

Easy Creation of Game-Like Virtual Learning Environments

Emmanuel Blanchard

HERON Lab, Computer Sciences Dept,
University of Montréal.
blanchae@iro.umontreal.ca

Claude Frasson

HERON Lab, Computer Sciences Dept,
University of Montréal.
frasson@iro.umontreal.ca

Abstract: Motivating students is a major issue for current Intelligent Tutoring Systems. Videogames are known to have a very addictive effect on their users. We have the intuition that such a phenomenon is partially due to the autonomy support videogames provide to their user. In fact, modern theories of motivation such as the Self Determination Theory (SDT) have shown the positive motivational impact of autonomy-support. In SDT, the need for autonomy refers to the need for someone to see his behaviors as self-endorsed, to be in total control of his activities. Following this, we are currently developing a multi-agents system called MOCAS (MOtivational and Culturally Aware System) in order, among other things, to investigate and clearly determine the impact of autonomy support on learners' motivation in hybrid applications of ITS and videogames. The creation of such a kind of ITS is in general very time consuming and necessitates highly technical knowledge (3D programming, networking...). If the development of virtual learning environments was made easier, a new population of non technical authors would have access to such a tool. For example teachers could create virtual learning environments for supporting their student's home works. This paper introduces MOCAS and presents our ongoing efforts to facilitate the creation of virtual learning environments through it. To this end we are also developing a set of authoring tools integrated in a more complex system called IBIS (Intuitive Builder for Intelligent Systems). IBIS is aimed at providing easy-to-use authoring tools for MOCAS (even for non-computer specialists)

Keywords: Multi-Agents System, Pedagogical Agents, Cultural Adaptation, Motivation.

INTRODUCTION

Motivating students is a major issue for current Intelligent Tutoring Systems [1, 5, 6, 14]. In recent years, many studies in the ITS domain have tried to take advantage of the well-known addictive effect of videogames on their users [10, 22] in order to tailor motivating learning systems. However, the explanation why videogames are so addictive is not very clear and it appeared that all the developed learning games are not so effectively motivating.

Modern theories of motivation such as the Self Determination Theory (SDT) [4, 13, 16, 20] have shown the positive motivational impact of autonomy-support i.e. the need for someone to see his behaviors as self-endorsed, to be in total control of his activities. One possible explanation of the strong motivational effect of videogames could be the autonomy support provided by games.

Among other things, in order to verify such an affirmation, we are currently developing MOCAS (MOtivational and Culturally Aware System), a multi-agents ITS which is aimed at enhancing/maintaining the learners' motivation by providing an autonomy supportive game-like learning environment to learners. The importance to take learner's culture into account soon appeared obvious to us if we wanted to correctly enhance/maintain the learner's motivation [2]. Thus we developed a methodology for cultural adaptation [3] that is also inserted in MOCAS. We also realized that creating such kind of system was very time consuming and technical. Thus, we decided to develop a generic framework in order to be able to create virtual learning environments for different domains of knowledge.

In this paper we discuss our ongoing research on the MOCAS project. In the first part, we present the basic principles behind the development of MOCAS. We present the autonomy-supportive design of MOCAS, our work to make it culturally adaptive and we describe the multi agent architecture of this system. Then in the second part, we present different tools we are building for helping creating virtual learning environments for MOCAS. These tools are grouped in a more general system called IBIS (Intuitive Builder for Intelligent Systems) and are designed to be easy-to-use even for non-computer specialists. Finally we briefly discuss the difference between the approach in our system and other similar studies.

MOCAS: A MOTIVATIONAL AND CULTURALLY AWARE SYSTEM

A Motivation-Oriented System

One of the currently most popular approaches to deal with the motivation of people is to try to enhance the sense of autonomy people feel during an activity. Hence, according to a review of literature made by Reeve and his colleagues [20], many studies have shown that:

“Students with autonomy-supportive teachers compared to student with relatively controlling teachers, show greater mastery motivation, perceived competence and intrinsic motivation, greater conceptual understanding, higher academic performance, and greater persistence in school” [20].

