

Modelling Autonomous Virtual Agent Behaviours in a Virtual Environment for Risk



Lydie Edward¹, Domitile Lourdeaux¹, Jean-Paul Barthès¹, Dominique Lenne¹ and Jean-Marie Burkhardt²

¹ University of Technology of Compiègne, Compiègne, France.

² Cognitive Ergonomics at ECI lab (Ergonomic Behaviour and Interactions) of Paris Descartes University, France.

Abstract—Our research deals with the design of a training system to support decision-making in the preparation and the management of maintenance interventions in high-risk industries namely SEVESO sites. The proposed system incorporates virtual reality and artificial intelligence to simulate virtual autonomous characters and their cognitive processes in dangerous working situations. It generates behaviour-based errors to support learning and risk prevention. It uses new mechanisms taking into account human factors with respect to cognitive modelling of human behaviour regarding risky situations. In the simulated environment the trainee can visualize the risks incurred during his work with the virtual agents. The emergent risks depend on the cognitive characteristics of the virtual operators and on the expertise of the trainee. We propose a multi-agent system to support the control of virtual operators represented by virtual cognitive agents. The difference with a classic MAS is that our cognitive agents are enriched with a planner for selecting actions according to goals, the environment and to the personal characteristics of the agents (time pressure, caution, tiredness, hunger).

Index Terms—Cognitive planning, intelligent virtual environment, multi-agent system, risk prevention.

I. INTRODUCTION

1.1. Research context

The combination of virtual reality with multi-agent systems gives the opportunity to offer virtual environments for improving training and decision-making with avatars and virtual autonomous agents in hazardous surroundings, on high-risk so-called SEVESO plant. The context of our research is to model collaborative and cooperative work to improve training, decision-making and risk avoidance and to show the results of the activity in a hazardous environment [1]. As a supplementary benefit, these combination enable introduce unexpected variations so that training scenarios are no longer predefined and ideal. This feature is particularly interesting when training objectives are not only to acquire the correct procedure, but also to highlight the difficulty of cooperative work and the constraints linked to hazardous professional activities. This paper addresses the design and the implementation of a system to simulate autonomous behaviour of virtual operators interacting with real operators in a virtual

environment for training/learning. The system aims to take into account various situational constraints and human-factors issues to support the generation of the subsequent variations in the way virtual operators are behaving and applying the procedures to achieve their task. Such constraints could be the result of the combination of cognitive and physical parameters such as tiredness, stress, expertise, hunger, motivation, etc. They can also come from time pressure, missing tools, etc. Taking these human factors into account we are able to simulate erroneous behaviours and some deviations. These deviations could be: errors, failure, indiscipline and botching.

Virtual reality is useful for training people but systems based only on this technology are rather reductive and deterministic. We want to let our virtual characters react to the changes of the environment, take their own decisions and work together with the real operator. We want them to react in an autonomous way. We therefore propose coupling virtual reality with multi-agent systems.

Our work is based on cognitive modelling, errors modelling, multi-agent planning and activity modelling. We propose an ontological and knowledge-based model to describe the operator's activity (HAWAI-DL²), a related risk model, a model of the environment (COLOMBO³) and a decision-maker module based on a multi-agent system (MASVER⁴) to interpret the models and to simulate the operators' behaviour. A challenge is to develop an intelligent virtual environment that will be connected to MASVERP that will support the control of virtual operators represented by cognitive agents.

1.2. Our Assumptions

Our work is based on two assumptions: (i) there is a need for such systems; and (ii) such systems can be implemented with MAS.

The first assumption is that virtual reality can improve safety training for industrial interventions [2]. Real working situations on SEVESO site can be costly and dangerous; therefore virtual reality will be a knowledge and know-how cheaper vector. The potentials offered by VR are offering new perspectives to training: virtual presentation of the working situation, of the installation, the tools function, learning of how to detect the defaults while playing a scenario, etc. Indeed, virtual environment might favour learning by doing since the user is in constant interactivity with the environment (perception-

Manuscript received on April 4th, 2008.

E-Mail: Lydie.edward@utc.fr.

This work has been partially funded by ANR under the V3S (Virtual Reality for Safe SEVESO Interventions) project.

² Human Activity and Work Analysis for sImulation-Description Language.

³ Ontological Creation Linked to the Organization of the OBjects Modelling-French translation.

⁴ Multi-Agent System for Virtual Environment for Risk Prevention.

decision-action loop) [3, 4, 5]. Training and decision making related to safety management can be improved when operators and managers see the impact of their decisions and actions on the technical and human system they have in charge.

Our second assumption is that such an environment can be fruitfully implemented by a multi-agent system. We propose a new system, MASVERP [1] that endows the environment (virtual characters and objects) with autonomous decisional capacities. Three different entities share the same environment: (i) virtual characters represented by autonomous cognitive agents; able to make decisions using a planner that interprets the high level task model; (ii) the behavioural objects of the environment represented by reactive agents and finally (iii) an avatar representing the trainee.

In our project, our goal is neither to reproduce perfect cognitive mechanisms nor to describe the operator's cognition.

Instead, we aim to emulate the behaviours that could occur with a reasonable level of faithfulness dynamically, according to a cognitive model representing the evolution of the cognitive state of an operator and the incurred risk. This approach requires merging three scientific domains as shown Fig. 1.

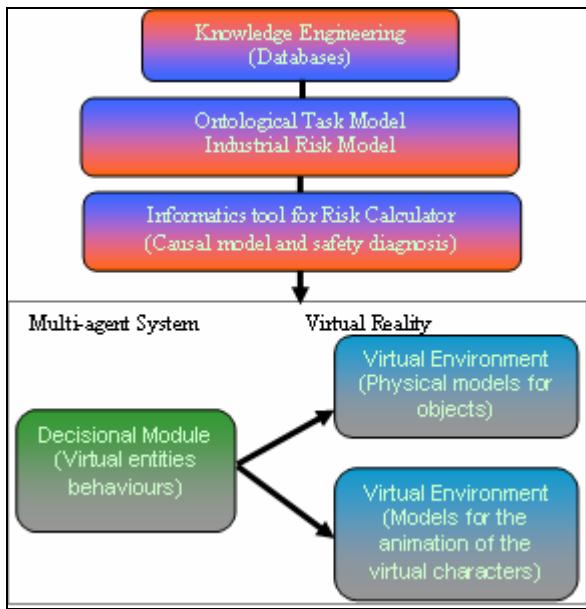


Fig. 1. Coupling knowledge engineering and virtual reality.

