International Journal of Computers and Applications, Vol. No. , 2004

MULTIAGENTS SYSTEMS FOR VIRTUAL ENVIRONMENT FOR TRAINING. APPLICATION TO FIRE-FIGHTING

R. Querrec*, C. Buche*, E. Maffre*, P. Chevaillier*

Abstract

This study concerns virtual environments for training in operational conditions. The principal developed idea is that these environments are heterogeneous and open multiagent systems. The *MASCARET* model is proposed to organize the interactions between agents and to give them reactive, cognitive and social abilities to simulate the physical and social environment. The physical environment represents, in a realistic way, the phenomena that learners and teachers have to take into account. The social environment is simulated by agents executing collaborative and adaptive tasks. They realize, in team, procedures that they have to adapt to the environment. Users participate to the training environment through their avatar. To validate our model, the *SECUREVI* application for fire-fighters training is developed.

Key Words

Virtual Environment for Training, Multiagents Systems, Collaborative Work.

1. Introduction

This study concerns virtual environments for training in operational conditions. We want to simulate and immerge the learners in their professional environment. This enables them to manipulate the environment so that they can *"learn while doing"*. This idea is driven by the "constructivism" paradigm defined by Piaget [1] and can find a good implementation in virtual reality techniques as presented by Fuchs or Burdea [2, 3]. Our definition of Virtual Reality is Tisseau's [4] one, which proposes to provide autonomy to models involving in the virtual environment by giving them the *"triple mediation of senses, decision and action"*. So, the main developed idea is that virtual environments for training are heterogeneous and open multiagent systems. Those MultiAgents Systems (MAS) have been presented by Demazeau [5] using the *VOWELS* model considering a MAS with four vowels: **A**gent, **E**nvironment, **I**nteraction and **O**rganization. From our point of view, we consider the user of a virtual environment as other autonomous agents because he can interact with the environment and with other agents or users in the same way. Then, like Tisseau [4], we propose to add a last vowel, the letter **U** for **U**ser, in the *VOWELS* model.

Our work is to design software to immerge users in operational situations that they cannot meet in reality for training exercises because they are too dangerous or inaccessible. The subject of our application is to train fire-fighters officers to operational management and command. We want then to design software to train to collaborative and procedural work. Our goal is to help users to take decision in operational situation and not to teach them technical gestures.

2. The MASCARET model

Our objective is to train teams to collaborative and procedural work in a physical environment. In this case, we have to simulate in a *realistic* way the physical environment and the *collaborative* and *adaptive* team member's behaviour in the social environment. We consider that the evolution of those environments results from simulation of autonomous agent's behaviour and their local interactions. Thus, we propose a model called *MASCARET*, in which we use MultiAgents Systems to simulate Collaborative Adaptive and Realistic Environments for Training. It intends to organize the interactions between agents and to give them abilities to evolve in this context.

2.1. The organisational model

As the users have to be integrated both in the social (member of a particular team) and in the physical environment (to undergo a lick of gas for example), we propose, first a generic organisational model allowing to represent the physical and the social environment. The model we propose is inspired from the UML meta-model and is founded upon the concepts of `Organisation` (instead of collaboration in UML), `Role`, `Behavioral Feature` and `Agent` (instead of object) (Fig. 1). Hannoun [6] has already proposed a

* Laboratoire d'Ingénierie Informatique, Ecole Nationale d'Ingénieurs de Brest, Technopole Brest Iroise, CS 73862, 29 238 Brest cedex 3, France, email : [querrec, buche, maffre, chevaillier]@enib.fr

(paper no. 202-1453)

organisational model for multiagents systems, but this model, dedicated to collaborative realisation of procedures, is not enough generic to solve our problem. Ferber [7] has also proposed such a model called Agent/Group/Role, but this model is more a pattern for conception than a model which really formalizes the concepts of organisation and roles. In our model, the aim of the `Organisation` is to structure interactions between agents; it enables each agent to know its partners and the role they are playing in the collaboration. The concept of `Role` represents the responsibilities (realized by `Behavioral Feature`) played by agents in the organisation. Agents have then an organisational behaviour which permits them to play or abandon a role in an organisation. This behaviour also enables agents to take into account the existence of the other members.

This model is a generic model in the way that all the resulting classes are abstract. This organisational model is then derived to implement two concrete organisations representing physical and social environment that have to be simulated in the virtual environment for training.

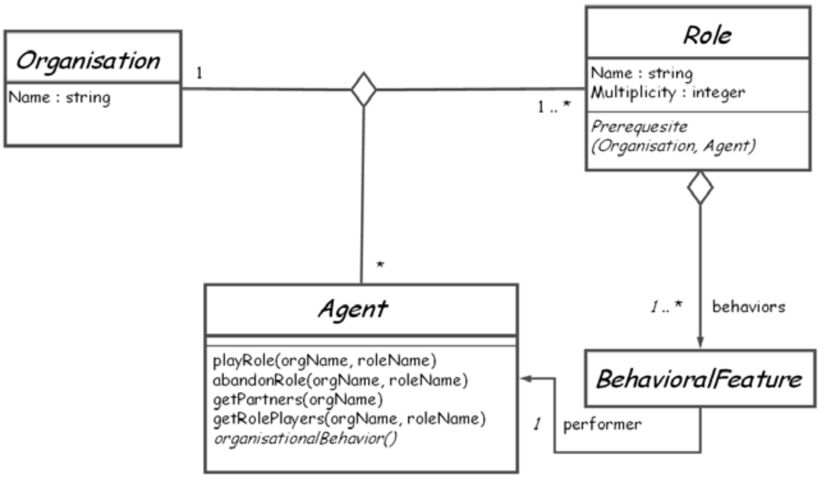

Figure 1. The *MASCARET* organisational model.

