Intelligent Tutoring Systems and Serious Game for Crisis Management: a Multi-Agents Integration Architecture.

M'hammed Ali Oulhaci, Erwan Tranvouez, Bernard Espinasse, Sébastien Fournier
Aix-Marseille Université, CNRS, LSIS UMR 7296
38, Av. Normandie-Niemen 13397 Marseilles Cedex 20 FRANCE
Email: ali.oulhaci,erwan.tranvouez,bernard.espinasse,sebastien.fournier@lsis.org

***Abstract*—Crisis management, serious games (SG) are more and more used for training. SG permit to reduce the cost of a such training and saving time, and in general provide a fun way to learn. In this paper we propose to bring together SG and Intelligent Tutoring Systems (ITS) in the context of the SIMFOR project, a SG for training in crisis management. We discuss the problems and needs of serious games and an overview of existing works. To enhance the learning aspect in SG, we propose the integration of different ITS functionalities to the SG. To perform this integration process, we propose an collaborative agent-based architecture. This architecture is presented in detail and illustrated on a realistic game scenario.**

***Keywords*-Serious games, intelligent tutoring system, crisis management, multi-agent system.**

I. Introduction

All cities are confronted to major risk management. Today, in response to major risks, business leaders, schools, cities or regions should implement specific prevention plans and improve stakeholders awareness with scenarios exercises. Indeed, the only way to test these plans is to make exercises in real conditions, which can become very heavy in terms of organization and very expensive. To reduce the cost and saving time, computer tools are solicited. There are many tools dealing with the issue of risk management. In [1], Amokrane et al. discusses the problem of risk management in SEVESO sites (high-risk site) for staff training. In [2], Querrec et al. proposes SecuReVi, a training tool for firefighters. In Marion et al [3], risk management is discussed in safety on aircraft carriers. However the large majority of these tools do not satisfy the demand of risk management actors and are reserved for specialized trades target (eg firefighters) or to a particular domain. The objective we have with SIMFOR project is to propose a serious game dedicated to train non-professional to risk management, and this through enhancing the educational aspect by adding different modules related to Intelligent Tutoring System (ITS). This integration is achieved according to a collaborative multi-agent system to ensure the monitoring process and evaluation of learners.

The paper first presents in section II, the concept of serious game in the context of the SIMFOR project (a serious game for training of non-professionals in crisis management), and underlines the needs for pedagogical support in serious games. Section III present general functionalities and architecture of Intelligent Tutoring Systems. Section IV presents an agent-based integration architecture permitting to integrate in a GS different ITS functionalities, with its components. Role of each component in the game and in the evaluation process are detailed. In section V we illustrate how the architecture enables ITS-like functionalities in running a game scenario. Finally we conclude and present future works for the SIMFOR project.

II. Serious games : the SIMFOR project

Many definition of Serious Game are available in the literature either from a game or training perspective or seen as a offspring of elearning. In

[4], Julian Alvarez offers a unified definition of a serious game: *A computer application, [aims] to combine with consistency, both serious (Serious) aspects such as non-exhaustive and non-exclusive, teaching, learning, communication, or the information, with playful springs from the video game (Game), adding this association must be done by implementing a pedagogical scenario".*

According to this definition, SIMFOR is a SG[1] developed by SII in partnership with Pixxim company[2], SIMFOR is dedicated to training non-professional of *"risk management"* actors required at a moment or another to play a part in the global coordinated response to an incident. The project objective is to create an adaptive real-time and realistic training tools involving multiple actors in the context of crisis management.

As major crisis can mobilize several hundred stakeholders (from the regional Prefect in his office to the firefighter in the field), SIMFOR is a multi-player game and allows different actors' profile to learn shared or specific skill. These stakeholders must communicate and work together in order to restore a normal situation. The SIMFOR project thus faces two issues: i) allowing scenario with an incomplete set of actors by replacing missing roles by simulated avatars and ii) the ability to monitor and the evaluation of actors' actions and decisions in order to assess individually and collectively their actors (ie learners) capacity to manage a crisis.