We merge (i) knowledge engineering databases to provide an industrial risk model and an ontological task model both issued from field analysis, (ii) multi-agent systems and (iii) virtual reality. In this article we focus only on the aspects related to the knowledge and decisional module. We do not consider the mere design of the 3D virtual environment, the animation and visual rendering of the virtual characters.

We present the language HAWAI-DL specified to support the description of human activity, the decisional architecture of our system, the multi-agent systems MASVERP and methods to augment the virtual agent behaviour with AI structures that incorporate among other things behaviours, knowledge, environment, and characteristics models. We present our decision algorithm taking into account cognitive aspects and we finally present some preliminary results and the virtual

environment for risk prevention.

II. RELATED WORK

2.1 Behaviour modelling and cognitive structures in characters animation

Dimitri Terzopoulos and Wei Shao [6] developed a system representing intelligent virtual pedestrians evolving in a virtual urban environment. Their algorithms aims at emulating not only appearance, locomotion, and behaviour of each intelligent character but also at generating complex collective behaviours and actions as can be observed in urban spaces. Their system is based on cognitive modelling [7] and a hierarchy of high data structures integrating motor, perceptual, behavioural, and cognitive components. Each pedestrian has a stack of goals set by the behavioural and cognitive control components. Each one also has an action selection mechanism that determines the action to implement according to the mental state of the pedestrian (physiological and psychological social needs: tiredness, thirst, curiosity). The value of the mental state matches a specific behaviour.

Along the same line, IRISA developed a crowd simulator based on studies of pedestrian behaviours (Virtual Urban Environment System) [8]. The behavioural model of pedestrians includes social and driving rules of interaction based on Gibson affordance theory [9] and on the perception model proposed by Badler [10]. They proposed the HPTS++ language inspired by a cognitive theory [11] to model their autonomous agents organised in a hierarchy of automata.

In the field of training tools using virtual reality and human behaviour modelling, Jeff Rickel and Lewis Johnson designed STEVE, an autonomous animated agent that lives in a virtual world and interacts with students [12, 13]. It has been designed to help students in learning to perform procedural tasks. STEVE has a cognitive mono-agent architecture based on SOAR [14], which allows it to know the state of the environment in real time and to decide what actions to undertake. STEVE's technologies have been further used for leadership training in Virtual Bosnia [15, 16].

2.2 Multi-agent systems combined with VR in VET/L

A good overview of the combination of virtual reality and multi-agent systems can be also found in [17] and in [18]. The GRIC-GRAAL [17] group of researchers developed a training tool for firemen. It aims at keeping them out of danger. The architecture is a multi-agent system composed of emotional and reactive agents. It is based on a personality model (Five-Factor Model) and on an advanced emotional ontological model. Agents have a goal to achieve and use a Prolog planner to determine their actions. In the MASCARET project [18], the physical environment represents a plant where an exercise takes place including physical phenomena that can appear on the plant (fire, smoke, water spreading). The trainees play the role of the different group managers who intervene during an incident and the trainer participates to the simulation as a troublemaker. The system is driven by the MASCARET model proposed for organizing the interactions between agents (give them reactive and social abilities).

2.3 Positioning our work

Our approach differs from the previous work in the following way. Current approaches rely mostly on theoretical architectures (automata, Petri networks, expert systems). Only a few works propose to build systems taking into account cognitive behaviour model and error models for virtual human animation [6, 7, 8]. In our work, taking into account human factors is essential. We based our research on cognitive models in the domain of safety and human behaviour in risky situations [19, 20, 21]; from the models we propose new mechanisms using multi-agent system to represent human decisional process and human behaviour-based errors to finally simulate them in a virtual environment. To achieve our goal we developed a generic language that describes the goal-decomposition of an intervention and integrates some of its related cognitive and safety aspects.

III. COGNITIVE AND ACTIVITY MODEL

In the virtual environment the objective is to simulate the agents' operating mode in both normal and constrained work conditions. These situational constraints could have three origins: (i) physical dimensions of the environment like difficulties related to the geometry of the site, operator morphology, coldness, or wind; (ii) organizational dimension, prescribed procedure; and (iii) cognitive and mental characteristics of the virtual operators like objectives, time pressure, cautiousness, tiredness, stress, expertise, etc. Cognitive planning lets us to simulate some of the possible deviations of virtual agent's behaviours while interacting with the environment and with the other operators.

3.1. The COCOM model

Depending on the situational constraints describe previously, our virtual operators have different behaviours. We rely on a model proposed by Hollnagel [21] the COCOM model. The model enables to describe, what he called, the control mode of an operator that depends on the time pressure. Hollnagel defines four types of control modes associated to time zones in which an agent can operate. (i) In the strategic control mode, the agent has a wide time horizon and looks ahead at higher-level goals. He has a both large and detailed anticipation of the work system. (ii) The tactical control mode characterises situations where performance more or less follows a known procedure or rule. The user's time horizon goes somewhat beyond the dominant needs of the present, but planning is of limited range. He often chooses the simplest situation and can therefore not respect the safety constraints. (iii) In the opportunistic control mode, the next action reflects the salient features of the current context. Only little planning or anticipation is involved, perhaps because the context is not clearly understood by the agent or because the situation is chaotic. Opportunistic control is a heuristic that is applied when the knowledge mismatch is large, either due to inexperience, lack of formal knowledge, or an unusual state of the environment. (iv) In the scrambled control mode, the next action is in practice unpredictable or random. Such a performance is typically the case when people act in panic,

when cognition is effectively paralyzed and there is accordingly little or no correspondence between the situation and the actions. Thus depending on the control mode, the operator will plan broadly and choose the actions more adapted to the situation or plan to a more limited degree and compromise on safety aspects to gain productivity.