2.2. The physical environment

In a virtual environment for training, the users' (learner and teacher) physical environment must be realistic, interactive and act in "real-time". Then, to reach the constraints of virtual reality, the models we use to simulate physical phenomena are obviously simplified. Moreover, the teacher may want, for pedagogical reasons, inhibit some phenomena. Therefore, we have to propose models compatible with a weak coupling between each phenomenon. Moreover, although all interactions have potentially effects on the two agents involved, in most cases the effect of one agent is more important. We consider then that the interactions between the agents have a privileged direction. Then, reactive behaviour of agent evolving in physical environment is to perceive interaction situations and to act consequently. One practical limit of individual-based models is that each agent can potentially perceive all the others. The complexity of the algorithm is in this case $O(n^2)$. Then, we have to design rules to organize these interactions between reactive agents. Then, we use the generic organisational model proposed before. In this case, the organisation is a network where agents are connected together when they are in interaction. We call this organisation an interaction network (`InteractionNet`, Fig. 2). To represent the concept of privileged direction in interactions, we define two particular roles called `Source` and `Target`. The goal of source agents is to give information on their internal states to other agents (targets) so that they can compute the interaction's value and their internal state. The interaction can be detected by the two agents involved, but, for "real-time" computation reasons, it is better if only one agent detects it (one of two agents or another one else). We then define a `Recruiting` role which has the responsibility to maintain the knowledge of each agent upon the structure of the organisation. This means that an agent playing this role has to detect the interaction situations. The internal architecture of reactive agents matchs the constraint of physical phenomena weak coupling presented before, because an agent can have several `Reactive Behavior`, each one participating in a different interaction network. This elementary behaviour consists in the computation of a vector of internal state variables after the evaluation of inputs (from the interactions where the agent is a target) and presents a pertinent internal state to other agents (potentially targets of an interaction where the former agent is a source).

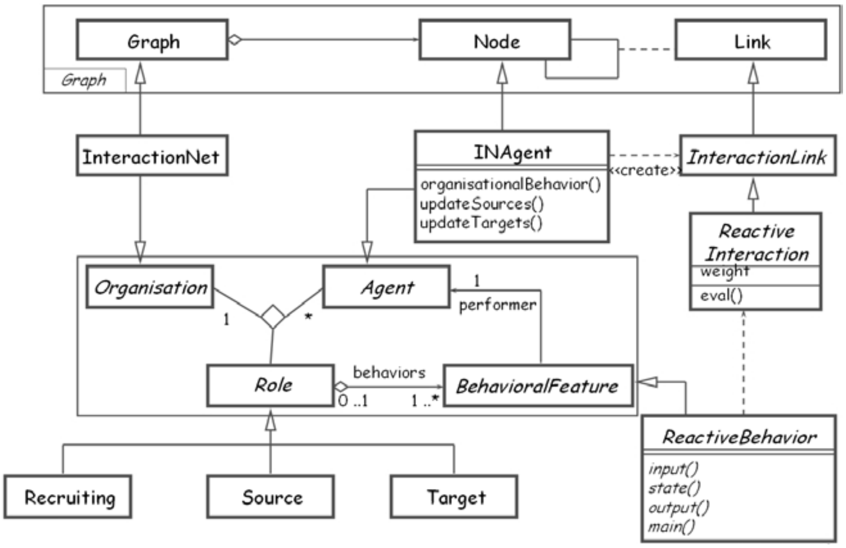

Figure 2. The Interactions Network.

2.3. The social environment

The physical environment is also populated by more "intelligent" agents. They undergo the physical phenomena and they act on them as reactive agents, but the way they choose their actions is carried out on a higher level of abstraction. Those agents are various humans involving in the training (learners and teachers) who are played by autonomous agents. In our case, the social environment is structured and each member knows its roles and those of its partners. The interactions between the team members are also structured and arranged by the mean of procedure known by all members. We thus derive our generic organisational model to formalize this concept of team. We are interested in the case where the action's coordination between team members is already envisaged and written in procedure. On the other hand, the environment being dynamic, agents can need to adapt the scenario to the environment. The procedure must then have a semantic representation so that agents can reason above it. To describe a procedure we use an executable interpretation of the temporal logic of Allen (logic on the intervals of time) given by Deloor and Chevaillier [8].

