24

The stabilization and harmonization of the knowledge used by the actors in the experiments are essential because not only do they ensure the coherence of the whole of the behavioural rules deduced from the logs but they also confirm that the modification of the agents' rule base by the designers increases the quality of the actors' games.

3.25

Therefore, we calculate, on all the set of logs collected, the diversity of the actors' perceptions and decisions in the same "`predicted situation`." As shown in Figure 2, the majority of the diversities do not exceed the standard diversities, and the average diversities decrease after each time we modify the agents' rule base according to the previously collected logs. These results show that the "logs of actor behaviours" (considered as a type of structured knowledge extracted in direct interaction with actors) appear useful for the improvement of the agents' behaviours. However, this validation only deals with the parameterizable elements of a log, i.e., it does not yet concern the abstract "`reasons`".

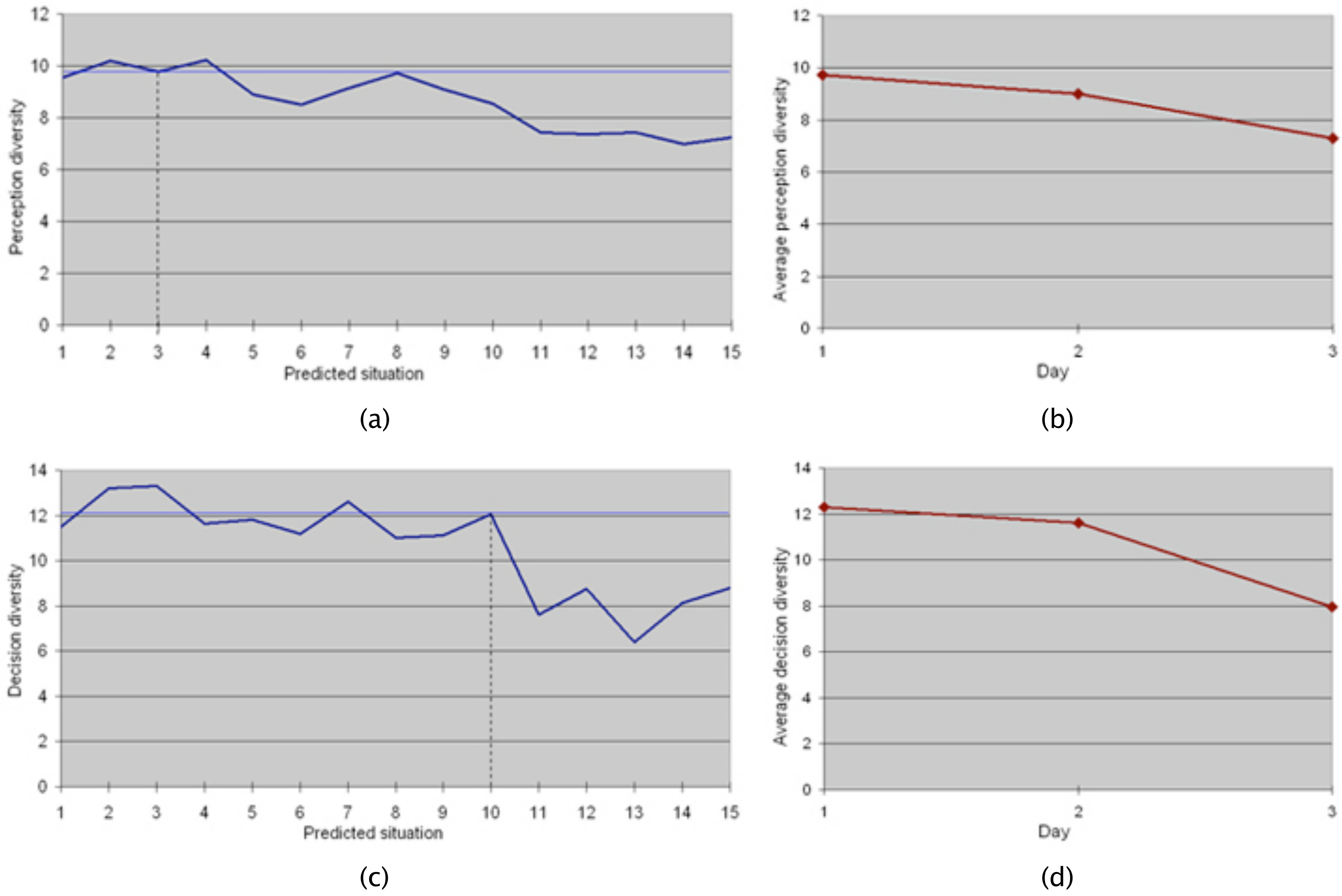


Figure 2. Quantitative results. (a) Diversity of the actors' perceptions. (b) Average diversity of the actors' perceptions. (c) Diversity of the actors' decisions. (d) Average diversity of the actors' decisions

Interest of the traffic flow managers

3.26

We tested the interest of the participant actors for the game by using specific questionnaires which the actors filled in during the inter-session periods. These questionnaires could also give them ideas about their own interest for discussions during the conclusion meetings (see 3.17-3.20 for the experimental setup). Based on the information obtained, we formulate the following methodological recommendations:

3.27

Concerning professional interest:

- even when the actors are already at ease with the simulator's user interface, the quality of their decision making strongly depends on the reality of the tools provided; for example, it is better to add a tool which represents the predicted "vertical profile" (varying altitude) of an aircraft so that the flow manager can more easily determine a feasible "level change" (altitude change) for this aircraft, even if the existence of this tool is not essential;

- it is necessary to access to each actor category's own knowledge to make sure that the presentation of a domain-specific concept is correct; this also means that general knowledge about the application domain is not enough for the designers; for example, for flow managers, an aircraft only corresponds to a flight plan containing a list of "estimated hours of arrival" at identified points, whereas a controller is only interested in the 4-dimension position of an aircraft (i.e., its position in terms of latitude, longitude, altitude and time);
- the generality of the simulation game always brings unrealistic possibilities compared to the actors' professional practice; it is thus important to be able to decrease this generality by imposing professional constraints; for example, in practice, it is extremely rare that a flow manager modifies more than two beacons (identified points) on the flight path predefined for an aircraft.

3.28

Concerning personal interest:

- the variety of the game strongly depends on that of the game scenarios; this requires a hard work to prepare data before each experiment day;
- paradoxically, to obtain the necessary variety of the game, a certain degradation of the actors' professional practice can be useful; indeed, the more neglected the professional constraints, the more experiment scenarios can be created.

Discussions

3.29

To deal with accompanying problems (see 3.16), two methodological solutions from *Cooperative Prototyping* are employed in our design process: determination of the purpose of each RPG session, and use of the frame tasks. Although these solutions require new functionalities for the multi-agent simulator, they help guide the social actors' (or stakeholders') activities within the simulation environment. The designers must work with the actors before each experiment day to prepare frame tasks, and then concentrate the actors' attention on each task during RPGs. However, this careful experiment planning must leave the actors sufficiently free to play the game. For example, it is necessary that the frame tasks be defined in a sufficiently general manner.

3.30

In order to maintain the actors' personal interest throughout an experiment day, it is suggested that the game scenarios prepared for the day be sufficiently varied and interesting. On the other hand, we apply *Contextual Inquiry* (Holtzblatt and Jones 1993; Beyer and Holtzblatt 1997), a method of *Ethnographic Field Study* (Hughes et al. 1994), to the design of the simulator's user interface, in order to attract the actors' professional interest during RPGs. In fact, the interface includes simplified professional tools familiar to the actors.

Four-step PSS design

4.1

Starting from our participatory project in ATM, we formalize a general PSS design method based on software agents. We use examples from several recently published papers for illustrative purposes.

Accompanying methodology

4.2

As presented from 1.16 to 1.19, stakeholders' (social actors') activities during participatory sessions are strongly affected by accompanying problems concerned with the artificial environment in which they are immersed. It is therefore necessary to formalize an accompanying methodology to guide them during design sessions. This methodology is not an independent design step, but is applied to all the steps.

4.3

The first accompanying problem to be solved is how to help the actors focus on the tasks belonging to the scenario they are exploring. We currently apply solutions from *Cooperative Prototyping*, i.e., "session purpose" and "frame task."

4.4

One should distinguish the concrete session purposes given here and the general purposes for exploring scenarios discussed from 1.16 to 1.19. When software agents are not employed yet, a participatory session has one of the two following purposes:

- show the actors how to act, including how to make decisions,

- extract actors' decision-making procedures (see 4.12-4.15).