3.2. Border-line tolerated conditions of use (BTCU)

To characterize the deviations eventually operated by virtual operators we were inspired by the notion of border-line tolerated conditions of use (BTCU) from studies in ergonomics and human reliability. This notion highlights the individual and social regulations operated on the field which bring the use conditions of the tools and the realization mode of the task to some compromise zones affecting safety [22]. For example, some tasks are partially or not at all done because of a lack of time due to the compromises made between safety and production. This concept is a complement of others elements linked to the individual, like those reported to the consciousness of the risk, the tiredness effects or the time pressure on the performance, etc. It is a characteristic of the task with safety implications as "allowed to be performed", "allowed not to be performed", or "not allowed". If a task is considered as a BTCU, the agent will decide to perform it depending on its control mode and its characteristics (e.g. if it is in a hurry).

3.3. Description of the human/operators activity

We took our inspiration from studies on human activity in natural situations like those conducted in ergonomics or psychology domain. Researchers in ergonomic, generally talk about task and/or activity models to refer to models constructed on the basis of data collected by ergonomics analysis of work and activity. The various existing analysis methods are sharing the same goal which is to provide an efficient model to describe the specific exigencies, strategies and modes of operation related to any vocational activity. Among the usual formalisms used to figure out the results of work analysis, some are well-adapted, generic and have the interesting ability of taking the objectives, resources and working process of a subject in relation with his complex environment into account. Such methods are focused on the task that can be decomposed in subtasks, operations or actions.

We can distinguish (i) HTA (Hierarchical Task Analysis), GTA (Groupware Task Analysis), METISSE (Tasks Description Model to Assist and Track the Trainee- French translation) or MAD* (Analytic Method for Describing users TaskS orienTed of inteRface specificAtion-French translation-). GTA is specifically designed to model collective tasks. MAD* is more focused on the activity of an individual even if the latest version tends to integrate a collective dimension.

We propose a description language derived from MAD and METISSE formalisms [23], HAWAI-DL. It supports (i) the integration of factors affecting the performance at a collective level; (ii) the expression of the operator's activity as it should be; (iii) how the activity may change in deteriorated situations (lack of time, imprudent behaviour, safety behaviour, tiredness, etc.). Our language takes into account the safety characteristics

linked to risky activities. HAWAI-DL provides a hierarchical description of the activity, allowing activities corresponding to the procedure descriptions provided by experts during interviews with ergonomics researchers to be represented. In addition, this description is extended by the variances and the errors exposed in interviewing the operators on the plant, and/or the observations and the analyses obtained by the experts in ergonomic. Our model allows taking into account tasks that are hierarchically independent (i.e. that have no hierarchical relations with the main task). We call this kind of tasks “hyperonymic tasks”. For example it can be a task that indicates how to manage a leak. Our model also allows taking into account exceptional tasks that are triggered by exceptional events such as a leak or a fire. In addition, we integrate into this model the conditions under which an activity should be done, such as BTCU and their associated risks.

A task in HAWAI-DL is described by several attributes among which the following ones are directly relevant for our purpose:

- Name of the task.
- Goal: the expected state of the world after completing the task.
- Constructor: SEQ (sequential), PAR (parallel), AND, OR, SIM (simultaneous), as well as ALT (alternative).
- Complex : if the task has subtasks or not
- Iterative: if a task is repeated several times, depending on its duration or its frequency.
- Focus: if a sub-task is the main task in the hierarchical decomposition of a task.
- Dangerous: if a task is dangerous or not.
- Interruptible: if a task is interruptible or not.
- Optional: if a task is optional or not.
- Priority: a task can have a high, medium, or low priority.
- Safety-related task: if a task with safety implications is “allowed to be performed”, “allowed not to be performed”, or “not allowed”.
- BTCU : if the task have safety implications
- Agents : agents and roles that can performed the task

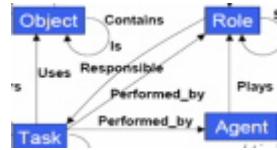


Fig. 2. Role, task, and agent relations in GTA.

HAWAI-DL allows a detailed description of tasks. It goes beyond the listing of the pre-conditions and post-conditions of a task to a more detailed description, in particular for the pre-conditions where we define:

- Mandatory pre-conditions: state of the world required for the task to be performed.
- Regulatory pre-conditions: state of the world required by safety regulations.
- Resources: tools that are required to accomplish the task.
- Triggering pre-conditions: events that require immediate treatment for a specific task (triggered task).

- Favourable pre-conditions: the context in which the task is relevant, for example:

- A safety-related task: for the sake of safety.
- A time-related task: if there is enough time, it is better to perform it.
- An expertise-related task: is a task that can be performed either by experts or by inexperienced.
- An environment-related task: is an alternative task that depends on the environment conditions. For example, there are two ways to remove a pipe. The environment conditions can be: bolts are not rusted. If the bolts are rusted we prefer to do the task with the good environment conditions.

HAWAI-DL represents the impact of a task on the environment as:

- Post-conditions: state of objects expected after performing a task, and/or results after completing the task.
- Perceptive feedback: feedback effects of the task on the environment.

In addition to the previous conditions, we define, for iterative tasks:

- Stop conditions: the task is repeated until we reach a predefined state (by default, the outcomes of the task). For example, these may be defined as duration, a specific number of iterations, or a number of instances (depending on the current scenario) of the task.