Many works, in the literature, relates the learner support and assessment issues [5][6], but SIMFOR is a multi-actor game dealing with two types of evaluation: individual and collective. Solving the crisis requires the resolution of all procedures of the stakeholders, so individual evaluation can affect the collective evaluation and conversely. For example if a learner has successfully realized his procedures, but the main purpose was not reached (with material and human loss), the learners must be evaluated on their individual and collective performance to infer the reason of failure (lack of communication, missing procedure of another learner, ...).

[1]selected from the serious gaming call for project launched by the French Secretary of State for Forward Planning and Development of the digital economy

[2]resp. http://www.groupe-sii.com I& http://www.pixxim.fr

III. Intelligent tutoring systems and Serious Games

The most important point in a serious game is the pedagogical support provided to the learner. In this context, the concept of adaptive serious games is often used [7]. Adaptation, as applied to a serious game, reflects its ability to change structurally in response to certain events triggered by the learner (player). In these different works [8][9], a simple representation of the learner is proposed whereas knowledge about application domain is not represented explicitly hindering learner evaluation. Intelligent Tutoring System (ITS) constitutes another approach stemmed from the Technology Enhancing Learning (TEL) researchers community aiming at individualizing training. ITSs propose to represent (explicitly or not) knowledge (declarative or procedural) from the domain under study [10] as well as knowledge to be acquired by the learner (its mental state)[11] during training session.

Burns [12] defines an ITS architecture with: i) an *Expert module* containing the expert domain knowledge; ii) a *Learner module* that contains what learner knows in the domain; iii) a *Tutor module* which aims at identifying the learning gaps so the system can adapts its strategies to help the learner in filling these gaps (this module is also called Pedagogical module) and iv) two others modules representing the communication channel of the ITS : the *The instructional environment* and the *Human-Computer Interface*. Their roles are detailed below.

- **Expert module**: J. Anderson [10] identifies the most important task in the design of the expert module as how to model (codify) knowledge. The expert module must contain specific and detailed knowledge from people with many years of experience in a particular domain.
- **Learner module**: an ITS must construct a model for understanding the learner and then use this understanding to adapt instructions to the specific needs of the learner. The learner module typically use the same type of knowledge representation used in the expert module ie represented as a subset of expert knowledge.

- **Tutor(Pedagogical) module**: An ITS must have three tutoring characteristics: a) the control over the representation of educational knowledge for the selecting and scheduling the exercises; b) the ability to answer questions from students on educational objectives and content, and finally c) strategies to determine when "students" need help with their solutions.
- **The instructional environment**: The instructional environment is an important component in an ITS. It supports the learner during his training (implying monitoring), it includes the tools provided by the system to facilitate the learning (e.g. hints to the learner).
- **Human-Computer interface - HCI**: The problem in the design of human-computer interface is that the student must use the technology itself to learn, while being not necessarily an expert user.

Our goal is to improve SIMFOR SG in adding specific ITS modules: a *Learner module*, an *Expert module* and a *Pedagogical module*. The *Interface module* will be assumed by the SIMFOR interface maintaining a playful learning principle while improving the adaptive aspect with the ITS modules.

IV. A Multi-Agent architecture for ITS integration in the SIMFOR project

Our research work aim to transform SIMFOR into an ITS while maintaining the didactic aspect of a SG. To reach this aim we need to add two main functionalities to SIMFOR:

- A simulation function based on a multi-agent system (MAS) to simulate the behaviour of the scenario's non-playing actors.
- An evaluation function, based on an agent-based collective evaluation method.

To add this two new functionalities to SIMFOR, we propose to develop a cooperative multi-agent architecture that contains and integrates the different modules related to an ITS including and a SG. Figure 1 describe the different modules of the integrated system (SG+ITS). The integration process is performed by a multi-agent architecture composed of these main components:

A. The SG module (SIMFOR)

SIMFOR has 3D models, an user interface, a simulation module, and data models (Actor, Means and Disaster). By analogy with ITS, the user interface and the pedagogical environment are represented by SIMFOR 3D UI. The 3D environment plays a very important role because 3D brings immersive aspect to the players, allowing them to enact their role and to ignore the tool used (computer, simulator, ...).

B. The behaviours simulation module

The behaviours simulation module simulates humans behaviours to replace absent players with "artificial" actors (game agent). To do so, an *ad hoc* BDI (Belief Desire Intention) [13] agent model has been developed. We have facilitated the design of agent by providing the designer (of the scenario) an editor to configure and set up agents. The multi-agent system can handle game models to simulate human behaviour (displacement of an actor, intervention on disaster ...).