The reasoning of team members relates on organisation, procedures and actions. We propose a model of agent having local organisational knowledge. An agent is divided into a decisional part and a part represented by modules of perception of the physical environment, communication and actions (Fig. 3). MASCARET doesn't propose a model for action recognition then when an agent starts or stops an action, it broadcasts a message that enables other members to follow the evolution of the procedure. Agents must carry out actions of the procedure and adapt to situations not envisaged. The procedure describes interactions between agents in an optimal case, and leaves to the agent the responsibility to build implicit plans (not clarified in the procedure) considered as natural within an applicative situation. Moreover, the procedure organize actions of a semantic level which we call « actions trades » such as « sprinkling a fire » in the case of firemen procedures, whereas the implicit plans arrange actions of a generic semantic level for humans such as « going at a point ».

For that, the agent must be able to reason on actions and we propose a model of goal directed actions having pre-conditions and post-conditions. Thus, before carrying out an action, the agent must make sure that pre-conditions of this one are checked. If it is not the case, it builds itself a plan by back chaining on pre-conditions and post-conditions of actions. When this behaviour is at fault, the agent calls its organisational behaviour which can help it to find a solution with another team member. Thus, in a hierarchical organisation, when an agent has a problem which it cannot solve, it refers to its superior. Then, the superior has the responsibility to find a solution among its subordinates. If it does not find any, it refers to its own superior about it. This mechanism is based on a slightly modified version of the *Contract Net Protocol*.

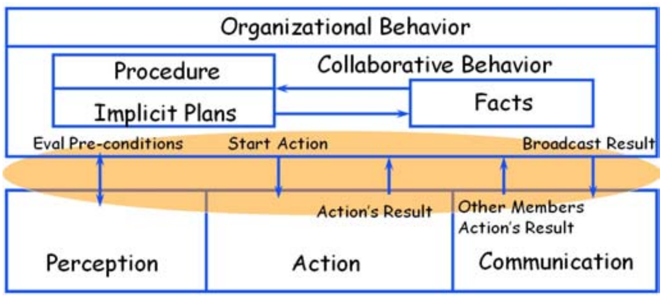

Figure 3. Architecture of rational agents.

2.4. The users

The avatar is the representation of the user in the environment. The model of avatar we propose is the same as rational agents except for the inhibition of some links between the decision part and the action part. So the avatar is also an autonomous reactive agent undergoing the physical environment. It is not from the user responsibility to decide if he has to undergo it, the avatar does it autonomously. For example, if a user is walking through a leakage of gas, its speed and its perception abilities will automatically decrease.

Moreover, all modules composing the avatar (as a rational agent) are active and thus remain potentially usable; however certain links are deactivated. Thus the collaborative behaviour does not call any more the organisational behaviour (in case of a failure in an action) and the reasoning module does not communicate with the operative part to ask it to execute actions. These decisions are under the responsibility of the user. This model enables the avatar to follow the evolution of the procedure and the choice of the users. Having this capability, the avatar of a learner can explain, give advice or show the realization of a task to the user.

2.5. MASCARET and Intelligent Tutoring Systems.

Intelligent Tutoring Systems (ITS) aim at providing students with dedicated tutoring. In our case we consider an ITS as a system allowing to communicate a "know-how" more than a system to communicate knowledge. An ITS can be composed of four components: model of the domain, model of the student, pedagogical model and model of the interface [9]. It usually uses only two of the fourth components (domain and student). The pedagogical and interface model is not often well specified.

MASCARET allows the establishment of models necessary to the creation of Intelligent Tutoring System. Concerning the model of the domain, we propose as [10] the use of the notion of procedure as describe in the social environment. Thanks to this formalism, an expert is able to provide such a model using semantics and temporal logic. There is no explicit modelling of the student in *MASCARET*. But the student model could be obtained using active detection of the student's actions and comparing their to the domain model. To do this, the avatar model proposed in *MASCARET* can be useful.

The pedagogical model could take decision using the model of the domain and the model of the student. This leads to pedagogical actions which can take several forms. First, we propose to use the physical environment of *MASCARET* in a pedagogical way. In fact, thanks to different levels of interaction network, physical phenomena could be adapted to the student level. For example, it is not necessary to display every disturbing element for a novice. As opposite, it is possible to add new physical disturbing phenomena for an experimented student in order to improve his skill. The physical environment of *MASCARET* also allows showing elements that are not visible in the real world. For example, we could display wind curve or gas cloud in order to inform to the student on specific conditions. In addition, the architecture of rational agents can supply to the user the action plan needed to realize a task. Using the notion of procedure as described in the social environment, the avatar can provide deductive reasoning and explications [11]. In fact, the ITS answers how and why something must be done during the exercise. The ITS can also reply to what should be done next and therefore a simulated tutor can show a demonstration of the next action in line. Finally, we propose to use *MASCARET* to define the organisation of pedagogical roles in a multi strategic pedagogical teaching. Our perspective is to integrate pedagogical actors, taking different roles. Therefore such actors have a common objective: to increase the student's skills.

3. SecuRéVi

SECUREVI (Security and Virtual Reality) is an application of *MASCARET* to civil safety. It helps with the training of firemen officers for operational management and for commandment. A picture taken from *SECUREVI* is shown on Fig. 4. This picture represents a typical scene in this VET and shows the level of details and immersion of this application.