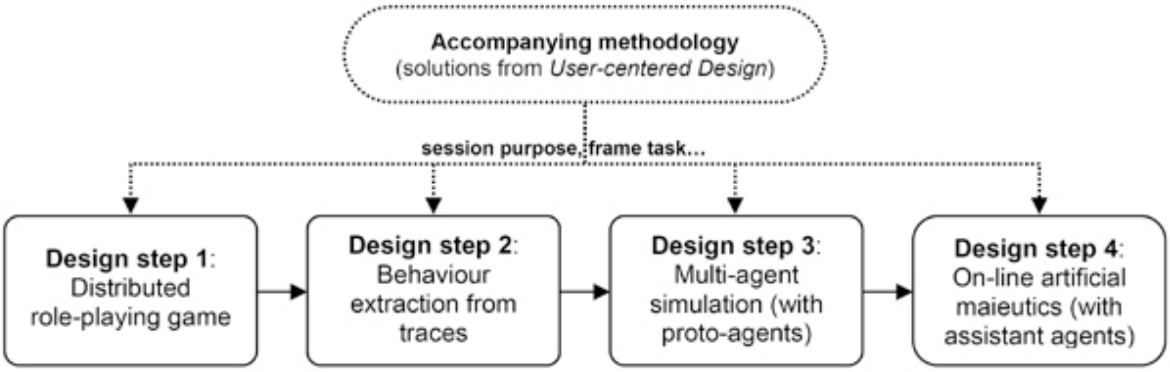


Figure 3. Four-step PSS design

4.5

The actors must be clearly aware of the session's purpose throughout the session. The indispensable role of the monitor (normally among the designers) is thus to adequately inform the actors of the purpose before each session, and to ensure their focus on it during the session. That also means she/he must be sensitive to any focus change in the actors, and remind them of the purpose without interrupting the session too much.

```
FRAMETASK PROPOSE_MEASURES
        BEGIN_EVENT             SUBMIT_TACTIC
        END_EVENT               PROPOSE_MEASURES
        PRESENTATION   "In this part, you are free to accept/refuse or modify each predefined
solution (measure) to be proposed, based on:\n - the flight plans of HHI2252,DLH4260,MSK20N and
TRA1493\n - the geographical information of concerned sectors and beacons\n - the predicted
number of aircraft situated in each concerned sector"
        HELP                    "In this part, you:\n - accept/refuse or modify each predefined
solution (measure) (\"Accept\"/\"Refuse\" or \"Modify\")\n - propose own measures to all
concerned traffic managers (\"Propose own measures\")"
        SUB_SESSION_PURPOSE     EVALUATING_SIMULATION
        TOKEN_OWNER             OWNER_03
END
```

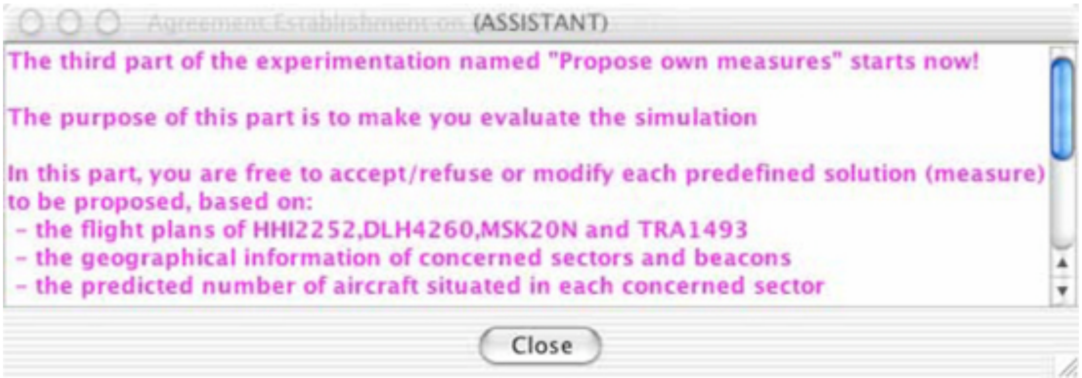


Figure 4. Examples of "frame task" and "(sub-)session purpose" used in an air traffic management simulation

4.6

The introduction of software agents (from the third design step) gives more facilities to guide the actors, but also requires more work in participatory sessions. In fact, two possible session purposes are added:

- present the agents' decision-making procedures,
- evaluate/improve the agents' decision-making procedures.

4.7

When we decompose an entire session into sub-sessions, the purpose can change from a sub-session to another, e.g., the beginning of a session seeks to show the first part of the game and the following aims at making the actors evaluate agents' decisions.

4.8

It is important to precise for each (sub-)session what to show, extract, present or evaluate, so that the actors understand it. A frame task is therefore used to frame all the demonstrations, presentations and evaluations during a (sub-)session. Such a task is defined in collaboration with the actors and described in an abstract way (see Figure 4). Before taking part in an entire session, the actors already know the frame tasks of all its sub-sessions. The data entity shown in Figure 4 represents the frame task of a sub-session; it contains the frame task's presentation (PRESENTATION) and the corresponding sub-session's purpose (SUB_SESSION_PURPOSE).

4.9
Introduced software agents can in addition contribute to the methodology by automating the frame-tasks' and (sub-)session purposes' communication. At the beginning of each sub-session, in order to inform the actors of the corresponding frame task, we make special agents known as accompanying agents give a message to each actor as illustrated in Figure 4. In fact, the scheduling server fires the sub-sessions' ending and beginning events. Once an accompanying agent captures such an event, e.g., SUBMIT_TACTIC, it waits until the actor, to whom it is dedicated, finishes all her/his actions corresponding to the current frame task before informing her/him of the new one.

Step 1: Distributed role playing games with actors

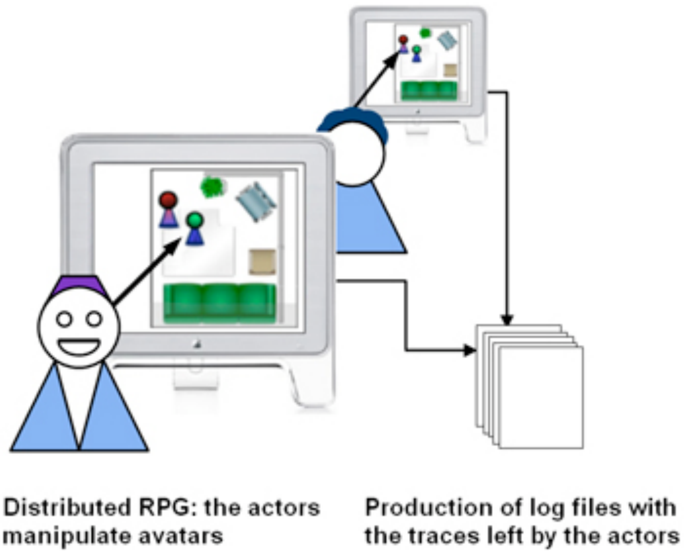


Figure 5. Distributed role-playing game

Figure 6. User interface of SimCafé, a game that simulates a coffee market (Guyot and Drogoul 2005)

4.10
The first design step is the development of a new kind of RPG. Identical to the ones used by Barreteau and colleagues (2001) or by d'Aquino and colleagues (2003), except that the game needs to be played on computers, able to log acts, speeches and interactions of all players, this kind enables the automatic processing of actors' traces. Actually, such an RPG runs on several connected computers each of which is dedicated to a player or actor, as illustrated in Figure 5.