Reeve has also succeeded in showing that increasing teachers’ autonomy support results in an enhancement of students’ engagement in a learning task.

Autonomy-support as a way of enhancing motivation is one of the major points of the Self Determination Theory (SDT, [4, 13, 16, 20]) among others. The autonomy notion must be understood and used cautiously. In SDT, it totally differs from the notion of individualism [4]. In fact, the need for autonomy refers to the need for someone to see its behaviors as self-endorsed [13], meaning that the individual has taken the decision to do such behaviors because he expects that these behaviors will allow him to have positive results while performing an activity.

According to SDT, encouraging people to make choices during an activity is one of the methods used to provide autonomy. On the opposite, any action which controls or restrains someone’s behaviors has a negative impact on someone’s sense of autonomy, which results in lowering its motivation to persist in this activity. The design of MOCAS is aimed at providing autonomy to learners, which should lead to increased learners’ motivation.

In previous work [1], we have argued that using an Open Virtual World is autonomy-supportive and, therefore, a well indicated design for supporting motivation. OVW is a frequent concept in the Role Playing Game field. MOCAS uses a 3D OVW but OVWs are not strictly related to 3D environments. The essence of OVW concerns the ability for the user to make choices while he is interacting with the environment, to be the initiator of his behaviors and not to have to follow a sequence of predefined interactions. To summarize, OVWs can be seen as discovery learning environments (environments where learners will try to “construct” the knowledge of the domain [21]) that are focused on ways of enhancing proactive behaviors of users.

Despite all positive motivational aspects of OVWs, a learner facing an environment where he is free to do many actions may sometimes forget his learning objectives. He may also decide not to do anything. Thus, even in an OVW, coaching the learner remains an important need in learning tasks. The objective for the system is to interact proactively towards the learner “only” when it is necessary.

Nick Jennings says that multi-agents systems can be used to present multiple perspectives of a problem [9] and, as we have seen before, autonomy-support can be done by offering multiple choices to a learner. Following this, we have inserted a set of pedagogical agents in our OVW. The aim of these agents is to provide to the learner, with different perspectives, the guidance/coaching necessary to keep him focused on the learning task and also on the learning contents. To this extent, our agents can have different roles (for example, in a medical simulation, roles could be doctor, nurse, patient...) and different attitudes. They are able to teach only concepts that fit with their roles (it means that they have access to all the resources related to theses concepts) but they can also refer the learner to another agent, the knowledge of which is better related to the needs of this learner. Furthermore, as noticed by Lester and his colleagues, such agents can “*play a critical motivational role as they interact with students*” and have “*an exceptionally positive impact on students*” [12]. Figure 1 presents the interface of a learner in MOCAS (the course here is concerning Greek mythology).

There are three different parts in this interface: *the interface for the game-like 3D environment* (where the avatar of the learner can navigate and interact with avatars of pedagogical agents), *the communication interface* (to allow the learner to communicate with other online learners) and *the learning content interface* (where information in relation with the domain to be learned will be culturally adapted. Then it can be displayed following different layouts given the recommendations of pedagogical agents). The learning content interface is a modular interface. For example, in figure 1, the loaded module displays an HTML file. But other modules exist that allow displaying video files, questions and answers interfaces... It is also possible to develop modules specifically related to the domain to be learned. For example, we could imagine developing a command board that allows manipulating a 3D model of a medical device in a medical course.