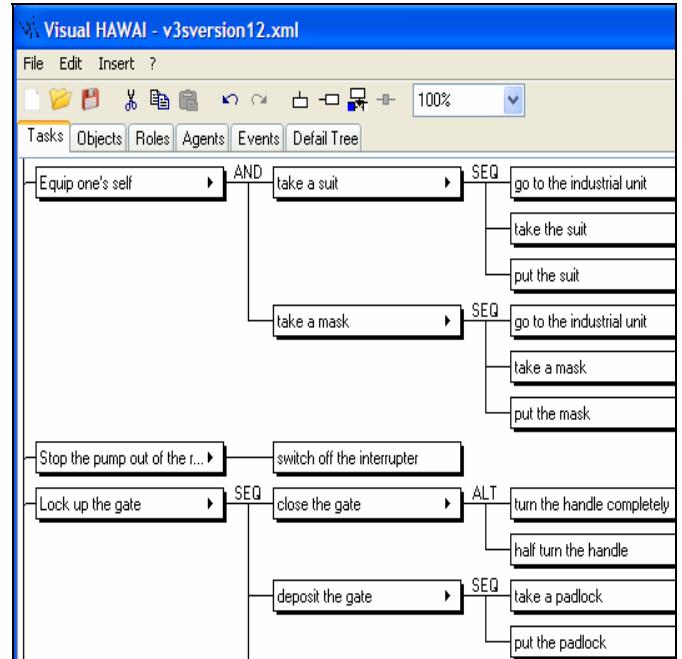


Fig. 3. Sample of a task model with HAWAI-DL.

IV. OUR ARCHITECTURE

4.1 Overall Architecture

Fig. 4 presents the architecture of our system on the decisional side. The multi-agent system loads the activity

model, the risk model and the world model. The activity and the world model are traduced into ontological models and are stored in the agent environment and in COLOMBO environment. The risk model is composed with decisional rules. The world model contains the instance of all objects present in the virtual environment, including their state and their position. COLOMBO sets up the state of the objects whenever something appends in the environment or after an action decided by MASVERP or by the real character, and then updates the state of the world in real time. MASVERP can access the state of the world by sending request messages to COLOMBO.

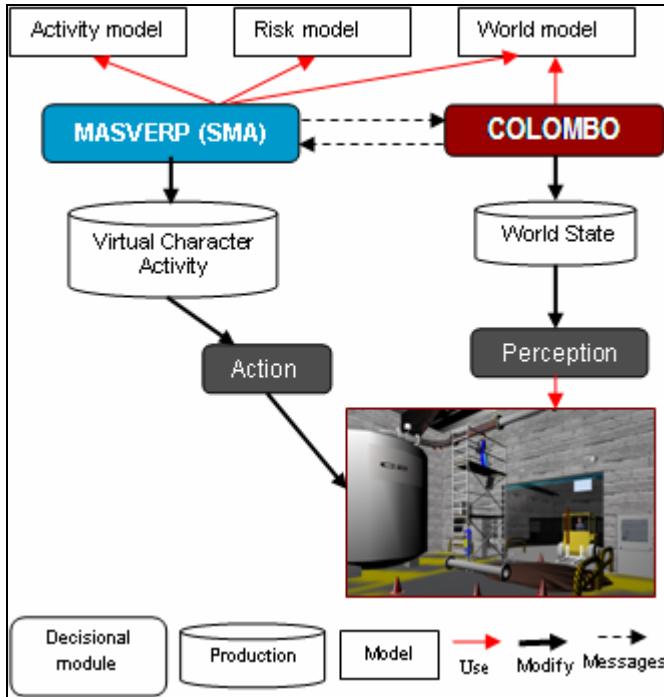


Fig. 4. Overall architecture: interaction between modules.

4.2 MASVERP agents

To remind our objectives, we want the virtual operators to adapt to the environment in an autonomous way, to respond and react to the manager's orders and to cooperate with the human operators. We assume that a MAS is a promising solution to reach this goal in terms of organisation, cooperation and planning [24] and we propose MASVERP in order to model the decisional module of the virtual agents and to give them the required autonomous abilities. The cognitive entities representing the virtual operators are complex and based on the BDI (Belief, Desire, Intention) model [25]. The agents are provided with a capacity of planning according to a high level cognitive activity model in the following way:

- Interpret the task model
- Generate a plan (sequence of tasks)
- Adapt to the environment and do plan repair

The agent evolves in an open and dynamic environment, and obviously it is a complex system. The agent reacts to any possible unexpected events considered as relevant to risk

prevention like e.g. a fire, a leak or any other incident that could happen. He must achieve the assigned task and find a compromise whenever possible to cope with simultaneous (and sometime incompatible) goals.

- Present a consistent behaviour

Substituting a virtual autonomous character to a human operator requires obtaining an operational behaviour in order to represent what a real operator could do.

We can see, as shown Fig. 5, that the cognitive agents comprise: (i) skills: in this part we specify what the agents can do and how they can do it; (ii) a list of goals: i.e. what they have to do. They also have an updater to refresh their goals, for example, if during the exercise they are thirsty their goal will be to quench their thirst; (iii) knowledge: what they know about the environment and also what they have learnt; (iv) environment: this part contains the environment variables and state; (v) address book: all the acquaintances of the agent, whom he knows and whom he can interact with; (vi) memory: this field regroups all the agent internal states and his characteristics, it could be progressive (pg) or permanent (pm) characteristics as cautiousness (pm), tiredness (pg), temporal pressure (pg), expertise (pm); they are used to determinate in real time the behaviour (control mode) adopted by the agent; the model of the activity is stored in the agent memory; (vii) a planner which produces their decisions i.e. their (viii) activity.