C. The evaluation modules

The evaluation module aims to provide an assessment of players in real time to the pedagogical agent. This module is MAS composed with the following interacting agents:

- **Data source agent**, for each "information" component, a data source agent is associated, the data source agent have for mission to collect the necessary information for the monitoring process.
- **The indicator agent**, this kind of agent have to compute or to select the appropriate information for learner evaluation.
- **Evaluation Agent**, using indicators, the Evaluation Agent assess learners performance. The result of evaluation is used thereafter by the pedagogical module and can also be used for post-game debriefing.

D. The pedagogical module

The pedagogical module plays the role of a virtual tutor accompanying the learner in his training. The pedagogical module provide support and

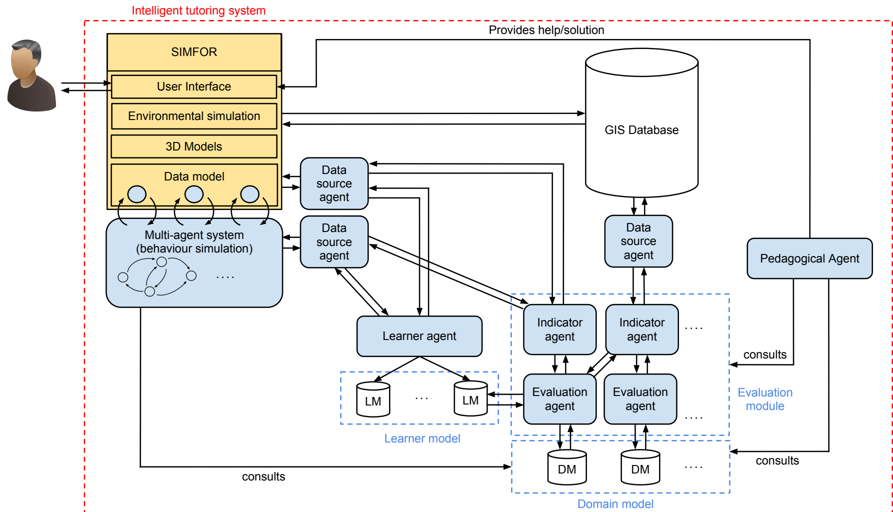

Fig. 1. General working of the system

help to the learner to optimize learning in the virtual environment. The pedagogical module analyses the situation (evaluation module, domain module, learner module) and select the appropriate strategy (propose action to perform, display help, correct, ...). This pedagogical module is mainly based on a specific agent called Pedagogical Agent.

E. The knowledge representation

All knowledge used or produced by the previous modules of our proposed architecture are stored in the following models:

- **The domain model:** the domain model represent the different concepts of crisis management and its segmented into parts representing a role or a skill to learn. The domain model is represented by an ontology that describe the different concepts of risk management. Presently, the ontology is specific to SIMFOR but will be extended to respond to every request of a domain expert. The domain model represents the expert module of an ITS.
- **The learner model:** for each learner or agent, a learner model is associated. This model represents the mental state of actors at the time *t*. The Learner Agent (LAg) collects learner actions and knowledge, and stores them in the learner model. The LAg also collects agent knowledge and actions, this information will be used for collective evaluation. The learner model and the LAg represent the learner module of an ITS. The data structure of learner model is the same structure of the domain model, and it is an ontology, in order to use the knowledge overlap by the evaluation agent.

V. General working of the integrated system: A scenario example

To illustrate how such ITS would work, we present an example of scenario defined with the help of a domain expert. We present a simplified example of the missions to be performed by the actor playing the role of CODIS (Departmental Operational Fire and Rescue Services Centre) for a TDM scenario (Transportation of Dangerous Material). The scenario begins with a TDM truck which has spilled due to a traffic accident. The tank is damaged and the fuel is spreading over the road. A witness to the accident gave the alarm by calling the CODIS.