Figure 4. Picture from *SÉCURÉVI.*

3.1. Implementation

To implement SecuReVi, we used the *AReVi/oRis* platform. *oRis* [12] is an environment for interactive simulation: it is a programming language founded upon active objects and an execution environment. Those characteristics make *oRis* a generic platform for multiagent systems implementation, more particularly dedicated to simulation. It is a dynamically interpreted language, with instance granularity which makes it possible to modify freely the course of simulation to observe the multiagent system, to interact with agents or on the environment and to modify them on line. In *oRis*, a multiagent system is compounded of agents (active objects) whose environment consists in objects, possibly located in space (2D or 3D) and time. *oRis* offers a homogeneous solution for interactions implemented by method calls, reflex or messages (peer to peer or broadcast, with immediate or differed processing by its recipient).

ARéVi is a virtual reality framework whose core is *oRis* and thus all the potentialities of this language are available when designing applications under *AReVi*. It is extended by C++ code offering suitable functionalities for virtual reality and it is founded on *OpenGL Optimizer*. Graphic objects are loaded from *VRML* files for which it is possible to define animations and manage level of details. Graphical elements such as transparent or animated textures, lights and *lens flares* are available. *ARéVi* introduces also kinematic functions. *ARéVi* proposes three-dimensional sound, synthesis and voice recognition functionalities. This framework manages various peripherals such as data glove, joystick, steering wheels, localization sensors and head mounted displays which extend the possibilities of user's immersion in the multiagent system.

3.2. Site modelling

From *MASCARET* model and *AReVi/oRis* framework, the first stage of the realization of *SecuReVi* is the modelling of sites. This task being tiresome, we take attention of reusability of elements already carried out. The result is the use of a standard representation format and the creation of a classification and a library.

All elements are located *i.e.* they have a position and an orientation. We chose to organize these elements, first to facilitate future development of tools for the sites designing and secondly to simplify the agent's behaviour. The criterion chosen to structure the elements is related to the capabilities of movement. From the located entities of *MASCARET*, we thus defined three classes of entities:

- The mobile entities (`GeoMobile`) have abilities to move in an autonomous way. They are characterized by criteria of mass, including box and kinematics attributes (speed, acceleration...). They are for example vehicles, humanoids and particles.
- The movable entities (`GeoMovable`) have the same characteristics as the preceding ones, but they do not have the capabilities to move alone. They undergo the forces of elements (mobile entities) on which they are dependent. They are the firemen's tools, some urban furniture and tanks.
- The static elements (`GeoStatic`) can move neither by themselves nor by the forces exerted by other entities. They have an infinite mass, but have an including box for the calculation of obstacle avoidance. They represent, for example, the buildings and large tanks.

Located entities take part, by default, in two interactions networks: a mobility network and a collide network. Mobile entities can then play any role in collision and mobility interactions. On the other hand, movable entities can play only a target role in mobility interactions whereas they can play any role in collision interactions. Lastly, static elements can play only a target role in collision interactions; they do not play any role in mobility interactions.

As for sensible rendering (visual and sounds), methods used in *SecuReVi* depend on constraints fixed by virtual environments for training such as credibility and the need of real time computation.

Users must recognize the environment, so it is impossible to use automatic methods for urban landscape generation. The method we use is founded upon the construction of geometrical models from plans provided by industrials and civil safety services. Within *SecuReVi*, the rebuilt environments are not very wide: in no case we want to reproduce a whole city, but only sites, or to a maximum a district. The quantity of data thus remains reasonable compared to the current available hardware resources and does not penalize real time calculation of the simulation. Fig. 5 shows the reconstitution of a factory site. This scene represents approximately 50 located entities for a total of 50 000 triangles.

Figure 5. Industrial site 3D reconstitution.

3.3. The physical environment

The second phase is the physical phenomena simulation. The physical environment consists in the site in which exercises take place as well as the physical phenomena (fire, smoke, jet of water...) which can there intervene. For *SecuReVi*, those phenomena must be as realistic as possible or exaggerated so that the learner feels immerged in the environment. *SecuReVi* is initially intended with outside environment such as industrial sites. The agents of the environment are then flames, gas particles, water jets… This paragraph emphasizes the realisation method of the interaction networks in *SecuReVi* by derivation of those defined in *MASCARET* as well as the agents taking part in these networks.

In *SecuReVi*, several types of interaction networks have to be developed, but most of them are defined according to the implementation of new scenarios. Once implemented, they feed a library and can be used again in other exercises. In the exercise of a gas leakage in a factory site, represented Fig. 6, the major phenomenon to take into account is the propagation of the gas cloud (displacement and collision) as well as the toxic effect which it can have on human beings.

Figure 6. Example of a physical phenomenon in *SecuReVi*.