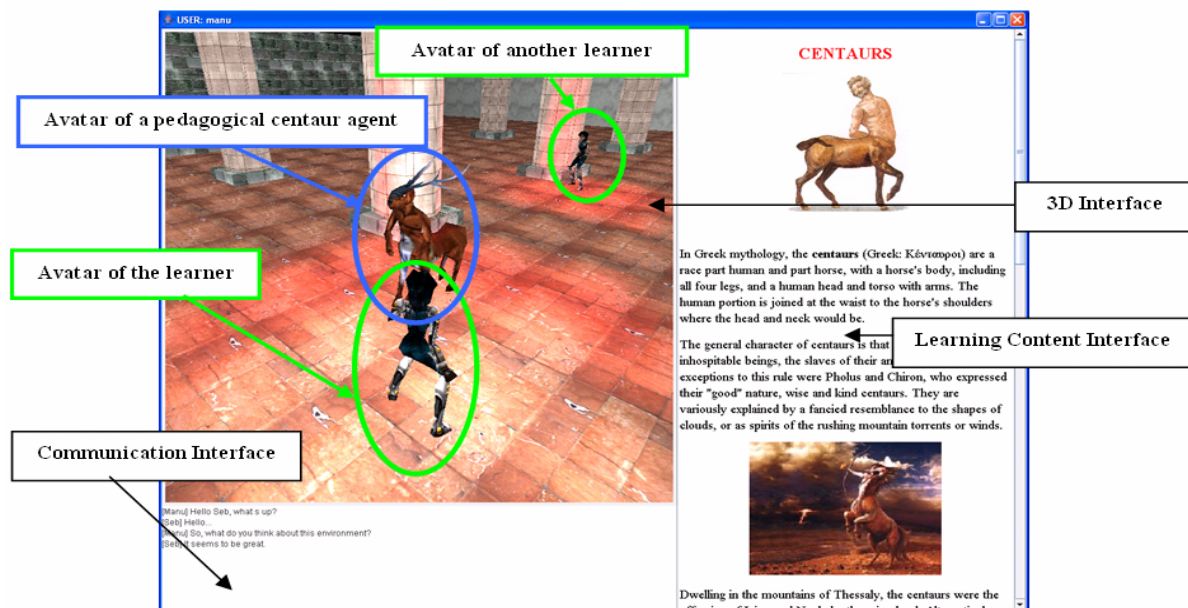


Figure 1: interface of the learner's application in MOCAS.

A Culturally Adaptive System

When reading cross-cultural literature, it appears clear that ITS and eLearning in general can highly benefit from dealing with cultural differences. In fact, the cultural background affects many elements that are important in the ITS research field such as *emotions* [11, 18], *preferences for a pedagogical strategy* [2], *meanings given to concepts and symbols* [8], *motivation and autonomy support* [4, 13] (which is our main interest in the development of MOCAS)... These examples stress the importance for ITS and eLearning systems in general to have some kind of *Cultural Intelligence* i.e. a “*seemingly natural ability to interpret someone's unfamiliar and ambiguous gestures the way that person's compatriots would*” [7].

In fact we believe that a culturally aware system such as MOCAS must have the ability for *cultural understanding* (i.e. culturally interpreting a learner's behavior/feeling/result) and *adaptation* (i.e. displaying different interfaces and/or starting different learning strategies depending on learners' culture). These suggestions resulted in a methodology for cultural adaptation [3].

In our methodology, *Cultural Facts* are deduced from the cross-cultural literature. In fact, we use the *Hofstede's system of values* [8]. It represents each national culture with a set of dimensions and associated scores. Hofstede's five dimensions are *Power Distance*, *Individualism/Collectivism*, *Masculinity/Femininity*, *Uncertainty Avoidance* and *Long Term Orientation* (see <http://www.geert-hofstede.com/> or [8] for further information). In a practical way, when a student registers to MOCAS, he indicates the country he is currently living in and also the other countries where he could have spent his childhood (we are currently thinking about adding complementary information in our process such as the religion of the learner or Schwartz' cultural values [17]). Given the answer of the learner, MOCAS can affect a score to each of Hofstede's dimensions for this learner. These scores are then used in order to initialize a rule-based engine. This engine is aimed at determining certainty weights of pedagogically-related attributes (for example the interest for collaboration).

To summarize, a learner is described by a vector of weighted pedagogically-related attributes that we call Rules Weights Vector (RWV) and that is initialized depending on the learner's cultural profile. Each cultural group has a similar RWV (which is the mean RWV for all its learners). The membership of a learner to each cultural group called Membership Score is also determined using the normalized distance between the learner's RWV and the RWV of a given cultural group. During the learning process and depending on learners' successes/failures, the weights of learners' RWV evolve and they in turn affect groups' RWV and also all the membership scores.