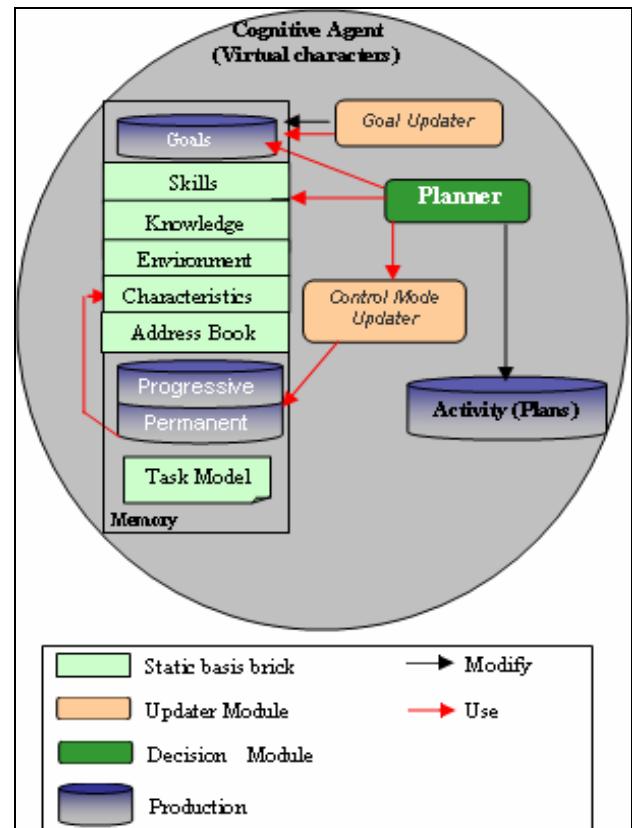


Fig. 5. MASVERP cognitive agents.

4.3 MASVERP Planner

The planning problem is at the intersection of two

domains: cognitive modelling and planning (robotics). Cognitive modelling is an area of computer science that deals with simulating human problem solving and mental task processes in a computerized model. Such a model can be used to simulate or predict human behaviour or performance on modelled tasks. Planning is used to produce action sequences or so-called plans to achieve a goal or to realize a complex task. It is used in robotics and commonly in multi-agent systems. Although the two approaches are quite different we propose to merge them into what we call the “cognitive planning agent”. We distinguish two phases of planning, (i) a provisional planning or pre-planning and (ii) a real synchronous planning during which it can be necessary to re-plan or to repair. The first phase imposes building a flexible plan allowing the adaptation to unexpected events. At the end of the pre-planning phase we have a provisional flexible plan and after the execution we have an effective adapted plan. The pre-planning is also useful to the operators to anticipate the resources they will need. Indeed, in the real activity, the operators prepare their intervention and define a number of tools depending on their mission and on the possible unexpected situations. The planner has the following entries: (i) the activity model of the operator used to specify the intervention, (ii) a risk model presenting the possible risks incurred during the intervention, (iii) a scenario and a world model containing the scenario data and the state of the different objects and resources of the environment, (iv) the characteristics of the agent taken into account to define the associated control mode in real time and finally (v) the goals. The planner provides the operators activity (i.e. plans).

V. IMPLEMENTATION

5.1 Multi-Agent platform

In MASVERP, an agent is built from an OMAS⁵ agent template. It has several files representing its environment and containing its skills. With OMAS the user can define agents, give them skills and goals (programmed as Lisp functions), then run them. Platform options allow tracing agent behaviour or messages. In addition, OMAS offers an advanced model of an assistant agent and allows creating local coteries. A coterie is the agent organisation; every agent belonging to the same coterie can communicate, receive and view all messages.

5.2 Computing the control mode

We implement the COCOM model adding physical and cognitive parameters to compute the control mode. The control mode determines how the agents coordinate their actions and select the task to execute. To compute the control mode of our cognitive agents we add several states of an agent. We also add a Knock out (K.O) mode which is reached when the agent is not able to do any task. We take into consideration (1) hunger, thirst, physical tiredness to determine the motivation; (2) cognitive load and tiredness to determine the vigilance, (3) Stress, motivation to determine the agitation; (4) Motivation,

agitation and stress to determine the control mode.

5.3 Planning Algorithms

In the mode strategic, tactic and opportunistic the agent will do the pre-planning for the resources. To select the action, we develop different algorithms depending on the control mode. Below, we present the algorithm used to cover the tree of tasks.

In the first step of the *Agent-Planning* algorithm (Fig. 6) we determine if the task conditions (mandatory, favourable, regulatory and BTCU) are activated. The mandatory conditions are the state of the world required for the task to be performed, therefore if they are false, the task cannot be achieved. The function that checks the mandatory and resources conditions is called *Solve-X-Task-Conditions*, because we are trying to make the conditions true. We evaluate the mandatory conditions (*Solve-Mandatory-Task-Conditions*). If it is not true, the *Apply-Solving-Method* looks for a task in the task tree and in the hyperonymic tasks that have for post-conditions the mandatory conditions in parameters. If the research is successful then we apply the task that has been found.

```

Agent-Planning (agent, tree, mode)
  If (mode = STRATEGIC or mode = TACTICAL or mode =
  OPPORTUNISTIC) then
    Resource-Planning (agent)
    Task ← Main-task (tree)
    Goal ← has-goal (Task)
    Subtask ← has-subtask (Task)
    Constructor ← has-constructor (Task)
    Result = Solve-Mandatory-Task-Conditions (Task)
    T-CLU = 2*Number-of-BTCU-Tasks (tree)/3
    If result = FALSE then Return FAILURE
    Else //step 1
      Mandatory-Activated-State = TRUE
      Result = BTCU-Treatment (T-CLU, tree, Task, mode)
      If result = TRUE
        Then BTCU-Activated-State = TRUE
        Else BTCU -Activated-State = FALSE
        Result = Favourable-Task-Conditions (Task, mode)
        If result = TRUE then Favourable-Activated-State = TRUE
        Else Favourable-Activated-State = FALSE
        If (agent-expert? = TRUE and agent-cautious? = FALSE) or
        (mode <> STRATEGIC ) then Regulatory-Activated-State = FALSE
        Else result = Regulatory-Task-Conditions (Task, mode)
        If result = TRUE then Regulatory-Activated-State =
          TRUE Else Regulatory-Activated-State = FALSE
        Result = Solve-Resource-Task-Conditions (agent, Task)
        If result = TRUE then Resource-Activated-State = TRUE
        Else Resource-Activated-State = FALSE
      If Mandatory-Activated-State = TRUE and //step 2
        Resource-Activated-State = TRUE then
          T-CLU = T-CLU - 1
          If Complex-task? = FALSE //it is a simple task, a leaf
            Then Apply-Simple-Task (Task)
            Else If Constructor = SEQ then
              Agent-Planning-SEQ (Subtask)
            Else If Constructor = ALT then
              Agent-Planning-ALT (Subtask)
            Else If Constructor = AND then
              Agent-Planning-AND (Subtask)
            Else If Constructor = OR then
              Agent-Planning-OR (Subtask)
        End.