In this example, three interaction networks exist. The first one is the mobility network; it is composed of the wind which is a source and a recruiter as well as the gas particles, which are only targets. The second network is the toxicity network in which humans take part by playing the roles of recruiter and target. The gas particles also take part in it by playing the target role. Finally the third network is the collision one in which the water walls and the "*tail of peacock*[1]" takes part as a source and a recruiter. The gas particles also take part in it by playing the source role. All these phenomena are interaction networks. Fig. 7 shows how the `CollideNet` network derives from `InteractionNet` of *MASCARET*. `CollideNet` has then the roles `Source`, `Target` and `Recruiting`, but defines also a `Collider` role. This role describes in its turn, a reactive behaviour `CollideBehavior`. Fig. 8 shows how collision is calculated for a gas bubble and water walls. This mechanism is used for all the interaction networks from the physical environment.

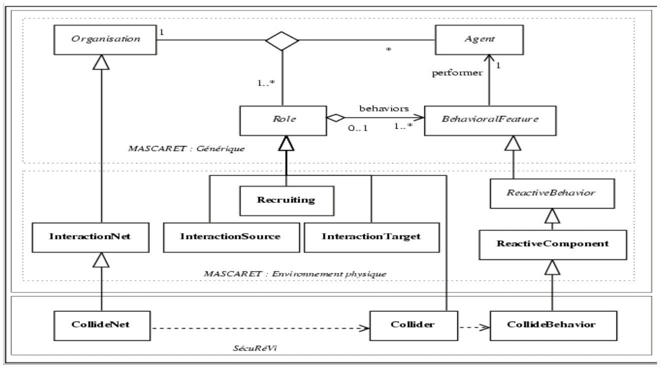

Figure 7. A collide interaction network.

[1] It is a firemen tool used to create a water wall. Its name comes from its shape (a tail of peacock).

In a teaching scenario we create each of these networks, but the role assignments to agents is done in a dynamic way. In the case of a gas leakage exercise in an industrial site (Fig. 4 and Fig. 6), only some agents are created at the beginning of the scenario. Thus, the wind starts playing the role `Source Recruiting` and `Mobile` at the beginning of the simulation in the `Mobility` network and the gas tanks play the roles of `Target` and `Recruiting` in `ThermicTransfert`. It is during the implementation of agents that the developer indicates the roles that each agent can play, because the assignment of the roles is static here. The water jets and the gas particles are created during simulation, and it is at this time that they start to take part into the interaction networks.

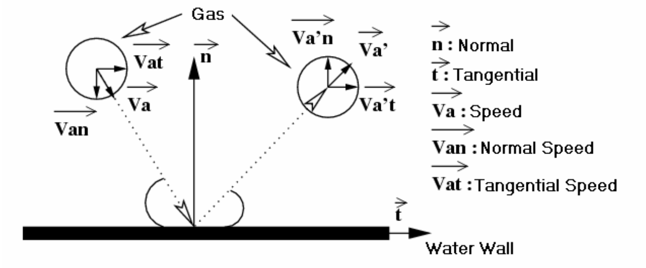

Fig 8. Collision calculous between gas and water wall.

Various techniques exist to represent the visual aspect of physical phenomena, but they are related to the model of phenomenon calculation which is often a global model. That makes them unusable in our case because these techniques do not allow user interactions. Moreover, some of them are not (or with difficulty) calculable in real time, which create a problem because we want to simulate several types of phenomenon in same simulation. Thus we cannot use these techniques in *SecuReVi*. Then we use systems with particles to represent the jets of water as well as the explosions, and billboards with transparent and animated textures to represent the gas particles and fires.

3.4. The autonomous actors

In *SecuReVi*, agents taking part in the social environment are humanoids. Characters are safety agents from the industrial sites and firemen. They are instances of agent's classes inheriting from `humanoid` class. This class describes traditional competences of humans such as walking to a point or to communicate. Designing new classes of agents from this one is justified by addition of new technical skills such as swimming for a plunger, and not by addition of responsibilities to an agent (represented by roles of organizations).

Various techniques exist for the reconstitution of humanoids as well as for their actions. We use the *Poser* software (in conformity with the *H-Anim* standard) to create characters and movements. Technically, in *SecuReVi*, each pose is defined in a VRML file. The platform that we use does not make it possible to reach VRML nodes, it is thus necessary to store in memory each pose. We count on the evolutions of the platform to solve this problem what will enable us to load only once the file and then reach the nodes to give them their new values according to pre-calculated animation. Fig. 9 shows two poses for one of the characters thus recreated.

Figure 9. Character's 3D reconstitution.

These characters inherit from *MASCARET* agents. So they are autonomous agents, able to realize procedures in collaboration with other members of their organization. These characters are also able to adapt such procedures to the dynamic environment and to calculate, autonomously, implicit plans to reach goals that experts did not explicit. For example, the first action of procedure n°16 of a FPT team (see next section for FPT explanation) is to get ready. In order to get ready, each agent needs to have its tools. So, agents compute (according to algorithm presented before) an implicit plan allowing them to have their tools. Fig 10 shows the unification and back chaining algorithm for this example. If for some reason (a mistake in the pedagogical scenario for example) an agent cannot find all its tools, it will detect the plan have failed and will ask other orders to its head-master.