Finally, all pedagogical resources and strategies are dynamically rated in order to represent the interest to use them with learners of a given cultural group: this is their Cultural Interest Score. When a learner needs to learn some concept, a pedagogical resource (among all the pedagogical resources that are available to represent the concept) will be selected depending on:

- The Membership Scores of this learner to different cultural groups and,

- The Cultural Interest Scores of all the available resources that could represent the concept.

A multi-agents architecture

Figure 2 presents the global multi-agents architecture of MOCAS.

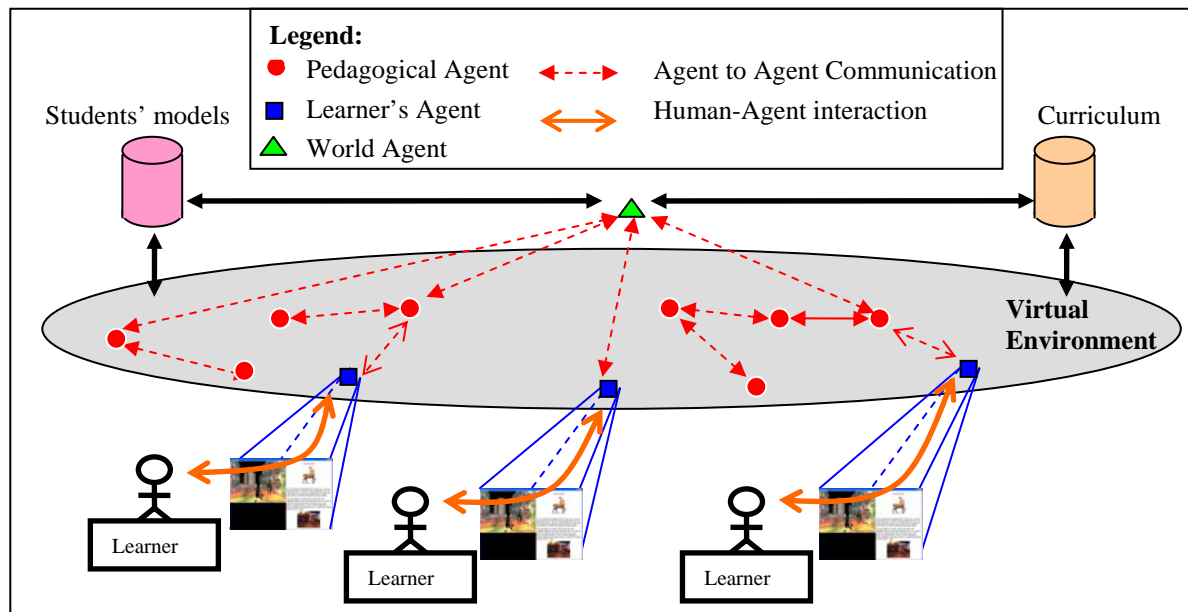


Figure 2: the multi-agents architecture of MOCAS

As we said before, there are 3 kinds of agents in MOCAS:

- Multiple *pedagogical agents*: they are in charge of processing pedagogical strategies and, if requested, they will cooperate with the learner's agent in order to provide the teaching lesson the way it has been determined.
- Multiple *learner's agents*: those are interface agents and each of them is in charge of scrutinizing one learner's interaction with the client interface. They also have the ability to modify the client interface to provide culturally adapted pedagogical support.
- One *world agent*. It is in charge of monitoring the whole activity in a MOCAS world and determining global strategies of action for pedagogical agents. It is also in charge of starting and managing the genetic evolution process of our population of pedagogical agents after a determined period of time (or once a given number of human-agents interactions have occurred).

Technical Implementation

MOCAS is fully developed using JAVA technologies. The choice of JAVA technology is not innocent. When reading ITS literature, we remark that this language is the principal one used in our community and we have the idea of releasing the Application Programming Interface (API) of MOCAS once we will have a more stable system. This is a frequent attitude in the game industry to allow teams of independent developers to create original applications called MODS (for Modifications) typically developed to personalize or enhance some aspects of the game.