The CODIS must perform four missions after receiving the alert: 1) CODIS has to send firefighters on the scene to retrieve information about the accident. Once the information on the accident

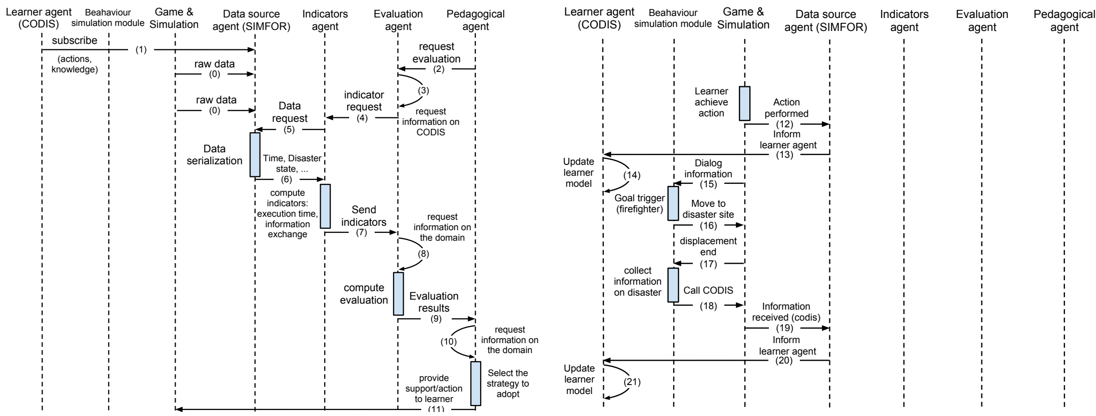

Fig. 2. Sequence diagram of the interaction of different agent

is received (transmitted by the firefighter in the field) and after confirmation of a TDM accident, the CODIS must 2) primarily inform an officer (firefighter) to take the necessary measures. Next, the CODIS must 3) complete an information sheet on the disaster transmitted by fax to the mayor, prefect and the sub-prefect. Finally the last mission is to 4) report to the acting officer in the local OCP (operational command post) once it is sent by the prefect.

Figure 2 describes, with an UML sequence diagram, the interaction between different agents of the training system during the monitoring process. At each time cycle, the *Pedagogical Agent* (**PAg**) analyses the game situation to help learner by sending request to *Evaluation Agents* (**EAg**) to get learner assessment (arrow 2, figure 2).

The first action to be performed by the CODIS is to call a firefighter to warn him of the accident. The **EAg** retrieve the relevant information from the learner model (3) (level, role, pending procedure, ...), then requests an *Indicator Agent* (**IAg**) to get the indicators' value pertaining the evaluation (4). The **IAg** asks the adequate *Data Source Agent* (**DSAg**) to retrieve the indicator input data (5) (from a simulator, a Database, ...). In our case (action call), the **IAg** must select a time of execution to perform the action and the information exchange during the call (if the actor called is played by an agent, information is retrieved from the dialogue history). The time of execution and information exchange is retrieved from **DSAg** associated to SIMFOR (if the actor called is an agent, information exchange is retrieved from **DSAg** associated to the multi-agent system simulating human behaviours).

The **DSAg** can receive raw data continuously (0), and when an **IAg** requests information, the **DSAg** serializes raw data (or consolidate it) and sends it to the **IAg** (6) which in turn calculates (selects) adequate indicator and send it to the **EAg** (7). With learner information, domain model and game difficulty, the **EAg** will compute an assessment and sends the result to **PAg** (9). The **PAg** can then use domain model information (10) to predict the next course of action and selects a strategy to help the learner through the SIMFOR 3D interface (11).

Different type of strategy have been defined according to the evaluation result:

- *Let the learner performs the action*: the study conducted by [14] have shown that learners who received delayed feedback have better retention of skills over time. If the student is experimented, the pedagogical agent let the learner find solution by himself.
- *Give a clue*: if the learner is a novice, the pedagogical agent begins by giving clues about the procedure to follow.
- *Propose action*: if the learner has difficulties to perform the procedure (time attributed to the action exceeded), the pedagogical agent pro-

pose action to realize (in the case of CODIS, call firefighter).
- *Do the action in place of the learner*: if the learner does not know how to do the action, the pedagogical agent performs the action in his place while explaining how to do it.

To keep a history of learner actions and his knowledge evolution, the learner agent subscribes to data source agent. So, when learner perform action or revive information, the data source agent inform learner agent (13, 20) and the learner agent update its learner model (14, 21).