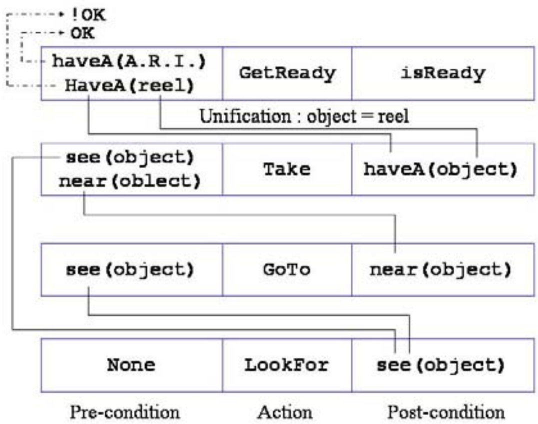

Figure 10. Example of implicit plan calculation.

3.5. The social environment.

Teams are groups such as the FPT[2], VSAB[3]... whose members are played by autonomous agents. *SecuReVi* is addressed to the head of group (Level N° 4 in firemen organisation) which orders from 2 to 4 groups. The goal of *SecuReVi* is to train to decision-making in operational situation, actions of agents thus must have a sufficiently realistic representation to support immersion of user, but not inevitably very precise.

In the case of a gas leakage on a factory site, the objective is to secure the populations, to reduce the propagation of the incident and to protect the tanks in case of ignition of the gas cloud. For that the intervention plan (PPI) envisages the engagement of several teams. The first committed team is the ICMC[4] team whose role is to acquire information on involved products and on their propagation. The plan also envisages the engagement of several FPT teams whose objective is to slow down the progression of the gas cloud using water walls and to protect tanks by sprinkling them. This requires water resources and material (water pipes, reels...) therefore the intervention of a reel cell is also designed to provide these resources. All these teams are made up from three to five members and for each of them there is a handbook describing the whole of the procedures which it can be led to carry out. All these teams derive from the models defined in *MASCARET*, as Fig. 11 shows for the example of a FPT team. Thus, in this example, the FPT team inherits the `TEAM` class of *MASCARET*. This creates three team roles `Head-Master`, `Second Head` and `Helper` inheriting `TeamRole` and defining actions such as `GetReady` and `ManipulateReel` inheriting `GDAction`. The work of the designer in *SecuReVi* is thus primarily to implement these elements while inheriting *MASCARET*.

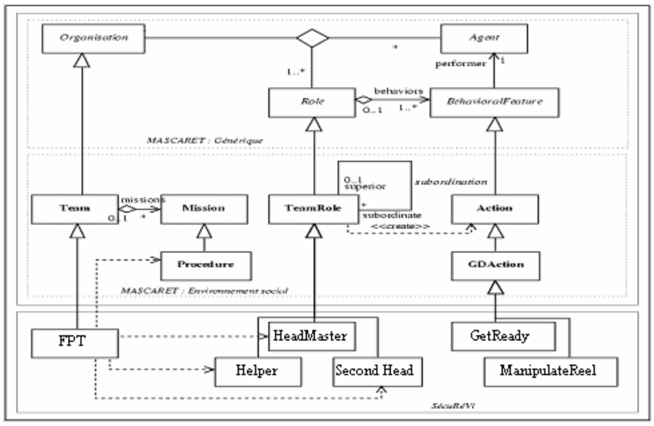

Figure 11. Team reconstitution.

---

[2] Fire engine Thunders Pumps
[3] Vehicle for wounded persons
[4] Intervention Chemical Movable Cell

According to the exercise's scenario, these organisational structures can be created at the beginning of a simulation or in a dynamic way. Organisations are then FPT or CMIC teams (team FPT n°1, team FPT n°2, team CMIC n°1...). For each team instance, the roles (Head, Second Head...) are also instantiated. On the other hand, in the case of firemen, an agent assignment to roles is described in the scenario, each agent knows, from the beginning, in which instance of team it plays and which role it plays: it is the real operating mode of firemen. For each agent taking part in a team, actions defined by roles are then created. Teams also define missions described in firemen teaching documents. Each of these missions is instance of `Procedure`, they are instantiated during the creation of teams, but the transfer in agent's knowledge is only effective during the execution of one of them.

3.6. Pedagogical use

Learners play the roles of various heads of groups intervening during the incident and trainer takes part in simulation to cause dysfunctions, to help learners or to play a role in a team. Learners must then follow an intervention plan, order the realization of procedures to groups and bring back the situation to their chief. The pedagogical use of *SecuReVi* is not yet completely formalized, but there is undoubtedly a phase of design of elements being able to intervene in exercises, a phase of design of scenarios, simulation and debriefing. Three roles (expert, trainer and learner) are involved in the teaching use of *SecuReVi* which proceeds on four phases (Fig. 12).

The first phase is the design of elements which can play a part in exercises. It is a question of conceiving physical phenomena and teams of firemen; it is the stage which we previously presented for the exercise of gas leakage. This phase needs an expert (or a group of experts) having competences in model *MASCARET*, programming and the specific model field: chemists, physicists or meteorologists for the physical phenomena and officers firemen for the operating teams.