```

Fig. 6. Agent planning algorithm.

We are doing the same treatment for the favourable

⁵ Open Multi-Agents System: a platform developed in University of Technology of Compiegne in France (UTC).

(Favourable-Task-Conditions) and regulatory conditions (Regulatory-Task-Conditions). We are trying to make them true. However, in the case of regulatory conditions, an expert and careless agent will not take them into account. If we are in an alternative case, we will try to solve the regulatory conditions only if there is no other alternative task possible.

The BTCU conditions are linked to the safety conditions of a task (BTCU-Treatment). We compute a value that conditions the number of BTCU that an agent can execute (T-CLU). If this value is not nil and if the task is a BTCU task, then depending on the control mode we determine if the agent will do the BTCU or not. If the agent is in a tactical mode, if the T-CLU is not nil we choose in a random way if he will do the BTCU or not. If the result is true then we check if the BTCU condition is true then the final decision is that the agent will do the task. In the other modes, the accomplishment of the BTCU task relies on another criteria such as caution, expertise and safety implications of the task.

Finally, to solve the resource conditions (Solve-Resource-Task-Conditions), we determine if the agent has the tool in his toolbox or with him. If not, if he is in a strategic or tactical mode and if he is not in a hurry he can choose whether to look for the tool in the industrial unit or to ask another agent. If he choose the latter solution, if the other agent is not available this will increase his stress and he will look for the tool in the industrial unit. If he is in the opportunistic mode he will reason by analogy. He will look for a tool that can replace the needed one. For example, he will take a pawl-spanner instead of an adjustable-spanner. In the scramble mode he tries to find the tool.

In the step 2, if the step 1 returned true for the mandatory and resources conditions then we check if the task is a complex task or not. If it is not a leaf of the task tree we called other functions depending on the task constructor.

The constructor SEQ obliges each subtask to be realized one after the other. The constructor ALT allows one task to be selected from among several tasks depending on: the environmental conditions, the state and level of the operator, and the available resources. Choosing one of the alternatives tasks depend also on specific BTCU and on the control mode of the operator. If the alternative task does not succeed then the agent can choose another possible one. The constructor OR works like ALT but the task can be chosen in any order. The constructor AND works like SEQ but each task must to be realized.

If the task is a leaf of the task tree, we called *Apply-Simple-Task*. We have different categories of tasks: decisional tasks, perceptive tasks and actions tasks. The leaves are actions tasks. Some actions tasks required another level of planning. Let us consider the task “put a deposit on the gate”. To do this task, the agent will first set up his goal. To solve the mandatory conditions it will look for a task whose effect on the world will be: gate is unlocked. After setting up his goal, the agent uses COLOMBO and the object model to look for a method to lock the gate depending on the kind of gate. He will: (1) ask COLOMBO which kind of gate it is, (2) look for a method to lock the gate in the ontological object model, (3) apply the method/action found. The preconditions of an action are for example to be near the object and to be in the good

position. All these preconditions are verified when we call *Apply-Simple-Task*. If the agent is an expert, the environment conditions will be integrated into the reasoning. At the simple task level we are in a planning mode and no longer in an action selection mode. When the agent selects an action or a decision, for example a task of moving and positioning, a request message is send to COLOMBO that in turn sends a message to the virtual environment. Then a lower level behavioural animation is called.

```

Solve-Mandatory-Task-Conditions (task)
Mandatory ← Evaluate (has-mandatory (task))
If Mandatory = TRUE then return TRUE
Else result = Apply-Solving-Method (task, Mandatory)
If result = TRUE then return TRUE
Else return FALSE
End.

```

Fig. 7. Algorithm to solve the mandatory conditions of a task.

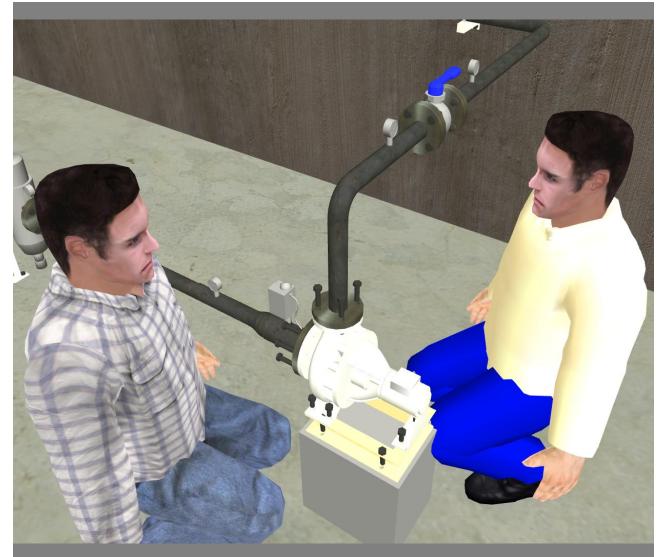


Fig. 8. Collaborative work in the virtual environment.



Fig. 9. Virtual environment for risk prevention. On the picture we can see the agent processing the task: open the gate.



Fig. 10. Virtual environment.