When a role is played by an agent (Game Agent), the behaviour simulation module can handle game model (16) and interact with the learner (18).

The evaluation request of the pedagogical agent (2-9) is repeated each time cycle, this is not shown in the diagram for better understanding.

At present the number of action that can be achieved in SIMFOR is somewhat limited, but nevertheless allows to analyse the activity of the learner by comparing the actions performed with the domain model.

VI. Conclusion and perspectives

Crisis management, serious games (SG) are more and more used for training. SG permits to reduce the cost of a such training and saving time, and in general provide a fun way to learn. In this paper we propose to bring together SG and Intelligent Tutoring Systems (ITS) in the context of the SIMFOR project,a SG for training in crisis management.

To perform this integration process, we have proposed an agent-based architecture permitting to add learning functionalities to SG. In the context of SIMFOR improvement, the different modules of this architecture are been presented, and the general functioning of this architecture has been illustrated on a realistic game scenario.

Future work in the SIMFOR project targets i) the finalization the multi-agent system architecture which simulates non-played actor with organizational architectures; ii) the addition of a (ITS like) pedagogical module for monitoring players in real time and to offer them pedagogical solutions to help them in their training, and a post-game diagnostic that will identify weaknesses of the learners and the skills and competencies that remain to be acquired. This ITS must also enable monitoring agents (game agent), so the individual and collective evaluation is therefore based on both the human player and the agents (with normal behaviour or intentionally erroneous).

References

[1] K. Amokrane, D. Lourdeaux, and J. Burkhardt, "Hera: Learner tracking in a virtual environment," *IJVR : International Journal of Virtual Reality*, vol. 7, no. 3, pp. 23–30, September 2008.

[2] R. Querrec, C. Buche, E. Maffre, and P. Chevaillier, "Sécurévi : virtual environments for fire-fighting training," in *Proceedings of the 5th Virtual Reality International Conference, VRIC 2003*, May 2003.

[3] N. Marion, C. Septseault, A. Boudinot, and R. Querrec, "Gaspar : Aviation management on an aircraft carrier using virtual reality," in *Cyberworlds 2007 proceedings*, Oct. 2007.

[4] J. Alvarez and D. Djaouti, "An introduction to serious game - definitions and concepts," in *Proceedings of Serious Games & Simulation for Risks Management Workshop*. LARSEN Science, December 2011.

[5] W. Johnson, J. Rickel, R. Stiles, and A. Munro, "Integrating pedagogical agents into virtual environments," *Presence*, vol. 7, no. 6, pp. 523–546, 1998.

[6] D. Lourdeaux, J. Burkhardt, F. Bernard, and P. Fuchs, "Relevance of an intelligent agent for virtual reality training," *International Journal of Continuous Engineering and Lifelong Learning*, vol. 12, no. 1/2/3/4, pp. 131–143, 2002.

[7] N. Hocine and A. Gouaich, "A Survey of Agent Programming and Adaptive Serious Games," Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier - LIRMM, Tech. Rep., 2011.

[8] C. Conati and M. Manske, "Evaluating adaptive feedback in an educational computer game," in *Intelligent Virtual Agents*. Springer, 2009, pp. 146–158.

[9] A. Ram, S. Ontañón, and M. Mehta, "Artificial intelligence for adaptive computer games," in *Twentieth International FLAIRS Conference on Artificial Intelligence*, 2007.

[10] J. Anderson, "The expert module," *Foundations of intelligent tutoring systems*, pp. 21–53, 1988.

[11] K. VanLehn, "Student modeling," *Foundations of intelligent tutoring systems*, pp. 55–78, 1988.

[12] H. Burns and C. Capps, "Foundations of intelligent tutoring systems: An introduction," *Foundations of intelligent tutoring systems*, pp. 1–18, 1989.

[13] A. Rao, M. Georgeff *et al.*, "Bdi agents from theory to practice," in *Proceedings of the first international conference on multi-agent systems (ICMAS-95)*. San Francisco, 1995, pp. 312–319.

[14] R. Schmidt and R. Bjork, "New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training," *Psychological Science*, vol. 3, no. 4, pp. 207–217, 1992.