JAVA offers many advantages such as portability, easy and fast prototyping and a set of useful APIs for many different purposes. However, until recently it was not commonly suitable for 3D applications. Things have changed with the apparition of OpenGL bindings such as LWJGL (<http://www.lwjgl.org>) or JOGL (<https://jogl.dev.java.net/>) that allow OpenGL hardware rendering.

In terms of API, we use the JADE multi-agents platform (<http://jade.tilab.com/>) for the development of agents, their behaviors and the communication between them (using the Agent Communication Language). The JESS rules engine (<http://herzberg.ca.sandia.gov/>) is used for the management of cultural rules and we develop and manage the 3D part of the system with the JPCT engine (<http://www.jpct.net>, based on LWJGL). JPCT allows us to use multiple types of popular 3D model formats (such as 3DS or MD2)

AUTHORING FOR MOCAS

As pointed in our introduction, the development of Virtual Learning Environments inspired by 3D games is a very complicated process. In order to reduce the amount of work for the development of original courses in MOCAS, we developed a set of authoring tools that we regroup in IBIS i.e. Intuitive Builder for Intelligent System. The Intuitive word is important in IBIS. We want to obtain an easy to use authoring tool even if authors have no big knowledge in computer sciences (it is frequently the case of teachers). That's why we chose a WYSIWYG approach and that we preferred having a simple system with less functionality than a complex but complete system.

Authoring virtual 3D environments

In the game development process, one of the first elements to be created is a map editor. Maps are crucial in virtual environments because it is the place where the action will occur and many process of the application logic depend of the chosen representation. When building 3D environments, the authoring tool is frequently based on a 2D representation. Following this, we currently use bitmap images to represent our world (see figure 3). Each color is a code that contains the following information: the ability or not to walk across the square (it will be used by the path finding algorithm) and the corresponding representation of this tile in the 3D environment. Currently we can use any kind of paint program to produce the bitmap. It will then be passed as an argument to the world agent that will generate the corresponding 3D environment. In the near future, an under development WYSIWYG module for such a creation will be added to IBIS.

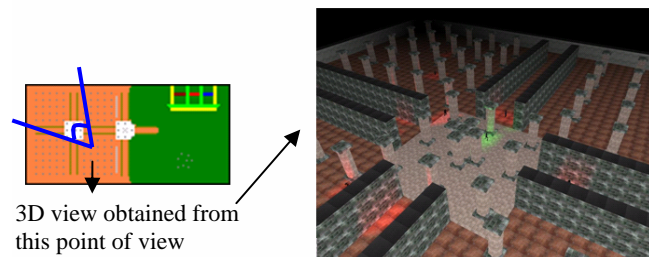


Figure 3: authoring 3D virtual environments

Authoring the curriculum

Authoring virtual environments for a learning purpose has specific issues. The main one concerns the question of knowledge representation. How to represent the curriculum? As we pointed before, the targeted audience of IBIS may comport non computer specialists. Following this point, we decided that conceptual graph was a decent technique for authoring the curriculum. It has the advantages to be a very graphical and rather intuitive technique. In fact our knowledge representation is not as efficient as some methodology currently developed but once again, our main objective was to make it simple. The resulting tool is presented in figure 4.

This tool contains two views: -1- a *graph view* on the left that is the representation of the sub graph of the currently selected concept (it means that it contains all the nodes representing the sub concepts that could be discussed when trying to study the currently selected concept) and -2- a *tree view* (on the bottom right) which shows all the concept defined in this curriculum. Concepts that are presents in the sub graph of another concept are seen as children of the concept in the tree view. Only the leaf nodes of the tree can contain pedagogical resources that could be used to illustrate a concept (i.e. the parent concept) during a course.