The second phase is carried out by the trainer, according to elements provided by experts (organisational structures and types of agent), it describes exercises or scenarios. It is the instantiation of organisational structures and assignment of roles to agents. A scenario also includes situations by which the trainer wants that learner passes, these situations are represented by events which can be broadcasted in the environment or be addressed explicitly to one of its agents so that it causes this specific situation.

The two last phases are simulation and debriefing; they involved the trainer and learners. During this phase all the exchanged actions and all messages are dated and recorded to be useful during the debriefing. This last phase was not formalized, but the traditional functionalities awaited during this phase are the capacity to *re-play* the simulation.

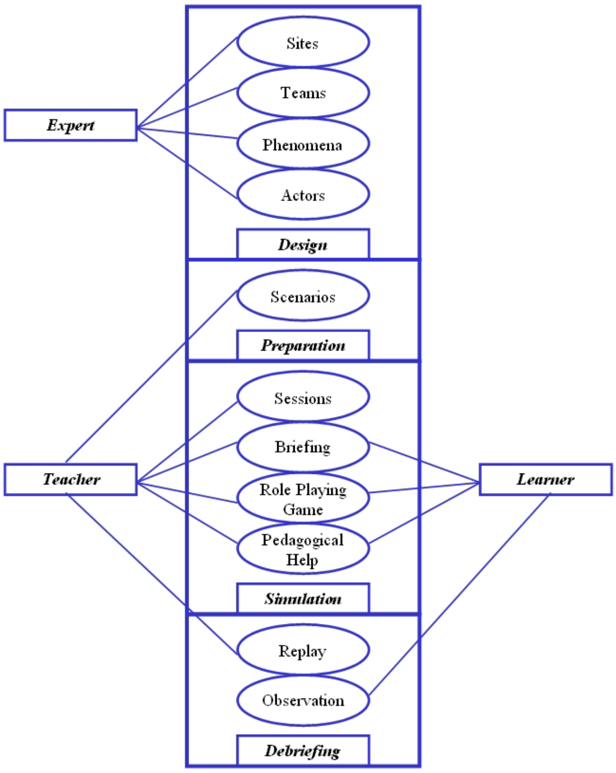

Figure 12. Pedagogical use case packages.

4. Conclusion and future works

Our objective is to place learners in operational conditions in their simulated physical and social environment. Considering a virtual environment for training as a multiagents system, we propose the *MASCARET* model. It allows the realization of a cooperative, adaptive and realistic virtual environment for training. The *SECUREVI* application is intended for the training of firemen officers and is based on this model. In *MASCARET*, classical pedagogical functionalities expected in a virtual environment for training have not been yet formalized. The study of different training environments showed that *MASCARET* allows the implementation of the functions present in existing environments and offers mechanisms allowing them to be improved.

Thus, we wish to integrate the notion of "pedagogical program" proposed by pedagogues [13]. It will be built around three phases: 1) the verbalization, the learner is able to explain the objectives of the formation, 2) the transfer, the learner is able to abstract concepts present in the training and to transfer them to another context and 3) the reinforcement, the learner effectuates series of exercises aiming to create automatisms on the subject of the training. Our contribution will concern the realization of agent models allowing us to automatically distinguish these three phases in the learner program and to generate adequate exercises.

In order to improve the learner's appropriation of his role in the exercise, we also propose the integration of the *"putting into operation"* and of *"pedagogical scenario"* [14] notion. In that way the learner will feel immersed in the context of the exercise and decides by himself to play a role. Furthermore, virtual reality and multi-agents systems allow the simulation of different elements (characters, physical phenomena…) allowing triggering learner emotions that are an important factor in pedagogy. Thus every exercise of the « program » begins by a phase of putting into operation of a pedagogical context.

Our work is addressed to professional training, where the teacher is an expert of the domain but not necessarily a pedagogue. Moreover several students are involved in the exercise, thus the teacher can't consider each learner separately. Therefore, he can't make dedicated tutoring considering all learner profiles. We propose to the human teacher, means the human expert, to specify different pedagogical agent behaviours (companion, troublemaker ....). Their goal is to help the teacher to adapt tutoring to each learner. For that, we propose to model pedagogical strategies by means of pedagogical actors. In this framework, the strategy of the critic, counselor, guardian, companion [15] and troublemaker [16] seems relevant. In the co-operative strategy, the companion is a virtual actor that will cooperate for the realization of tasks, exchange ideas on the problem and share the same goals. In the troublemaker strategy the goal of pedagogical actors is to disturb the learner by proposing solutions that can sometimes be erroneous. That way, we force the learner to evaluate his self-confidence in his own solutions. Such strategies are not simple to specify in a generic case. Thus, we wish to endow such agents with the capacity of learning by imitation [17] or by the example [18]. The human teacher will take control of the pedagogical agents during the preparation phase of the exercise and they will learn a behavior adapted to the situation. If it is considered we are in possession of such different pedagogical agents, an ITS will be multi strategic [19].