4.11
A key element is the user interface, which provides an actor with not only all the means of playing the

game, but also a vision for the simulation model. Indeed, an important objective of the game is always to explain the content of the underlying ABM.

Step 2: Extracting knowledge from actors' interactions

4.12

Based on a distributed RPG, this step aims at identifying patterns of interaction and at extracting specific behaviours from the traces left by the actors. They usually complement the initial knowledge acquired by the designers using observation or interviews of the actors.

```
(guard
   (false
     (...))
   ((?receive_pergamino qty : $Int-X from : $Agent-Z)
     (do
       (!accept_offer contract :
         (if
         (?have_less_pergamino than :
           get_cereza_bags_count)
           (if
           (?transformation_process_completed)
           $Contract-1
           $Contract-2)
         (if
           true
           $Contract-2
           $Contract-2)))
     (do
       (!send_message to : $Agent-X)
       (do
         (!transform_cereza amount :
           (if
             true
             $Int-Y
             (get_budget)))
         (!send_message to : $Agent-Y)))))
   (false
     (...))
   (otherwise
     (go
       (if
       false
       3
       1))))
```

Figure 7. Example of extracted behaviour in the form of a *Q program* (Guyot 2003)

4.13

This extraction can be done manually (as it is the case in most of the social models built so far, see for example d'Aquino et al. 2003) by the designer, but it can also be done automatically with the help of data-mining programs, for instance, that can extract patterns, repeated behaviours, from the traces. A good example is the knowledge extraction in the SimCafé game (Guyot and Drogoul 2005), which:

- has been implemented and employed in the framework of a project funded by LAFMI, the French-Mexican Computer Science Laboratory, http://lafmi.imag.fr, aiming at developing an agent-oriented programming language to establish an e-commerce system applied to the Veracruz (Mexico) international coffee market;
- aims to provide a participatory environment for the validation and consolidation of the coffee market's model;
- is a game between coffee producers and buyers, in which producers exchange coffee and money and form coalitions to fulfil buyers' offers (a producer can produce coffee, accept offers made by buyers, send messages, share funds with other producers, and also be a buyer).

4.14

This game uses logging agents (a particular type of software agent) to log all actors' interactions, and then processes the logs in order to automatically generate interaction patterns written in *Q language*

(Ishida 2002), a subset of the Lisp language (see Figure 7). To build such *Q programs*, genetic programming is chosen. The fitness function simply consists in the ability of programs to generate logs similar to the real logs. Strongly-typed genetic programming (Montana 1993) allows forcing the structure of the programs in order to get *Q programs*. Indeed, a "guard node" (see Figure 7) can be associated with a given type and force all the members of a genetic population to be of this type. This method reduces the size of the exploration set by limiting it to programs that have a given form.

4.15

Behaviours extracted from actors' traces are sometimes unexpected, i.e., not envisaged in the initial ABM. For example, the processing of the SimCafé's logs shows the emergence of specialized roles: because producers can communicate and exchange coffee without any initial offer from the buyers, some of them, who do not produce a lot of coffee, broadcast messages to buy and sell coffee, so become traders.

Step 3: Multi-agent simulation supporting maieutic processes

4.16

The aim is to implement software agents, either based on the initial ABM or on the behaviours previously extracted from actors' traces, and to introduce them to the distributed RPG in order to transform this one into a particular type of multi-agent simulation, as illustrated in Figure 8.

4.17

The originality of this new simulation scheme is that social actors and software agents play similar roles in the simulation, together or separately. This characteristic offers more possibilities to refine the agents' behaviours. For instance, an off-line maieutic process (versus the one-line maieutic process presented in 4.21 and 4.22) can be applied: designers trace and question actors about their decision-making mechanism. By comparing actors' and agents' traced behaviours, the designers find situations in which the actors act in an unpredicted way, so they ask them why they do not act like the agents. Thanks to the actors' answers, the designers can discover new behaviours, not coded in the agents. Therefore from now on, the actors are able to describe their decision-making mechanism in such a way that the designers can improve those of the agents.

4.18

Such off-line maieutics has been applied to actors' domains that are not formalized, or used in simulations whose goal is not sufficiently defined, as in the case of the *Game of Friends* (Sempé et al. 2005), which:

- has been developed in the framework of research activities of the French-speaking *Institute of Computer Science* (IFI—*Institut de la Francophonie pour l'Informatique*) in Hanoi (Vietnam);
- aims at studying difficulties that someone faces while modelling herself/himself;
- is an RPG about common-pool resource sharing, in which each player initially has some friends (other players), then downloads a movie (from shared resources) for each friend to keep this friend satisfied, and herself/himself can also leave another friend who does not offer any movie or offers a bad one.