When creating a classical concept (represented as a blue rectangle in figure 4), two children are automatically created, which means that classical concepts are never leafs (thus a classical concept never contains resources): the *entry point* will be the starting node to the study of the classical concept and the *ending point* will be the node to reach in order to have the study of the concept completed. At this moment the concept will be stated as "studied". Both these nodes can contains pedagogical resources (following the cultural adaptation methodology, one of the stored resource will be presented when the corresponding node will be reached). If the author wants to have intermediate nodes that contain resources, it can create a *validation point* (green hexagon). Furthermore and as we said before, nodes for sub-concepts can be added to complete the graph.

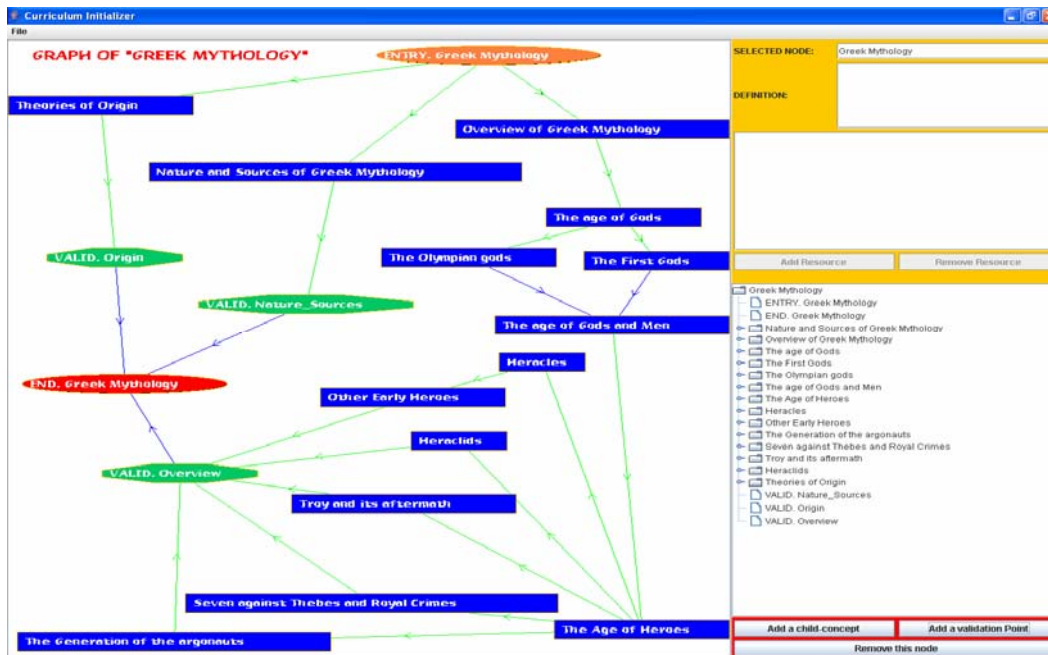


Figure 4: the curriculum authoring tool

In order to navigate through the sub graph of a concept, nodes can be linked either with *direct links* (green arrow) or *secondary links* (blue arrow). Figure 5 describes this aspect of our graph.