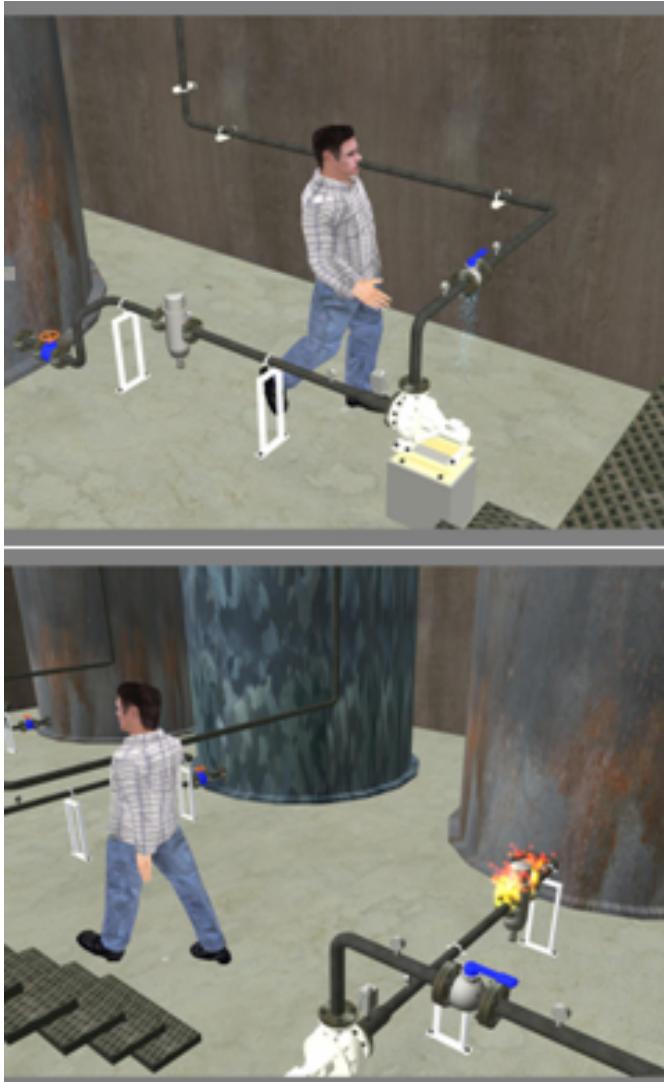


Fig. 11. Risk Visualisation: a leak or a fire.

VI. THE VIRTUAL ENVIRONMENT

6.1 The Virtual Environment

We develop the virtual environment corresponding to our working scenario: a pipe substitution operation in a high-risk industrial plant. The operations comprise three phases: (1) prepare and secure the intervention zone, (2) dismantling of the mono-pomp group, (3) assembly of the new mono-pomp group. The activity model has been developed with VISUAL HAWAI. The character is modelled with 3DSMAX and the virtual environment is designed with VIRTOOLS. In the environment, the learner (avatar) has to execute a collaborative maintenance task with other virtual characters (virtual agents) as shown Fig. 8. The manager in charge of the plant gives his order to the agents. When receiving an order, the agents compute a plan and then execute it. Each action modifies the state of the environment, the state of the object. We can see in the virtual environment in real time the progress of the agents and of the virtual characters (Fig. 9, Fig. 10). We are also able to see the impact of their decisions and the consequences of their errors.

For example, if the agent decides to open the gate without removing the rest of product (action of a novice) present in the pipe, leak of the dangerous product can occur.

VII. CONCLUSION AND OUTLOOK

In this article we describe a system developed to model virtual autonomous characters and their activities in risky situations to support learning, decision-making and risk prevention. Because human-factors are essential in such a training system we based our work on a cognitive model. We proposed a description language to represent human activity taking into account the safety aspects (HAWAI-DL). We also built a multi-agent system MASVERP, modelling the autonomous behaviour of virtual operators. The multi-agents system is connected to the virtual environment through the COLOMBO module in charge of managing every change in the environment or any change in an object state. These changes can occur after an action executed by the trainee (avatar) or by the virtual operators.

Our multi-agent system is based up on action selection and cognitive planning. The agents take their decisions and create a plan depending on their physical and cognitive characteristics. The control mode delimits the choice of an action. Depending on this parameter the agent plans broadly and chooses the actions more adapted to the situation or plans to a more limited degree and compromises on safety aspects to gain productivity.

Simulating virtual human behaviour remains a challenging issue, especially when real-time animation must be supported but we do not and will not focus on path planning, virtual character facial modelling and locomotion. Our future work will focus on removing some limitations of our system by having a better integration of knowledge in the agent environment, by letting agent learn or introducing memory features. The first results are encouraging and our work elaborated a new axis of research on cognitive simulation and modelling. We strived to develop a generic platform and model. A good test will be to use different scenarios seeing how much

work is needed to adapt it to our technology.

ACKNOWLEDGMENT

The partners of the project are UTC/UMR6599, Paris Descartes University, CEA-LIST, SEEMAGE, INERIS, TICN, CICR, SIGROUP, EBTRANS and APTH. We want to thank M. Sbaouni who developed the virtual environment. We thank R. Perney who implemented Visual HAWAI. We thank A. Ben Ayed who built the activity model with Visual HAWAI.