4.19

In this game, the players (IFI's students) are asked to build and discuss with their teacher a model of their own behaviours. It is the students who program the agents that play the game on their behalf. They then have to evaluate their model by comparing the game played by them and that played by the agents they code. Here, the students are both designers and actors. The teacher (who can be considered as designer too) also makes a behaviour comparison based on the game's log, and helps the students question themselves.

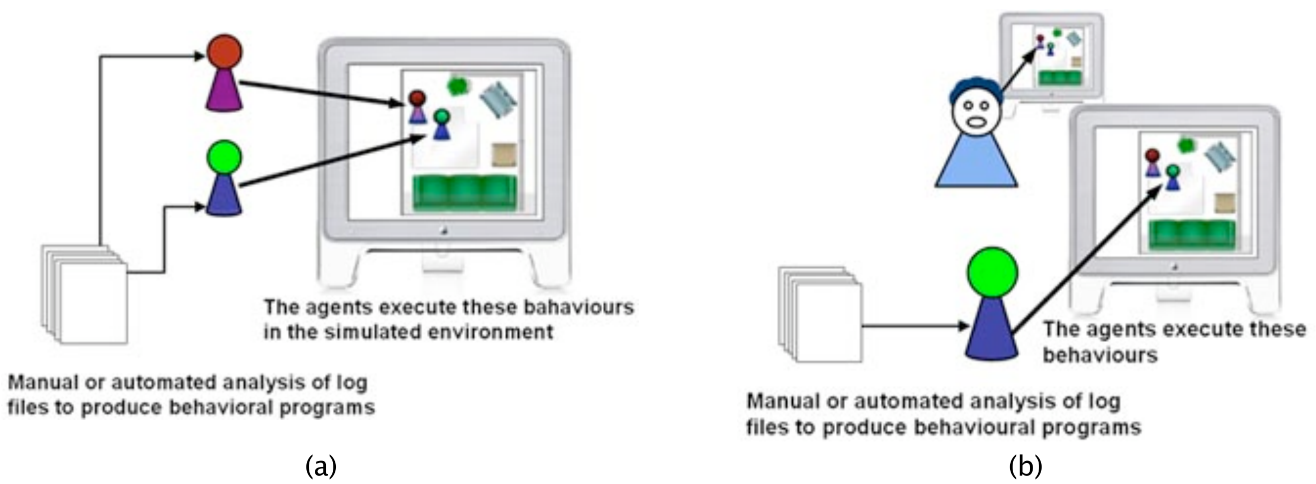


Figure 8. Multi-agent participatory simulation: the agents play the game (a) without actors or (b) together with the actors

4.20
Proto-agent: When software agents are introduced to a simulation, play similar roles as actors do and interact with actors, they present a "prototype" of an agent-based simulation model to be designed. The actors can not only observe their behaviours, but also communicate with them within the simulation, or even question them about their decisions. We thus call them "proto-agents" and use them for our (on-line) artificial maieutic processes.

Step 4: Artificial maieutics with assistant agents

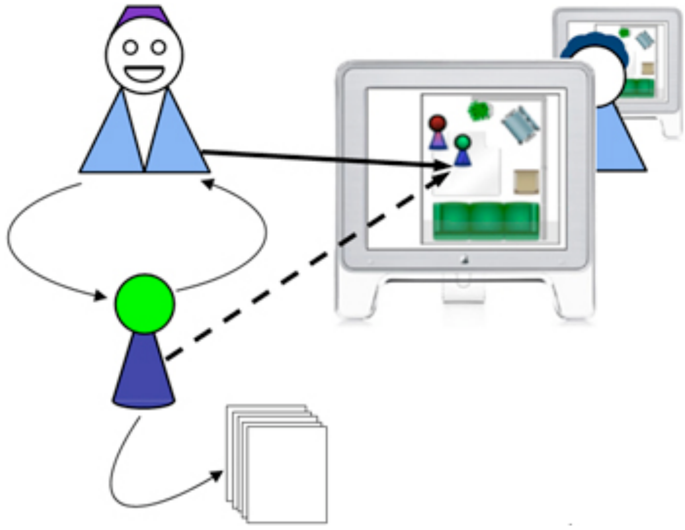
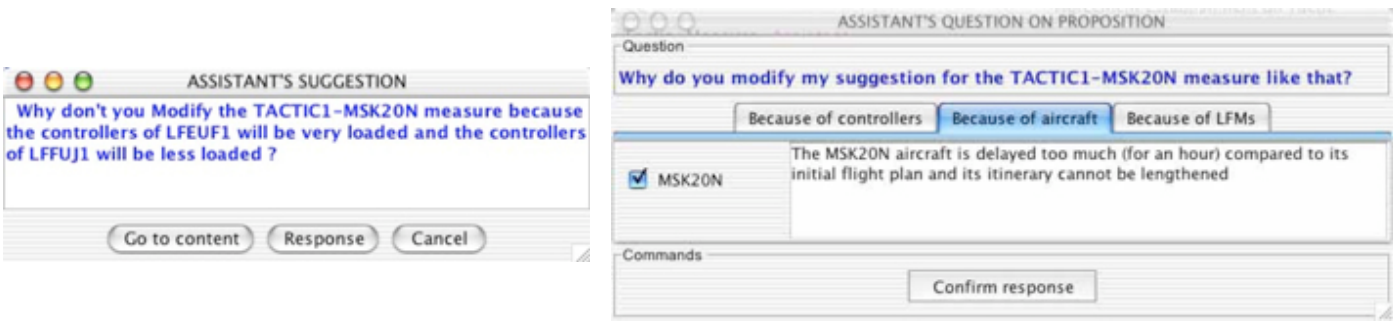