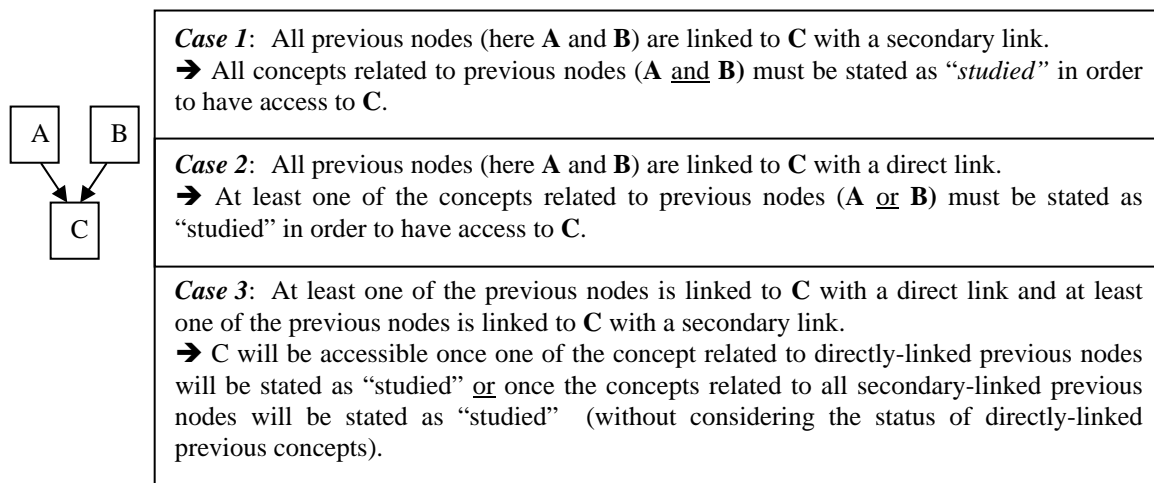


Figure 5: possible use cases of direct and secondary links in our authoring tool

An XML file is generated by the use of this tool. It describes the structure of the curriculum and that may be used by pedagogical and world agents to determine and organize the next learning session according to the learner’s profile. It is also used to represent the knowledge part of the student model (using the overlay methodology). A file system is also obtained. It contains folders representing the nodes that contain pedagogical resources. The client side of MOCAS contains this file system, which permits an easy access to pedagogical resources related to the domain to be learned.

Defining pedagogical agents and associating knowledge to them

Finally, the last important part of the creation of a learning environment for MOCAS concerns the definition of the pedagogical agents that will populate the virtual environment. Figure 6 presents the tool we developed for such purpose.

On the top of the interface, the list of already defined agents is displayed. Each definition contains the name of the agent and also the name of the 3D model file and the associated texture that will be used to represent its avatar in the 3D environment. Currently, we use the MD2 file format, a format for 3D models that allows the definition multiple key frame animations for a model. It was developed for the popular Quake2 game.

On the bottom left of the interface, the tree view of a curriculum can be loaded. The author will then have the capacity to add/remove knowledge (concept) to each agent definitions. Knowledge that is accessible to an agent is listed on the bottom right part of the interface. As we said before, agents can have different roles in MOCAS, which limits their area of expertise.

With this tool, authors can directly determine the expertise of each agent they create. The resulting XML file is passed as argument of the World agent that will have to initialize all the pedagogical agents once for a virtual environment.

COMPARISONS WITH RELATED WORKS

Many others research teams are currently working on game-like environments for learning. For instance, Tan and his colleagues have designed a game-like learning environment to teach an ecological course [19]. The CARTE group has also designed many game-like environments, mostly for the US Army [10, 22]. All these systems try to take profit from the motivational aptitude of videogames. Hence they all share a certain number of elements. However MOCAS presents significant differences with these systems:

- MOCAS tries to culturally adapt its interactions and the learning content depending on learners' cultural profile. Currently, we haven't found any ITS with a similar objective or ability.
- MOCAS design and pedagogical agents are all developed in order to follow recommendation of the Self Determination Theory, focused on autonomy-support as a way of enhancing motivation. Motivational aspects of other systems are not directly relying to any specific motivational theory, which make them hard to predict.
- MOCAS is fully developed using JAVA which is very popular among our community. As we said before, JAVA offers many advantages such as portability, easy and fast prototyping and a set of useful APIs for many different purposes. One of our major concerns during the development of MOCAS is to try to develop it in a modular way in order to facilitate a potential reuse in other 3D learning systems.
- MOCAS uses JADE, a FIPA compliant multi-agent platform. JADE is a well documented. The use of this normalized platform should facilitate developments, tests and comparisons:
 - of different types of pedagogical agents,
 - of multi agent teamwork strategies for pedagogical purpose,
 - of communication protocols for different tasks related to pedagogical strategies...
- MOCAS is developed as a generic ITS platform. For what we have seen, other systems seem to be quite domain dependants. Instead of that, the authoring tools we are conjointly developing with MOCAS give us the ability to easily create courses in many different domains.