REFERENCES

- [1] L. Edward, D. Lourdeaux, D. Lenne, J. P. Barthes, J. M. Burkhardt and C. Fabrice et al. V3S: A Training and Decision Making Tool to Model Safety Interventions on SEVESO Sites, *Virtual Reality International conference*, VRIC, France, April 2007.
- [2] J. M. Burkhardt, C. Gounelle and P. Cabon. Virtual reality, Human Factors and Safety-related Issues: The VIRTUALIS Project, *Virtual Concept*, France, 2005.
- [3] M. C. Salzman, C. Dede, R. Bowen Loftin and J. A. Chen. Model for Understanding How Virtual Reality Aids Complex Conceptual Learning, *Presence*, vol. 8, no 3, pp. 293-316, June 1999.
- [4] D. Mellet d' Huart. From Reality to the "The Real": Using Augmented Virtual Reality for Training. Virtual Reality: Cognitive Foundations, *Technological Issues & Philosophical Implications*, pp. 129-139, 2001.
- [5] M. Sahin, S. Yaldiz, F. Unsacar B. Yaldiz, N. Bilal, E. Maravelakis and A. Antoniadis. Virtual Training Centre for Computer Numerical Control, *International Journal of Computers, Communications & Control*, ISSN 1841-9836, E-ISSN 1841-9844, vol. 3, no. 2, pp. 196-203, 2008.
- [6] W. Shao and D. Terzopoulos. Autonomous Pedestrians, *Graphical Models*, vol. 69, no. 5-6, pp. 246-74, September-November 2007.
- [7] J. Funge, X. Tu and D. Terzopoulos. Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters, in *SIGGRAPH'99*; Los Angeles; pp. 29-38, August 1999.
- [8] G. Thomas and S. Donikian. Virtual Humans Animation in Informed Urban Environments, in *Computer Animation; IEEE Computer Society Press*: Philadelphia, PA, USA; pp. 129-136, May 2000.
- [9] J. J. Gibson. The Theory of Affordances, in R. Shaw & J. Bransford (eds.), *Perceiving, Acting and Knowing*. Hillsdale, NJ: Erlbaum, 1977.
- [10] N. I. Badler, D. Chia and S. Chopra. Virtual Human Animation Based on Movement Observation and Cognitive Behaviour Model, *Computer Animation Conference*, Switzerland, 1999.
- [11] A. Newell. Unified Theories of Cognition, *Harvard University Press*, 1990.
- [12] J. Rickel and W. L. Johnson. STEVE: A Pedagogical Agent for Virtual Reality, *proceedings of the 2nd International Conference on Autonomous Agents (Agents '98)*, ACM Press, New York, 9-13, pp. 332-333, 1998.
- [13] J. Rickel, L. Johnson and T. Marcus. STEVE: A Virtual Human for Individual and Team Training, 2000.
- [14] J. E. Laird, A. Newell and P. S. Rosenbloom. SOAR: An Architecture for General Intelligence, *Artificial Intelligence*, vol. 33, no. 1, pp. 61-64, 1987.
- [15] J. Rickel and W. L. Johnson. Extending Virtual Humans to Support Team Training in Virtual Reality, in G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, pp. 217-238. Morgan Kaufmann, San Francisco, 2002.
- [16] G. Mendez, J. Rickel and A. de Antonio. STEVE Meets JACK: The Integration of An Intelligent Tutor and A Virtual Environment with Planning Capabilities, in *Intelligent Virtual Agents, 4th International Workshop*, IVA03, no. 2792 in LNAI, pp. 325-332. Springer-Verlag, 2003.
- [17] B. Pazard, J. Dugdale and J. L. Soubie. A Pragmatic Development of A Computer Simulation of An Emergency Call Centre, *DCS, Al. IOS Press*, 2000.
- [18] R. Querrec, P. Chevallier and J. Tisseau. Virtual Reality and Multiagents Systems for Fire-fighting Training in Virtual Environments, *Workshop on Virtual Reality*, Rio de Janeiro, pp. 27-29, November 2002.
- [19] E. Hollnagel. The Phenotype of Erroneous Action, *International Journal of Man-Machine Studies*, vol. 39, pp. 1-32, 1993.
- [20] E. Hollnagel. Cognitive Reliability and Error Analysis Method-CREAM, *Elsevier*, 1998.
- [21] E. Hollnagel. Human Reliability Analysis: Context and Control, *Academic Press*: London, 1993.
- [22] P. Polet, F. Vanderhaegen and R. Amalberti. Modelling Border-line Tolerated Conditions of Use (BTCU) and Associated Risks, in *Safety Science journal*, pp. 111-136, Elsevier, 2003.
- [23] N. El-Kechai and C. Després. A Plan Recognition Process Based on a Task Model, for Detecting Learner's Erroneous Actions, *proceedings of ITS Conference: Intelligent Tutoring Systems*, M. Ikeda et al. (Eds.), Springer-Verlag LNCS n° 4053, Jhongli (Taiwan), pp. 329-338, 2006.
- [24] L. Edward, D. Lourdeaux, D. Lenne, J. P. Barthes, J. M. Burkhardt and C. Fabrice. Multi-agents Approach for Modelling Safety Interventions on a SEVESO Site through Virtual Reality, *Virtual Concept*, Mexico, December 2006.
- [25] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice, *AAII*, Australia, 1995.



Lydie Edward, (05 august 1982) integrated the University of Technology of Compiegne in 2002 where she completed her computer engineer diploma in technology, information and system fields. She joined the Heuristic, diagnosis and complex systems (HEUDIASYC) laboratory in 2006 where she is pursuing a PhD in the field on virtual reality and multi-agent systems for risk prevention. Her current research interests include virtual human behaviours, cognitive planning and action selection.



Domitile Lourdeaux (03 january 1973) is an associate Professor in Computer Science in HEUDIASYC/UMR 6599 laboratory since 2005. She has a PH.D in Robotic and Computer Sciences from "Ecole des Mines de Paris". She is the leader of the V3S project funding by the ANR (French National Agency). The principal interests of her current research are the Virtual Reality and Knowledge and more precisely the modelling of virtual autonomous character and the Intelligent Tutoring Systems for produce a Learner diagnosis and adapted feedback.



www.utc.fr/~barthes

Jean-Paul Barthès (1945) is a Professor in Computer Science, co-founder and member of the HEUDIASYC/UMR 6599 since 1981. He has a degree from "Ecole Centrale de Paris" and a PhD from Stanford University. His research interests are in artificial intelligence, knowledge representation and systems of cognitive artificial agents. He is a member of several editorial boards and chairs the steering committee of several international conferences. For additional information please refer to the personal web page



Dominique Lenne is an associate Professor at the University of Technology of Compiegne. He is a member of the "Information, Knowledge and Interaction" team of the HEUDIASYC UMR 6599 laboratory. He has a PhD in Artificial Intelligence and he has published many papers in journals and conferences in the e-learning domain. His current research interests include interaction modelling, knowledge engineering and virtual reality.



Jean-Marie Burkhardt is associate professor in Cognitive Ergonomics at ECI lab (Ergonomic Behaviour and Interactions) of Paris Descartes University. After a Master's Degree in Ergonomics (1992) and a Master's Degree in Cognitive Psychology (1993), he obtained a Ph.D. in Cognitive Ergonomics from the Paris Descartes University on cognitive processes and representations involved in object-oriented software reuse. His current research interests are ergonomics issues related to the design and evaluation of emerging technologies like Virtual, Mixed and Augmented Reality.