Figure 9. Behaviour refinement through actor-agent interactions

4.21
The (on-line) maieutic method, i.e., the on-line process of question and answer about specific parts of actor behaviour, can be automated by using software agents. Based on the proto-agents implemented and refined in the third step, we provide them with the capacity to directly communicate with actors through computer screens and thus transform them into assistant agents. Each of them is dedicated to an actor, can play the game alone on behalf of the actor or only assist her/him. In other words, an actor and her/his assistant constitute, with respect to the other players, one and only one hybrid player (see Figure 9).

4.22
Normally, the actor plays the role assigned to her/him within the simulation, while the assistant observes, and then proposes actions which can be amended by the actor. These modifications are taken into account by the assistant as well as the results of her/his observations. Such interaction results in a maieutic dialogue in which the assistant asks for reasons for the modifications and the actor explains her/his amendments (see Figure 10). This explanation helps extract the actor's behaviours in order to improve those of any agent (not only the assistant in question). The improvement can be made either by hand by the designers after studying the game's logs or automatically by learning agents. Since suitable learning techniques remain to be discovered, we only consider the first possibility in this design step.



(a) (b)

Figure 10.Examples taken from an interaction session between an actor (flow manager) and her/his assistant. (a) Suggestion of the assistant with justification. (b) Question of the assistant on the amendment and answer of the actor

4.23

Example of actor behaviour extraction: Based on the "log of actor behaviours" (see 3.15) shown in Figure 1, designers formalize agent behaviours by introducing the following abstract rules:

1. "if the difference between the maximum control capacity of a pair of controllers and the number of aircraft present in their sector is lower than three, these controllers will be considered as very loaded";
2. "if an aircraft is delayed for more than an hour compared to its initial flight plan, it will be considered as too delayed";
3. "the route of a too delayed aircraft cannot be more lengthened even if the lengthening of the route allows it to avoid the very loaded controllers";

and then use these rules to create new behavioural rules for the agents.

Towards the fifth step — automated participatory design

5.1

In summary, our PSS design method begins with the formulation of an accompanying methodology that guides the actors during participatory sessions in all the design steps. We currently put into practice methodological solutions from *Cooperative Prototyping* in order to help the actors focus on scenario exploration. The first design step is the proposition of a new kind of design game (i.e., distributed RPG), which needs to be played on connected computers, able to log acts of all players. Secondly, we extract automatically or by hand patterns of interaction or specific behaviours from actors' traces. These behaviours are, thirdly, used to implement artificial players (software agents) to be introduced in the RPG. Since the social actors and software agents play similar roles, we can refine the agents' behaviours by comparing the actors' and agents' traced actions taken in similar situations. In the fourth step, we apply the artificial maieutics to the knowledge extraction process by setting up actor-agent (maieutic) dialogues through user interfaces. Nevertheless, the logs of these particular interactions are still processed by hand to find out new behavioural rules for the agents.

5.2

With the aim to entirely automate the artificial maieutic process, we plan to use machine learning to improve the agents' behaviours. A requirement for the "logs of actor behaviours," produced during the maieutic dialogues, is that they must be formal enough that intelligent software components (e.g., agents) can understand them. It is therefore necessary for us to investigate formal languages like *Q language* (Ishida 2002), which can, at the same time, abstractly and formally represent actor behaviours. This means such a language does not describe the internal mechanisms coded in the agents but their external roles and the interaction scenarios between them. However, which kind of learning is suitable? We believe that interactive techniques from robotics, e.g., learning by demonstration (Atkeson and Schaal 1997; Bentivegna and Atkeson 2000) or by imitation (Hayes and Demiris 1994; Andry et al. 2001), could be applied, because they are based on the direct interaction between a robot and its human tutor.

References

ABEL, N (1999). "Resilient rangeland regions". *VIIth International Rangeland Congress*, Townsville.

ANDRY, P, GAUSSIER, P, MOGA, S, BANQUET, J and NADEL, J (2001). "Learning and communication in imitation, an autonomous robot perspective". In *IEEE transactions on Systems, Man and Cybernetics*.

D'AQUINO, P, LE PAGE, C, BOUSQUET, F, and BAH, A (2003). "Using Self-Designed Role-Playing Games and a Multi-Agent System to Empower a Local Decision-Making Process for Land Use Management: The SelfCormas Experiment in Senegal". In *Journal of Artificial Societies and Social Simulation* 6(3)5. http://jasss.soc.surrey.ac.uk/6/3/5.html

ATKESON C G and SCHAAL S (1997). "Robot Learning From Demonstration". *International Conference on Machine Learning*.