Acknowledgements

The authors would like to thanks the services for fire rescue of *Finistère* for its help and support to this project. We also want to thanks The National Institute of Civil Safety and the Ecole des Mines of Alès for their contribution. This work has been partially granted by the CUB (Urban Community of Brest).

References

[1] J. Piaget, *Le comportement, moteur de l'évolution.* (Gallimard, Paris, 1976).

[2] P. Fuchs, G. Moreau and Papin, J. *Le traité de la réalité virtuelle.* (Les Presses de l'Ecole des Mines, Paris, 2001).

[3] G. Burdea, and P. Coiffet, *La réalité virtuelle.* (Hermès, Paris, 2003).

[4] J. Tisseau, *Virtual Reality : in virtuo autonomy,* Accreditation to Direct Research, University of Rennes 1, 2001.

[5] Y. Demazeau, From interactions to collective behaviour in agent-based systems. *Proc. European Conference on Cognitive Science*, 1995, 117-132.

[6] M. Hannoun, O. Boissier, J. Sichman, and C. Sayettat, MOISE: An Organizational Model for Multi-agent Systems. *Proc. IBERAMIA-SBIA 2000*. 2000, 156-165.

[7] J. Ferber and O. Gutknecht, A meta-model for the analysis and design of organizations in multi-agent systems. *Proc. ICMAS'98: 3rd International Conference on Multi-Agent Systems*. 1998, 128-135.

[8] P. Deloor and P. Chevaillier, Generation of agent interactions from temporal logic specifications. *IMACS'00*, 2000.

[9] B.P. Woolf, Building knowledge based tutors. *Computer Assisted Learning: Proc. of the 4th International Conference (ICCAL'92)*, 1992, 46-60. Springer, Berlin.

[10] D. Lourdeaux, P. Fuchs, J.-M. Burkhardt, An intelligent tutorial agent for training virtual environments. *5th world multiconference on Systemics, Cybernetics and Informatics SCI'01*, 2001.

[11] J. Rickel, and L. Johnson, Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*, 13, 1999, 343 – 382.

[12] F. Harrouet, *oRis : s'immerger par le langage pour le prototypage d'univers virtuels à base d'entités autonomes.* PhD, Université de Bretagne Occidentale, 2000.

[13] J.F. Richard, *Les activités mentales. Comprendre, raisonner, trouver des solutions*. (Armand Colin, Paris, 1998).

[14] M. Crampes and G. Saussac. Facteurs qualité et composantes de scénario pour la conception de simulateurs pédagogiques à vocation comportementale. *Sciences et Technique Educatives,* 6(1), 1999, 11-36.

[15] T.W. Chan and A.B. Baskin, Learning Compagnion Systems., Intelligent Tutoring Systems: *At the Crossroad of Artificial Intelligence and Education*, 1 (C. Frasson & Gauthier Eds. 2000) 159-167.

[16] E. Aimeur, C. Frasson, and H. Dufort, Cooperative learning strategies for intelligent tutoring systems. *Applied Artificial Intelligence* 14, 2000, 465 – 489.

[17] P. Gaussier, S. Moga and M. Quoy, From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Proc. Man and Cybernetics volume 31*, 2001, 431-442.

[18]. A. Del Bimbo and E. Vicario, Specification by-example of virtual agents behavior. *Proc. IEEE Transactions on Visualization and Computer Graphics*, 1, 1995.

[19] T. Mengelle, and C. Frasson, A multi-Agent Architecture for an ITS with multiple strategies, *Proc. Conf. CALICE'96*, LNCS 1108, 1996, 96-104.

Biographies

Ronan Querrec is Professor assistant of Computer Science at the National School of Engineer of Brest. He is involved in the CERV (Virtual Reality European Centre held at Brest, France) researches. The projects of the laboratory deal with the modelling of mono and multiagent behaviours and the designing of group wares allowing the cooperation in 3D universes. He passed his PhD in 2002 on Virtual Environment for Training. His current research is based on the idea that those environments are heterogeneous and open multiagent systems where humans' users participate.

Cédric Buche received French Graduate Engineering Degree in July 2002. He is a PhD student in Computer Science at the CERV. He integrated the Immersive and Participative Simulation project of LI2 in September 2002. The objective of the project is to simulate complex systems integrating humans; these humans being spectators, actors and creators. His PhD subject concerns to design of agent model integrating adaptive pedagogical behaviour in order to provide dedicated learning to student in a virtual environment for training. His research topics are multi-agent systems, artificial intelligence, autonomous behaviours, fuzzy cognitive maps, soft computing and machine learning techniques.

Eric Maffre received French Graduate Engineering Degree in July 2001. He is in his second year of PhD in Computer Science at the CERV. His PhD subject is to define pedagogical organization using the *MASCARET* model in a virtual environment for training. Application is a simulator allowing training people to maintain military vehicles. His research topics are multi-agent systems and autonomous behaviours.

Pierre Chevaillier is a professor assistant in Computer Science and works at the CERV. He works on the use of the multiagent paradigm to design digital simulations where users have to play an active and collaborative role. He is the leader of the *MASCARET* project and focuses its research activities on the interaction and organisational facets of heterogeneous mutiagents systems (where both artificial and human agents are involved).