AUBERT, S, LE PAGE, C, JOLY, H, RAZAFINDRABE, R, RANAIVOSON, I F, RALALAOHERIVONY, S B, N'DIAYE, I C and BAIN, D (2002). "Conception, adaptation and diffusion of a computer-assisted role game about phytogenetic resources management at a rural level in Madagascar". *ISEE'02*, Sousse.

BARRETEAU, O (2003). "The joint use of role-playing games and models regarding negotiation processes: characterization of associations". In *Journal of Artificial Societies and Social Simulation* 6(2)3. http://jasss.soc.surrey.ac.uk/6/2/3.html

BARRETEAU, O and BOUSQUET, F (2001). "Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to Senegal River Valley irrigated systems". In *Journal of Artificial Societies and Social Simulation* 4(2)5. http://jasss.soc.surrey.ac.uk/4/2/5.html

BENTIVEGNA, D C and ATKESON, C G (2000). "Using primitives in learning from observation". *1st IEEE-RAS International Conference on Humanoid Robotics (Humanoids-2000).*

BEYER, H and HOLTZBLATT, K (1997). *Contextual Design: A Customer-Centered Approach to Systems Designs*, Morgan Kaufman Publishers, ISBN: 1558604111.

BØDKER, S and GRØNBÆK, K (1991a). "Cooperative Prototyping Studies - Users and Designers Envision a Dental Case Record System". In *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Amsterdam: Elsevier Science Publishers/North Holland.

BØDKER , S and GRØNBÆK, K (1991b). "Cooperative Prototyping: Users and Designers in Mutual Activity". In *International Journal of Man-Machine Studies*.

BØDKER, S, GRØNBÆK, K and KYING, M (1993). "Cooperative Design: Techniques and Experiences From the Scandinavian Scene". In *Participatory Design: Principles and Practices,* Hillsdale, NJ: Lawrence Erlbaum Associates.

BOUSQUET, F, BARRETEAU, O, LE PAGE, C and MULLON, C (1999). "An environmental modelling approach". In *The use of multi-agent simulations, Advances in environmental and ecological modelling*, Elsevier.

BOWELL, T and KEMP, G (2001). *The Critical Thinking*, Routledge, NY.

BURTON, M A (1989). "Experiences with the irrigation management game". *Irrigation and drainage systems*.

CLEEREMANS, A (2001). "Conscious and unconscious processes in cognition". In *International encyclopedia of the social and behavioral sciences*, vol. 4, pp. 2584-2589, Elsevier.

CONRAD, F, BLAIR, J and TRACY, E (1999). "Verbal reports are data! A theoretical approach to cognitive interviews". In *Federal Committee on Statistical Methodology Research*, Arlington.

COSTANZA, R and RUTH, M (1998). "Using dynamic modeling to scope environmental problems and build consensus". *Environmental management*.

DORAN, J (2001). "Intervening to Achieve Co-operative Ecosystem Management: Towards an Agent Based Model". In *Journal of Artificial Societies and Social Simulation* 4(2)4. http://jasss.soc.surrey.ac.uk/4/2/4.html

DRAY, A, PEREZ, P, JONES, N, LE PAGE, C, D'AQUINO, P, WHITE, I and AUATABU, T (2006). "The AtollGame Experience: from Knowledge Engineering to a Computer-Assisted Role Playing Game". In *Journal of Artificial Societies and Social Simulation* 9(1). http://jasss.soc.surrey.ac.uk/9/1/6.html

EDEP (2006). http://www.eurocontrol.fr/projects/edep/

EHN, P and SANDBERG, Å (1983). "Local Union Influence on Technology and Work Organization". In *Systems Design For, With and By the Users*, Amsterdam: North-Holland Publishing Company.

ETIENNE, M, LE PAGE, C and COHEN, M (2003). "A step by step approach to build up land management scenarios based on multiple viewpoints on multi-agent systems simulations". In *Journal of Artificial Societies and Social Simulation* 6(2)2. http://jasss.soc.surrey.ac.uk/6/2/2.html

GRØNBÆK, K (1991). *Prototyping and Active User Involvement in System Development: Towards a Cooperative Prototyping Approach*, Ph.D. thesis, Aarhus University.

GUERREAU, R (2002). *FAM Project - Operational Concept Document*, Research report, EuroControl.

GUYOT, P (2003). *Simulations multi-agents participatives: Vers une méthode de validation et de consolidation des modèles de pratiques collectives*, Master's Thesis, University of Paris 6.

GUYOT, P and DROGOUL, A (2005). "Two Sides of Emergence in Participatory Simulations". *Social Inspired Computing*.

GUYOT, P and HONIDEN, S (2006). "Agent-Based Participatory Simulations: Merging Multi-Agent Systems and Role-Playing Games". In *Journal of Artificial Societies and Social Simulation* 9(4)8. http://jasss.soc.surrey.ac.uk/9/4/8.html

HAMEL, A, PINSON, S and PICARD, M (2005). "A New Approach to Agency in a Collaborative Decision-Making Process". In *Intelligent Agent Technology*, Compiègne.

HAYES, G and DEMIRIS, J (1994). "A robot controller using learning by imitation". *2nd International Symposium on Intelligent Robotic Systems*.

HOLTZBLATT, K and JONES, S (1993). "Contextual Inquiry: A Participatory Technique for System Design". In *Participatory Design: Principles and Practices,* Hillsdale, NJ: Lawrence Erlbaum Associates.

HUGHES, J A, KING, V, RODDEN, T and ANDERSEN, H (1994). "Moving out from the control room: Ethnography in system design". *ACM 1994 Conference for Computer Supported Cooperative Work*, NY: ACM Press.

IHLSTRÖM, C, LUNDBERG, J, REHNSTRÖM, C and VIMARLUND, V (2002). "Online Newspapers in Search for Profitable Services". *Challenges and Achievements in E-business and E-work*, Prague.

ISHIDA, T (2002). "Q: A Scenario Description Language for Interactive Agents". In *IEEE Computer*.

KENSING, F and MADSEN, K H (1991). "Generating Visions: Future Workshops and Metaphorical Design". In *Design at work: Cooperative design of computer systems*, NJ: Lawrence Erlbaum Associates.

KIHLSTROM, J F (1993). "The rediscovery of the unconscious". In *The mind, the brain, and complex adaptive systems*, H. Morowitz, Santa Fe, N.M.: Santa Fe Institute.

NGUYEN-DUC, M (2005). *Vers la conception participative de simulations sociales - Application à la gestion du trafic aérien*, PhD thesis, University of Paris 6.

NGUYEN-DUC, M, BOUCHER, A, DROGOUL, A and DUONG V (2004). "Towards participatory design of agent-based simulations: An application in air traffic management". *Agent-Based Simulation 5*, Lisbon.

MONTANA, D J (1993). *Strongly typed genetic programming*, BBN Technical Report #7866, Bolt Beranek and Newman Inc.

MUCCHIELLI, A (1983). *Les jeux de roles*, Presses Universitaires de France.

NYGAARD, K and BERGO, O T (1975). "Trade Unions New Users of Research". In *Personnel Review*.

PYNADATH, D V and TAMBE, M (2003). "Automated teamwork among heterogeneous software agents and humans". In *Journal of Autonomous Agents and Multi-Agent Systems*.

PURNOMO, H, PRIYADI, H, YASMI, Y, YULIANI, L and PRABHU, R (2002). "Development of Multi-stakeholder Scenarios of Forest Management: A Multiagent System Simulation Approach". *Participatory Modeling Symposium*, Harare.

RAMANATH, A M and GILBERT, N (2004). "The Design of Participatory Agent-Based Social Simulations". In *Journal of Artificial Societies and Social Simulation* 7(4)1. http://jasss.soc.surrey.ac.uk/7/4/1.html

REITSMA, R, ZIGURS, I, LEWIS, C, WILSON, V and SLOANE, A (1996). "Experiment with simulation models in water-resources negotiations". In *Journal of water resources planning and management*.

REYNOLDS, T J and GUTMAN, J (1984). "Laddering: Extending the Repertory Grid Methodology to Construct Attribute-Consequence-Value Hierarchies". In *Personal Values and Consumer Psychology*, Pitts, R.E. and Woodside, A.G. (Eds) Lexington Books, Lexington.

ROTHMAN, D, ROBINSON, J and BIGGS, D (2003). "Signs of life: linking indicators and modelling in the context of QUEST". In *Implementing Sustainable Development - Integrated Assessment and Participatory Decision-Making Processes*.

SAKURAI, S and YOSHINO, H (2004). "Toward Socratic Method Support (Assist) System for Law". *Conference for Law School Computing*. Seattle.

SAMPLES, K (1998). "The Socratic Method". http://www.str.org/site/News2?page=NewsArticle&id=5631

SCHREIBER, G, AKKERMANS, H, ANJEWIERDEN, A, DE HOOG, R, SHADBOLT, N, VAN DE VELDE, W, and WIELINGA, B (1999). "Knowledge Engineering and Management". In *The CommonKADS Methodology*, Cambridge, MA: MIT Press.

SCHULER, D and NAMIOKA, A (1993). *Participatory Design: Principles and Practices*, Hillsdale, NJ:

Lawrence Erlbaum Associates.

SEMPÉ, F, NGUYEN-DUC, M, BOUCHER, A and DROGOUL, A (2005). "An artificial maieutic approach for eliciting experts' knowledge in multi-agent simulations". *Autonomous Agents & Multi-Agent Systems*, Utrecht.

STOLTZ, S and KY, P (2001)."Reducing Traffic bunching more Flexible Air Traffic Flow Management". *4th USA/Europe ATM R&D Seminar*, New-Mexico.

TAMBE, M (1997). "Towards Flexible Teamwork". In *Journal of Artificial Intelligence Research*, 1997.

TRIGG, R H, BØDKER, S and GRØNBÆK, K (1990). "A Video-based Analysis of the Cooperative Prototyping Process". *13th IRIS conference*, *Reports on Computer Science & Mathematics no. 108*, Åbo.

VAN ASSELT, M, MELLORS, J et al. (2001). *Participatory methods in Integrated Assessment*, Maastricht, Nederland, International Center For Integrative Studies.

WEUSIJANA, B, RIESBECK, C and WALSH, J (2002). "The SASK Mentor Project: Using Software and Socratic Methods to Foster Reflective Thought in a Learning Environment". *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Norfolk.

Return to Contents of this issue