CONCLUSION AND FUTURE WORKS

In this paper, we have presented the underlying principles of MOCAS, a MOTivational and Culturally Aware System. We explained that the main objective of MOCAS is to maintain or enhance eLearners motivation by fulfilling their need for autonomy as depicted by the Self Determination Theory. A resulting goal is to be able to deal with culture of learners in order to correctly adapt the teaching to the needs of eLearners.

MOCAS follows many principles used in the game domain and particularly in Role Playing Games. Learners proactively interact with pedagogical agents, all being represented by avatars in a 3D environment. Discovery learning experience results from these interactions.

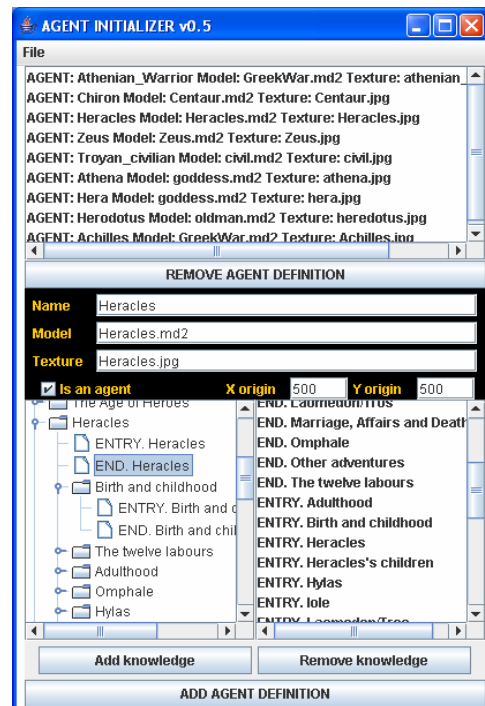


Figure 6: Authoring tool for pedagogical agents

The prototype is in an advanced phase thanks to authoring tools that have been developed conjointly with the main system and first evaluations in real context are currently performed. We will determine the most important thing to improve after having analyzed the results.

Among other things, the cultural student model which is currently very basic has to be improved. New pedagogical behaviors have to be developed. Once again, we may inspire from the game industry and use a scripting method to facilitate the creation of new behaviors. The development and integration of pedagogical agents that could physically exhibit emotions is also under discussion. We are also working on using genetic programming to represent the various attitudes that an agent can have. Finally, a way of publishing the produced virtual environments through the internet following the recommendations related to web services could be very promising.

REFERENCES

1. Blanchard, E., Frasson, C. (2004). An Autonomy-oriented System Design for Enhancement of Learner's Motivation in eLearning. International Conference on Intelligent Tutoring Systems, Maceio, Brazil, pp. 45-54.
2. Blanchard, E., Frasson, C. (2005). Making Intelligent Tutoring Systems culturally aware: The use of Hofstede's cultural dimensions. International Conference. on Artificial Intelligence, Las Vegas, pp. 644-649.
3. Blanchard, E. Razaki, R., Frasson, C. (2005). Cross-Cultural Adaptation of eLearning Contents: a Methodology. International Conference on E-Learning, Vancouver, Canada.
4. Chirkov, V. I., Ryan, R. M., Kim, Y., & Kaplan, U. (2003). Differentiating Autonomy from Individualism and Independence: A Self-determination Theory Perspective on Internalization of Cultural Orientations and Well-Being. *Journal of Personality and Social Psychology*, 8 (1), pp 97-110.
5. Del Soldato, T. (1994). Motivation in tutoring systems. Doctoral Thesis, Technical Report CSRP 303, School of Cognitive and Computing Sciences, University of Sussex.
6. De Vicente, A. (2003). Towards Tutoring Systems that Detect Students' Motivation: an Investigation. Doctoral Thesis. University of Edimburgh.
7. Earley, C., Mosakowski, E. (2004). Cultural Intelligence. *Harvard Business Review*, October 2004, pp. 139-146.
8. Hofstede, G. (2001). *Culture's Consequences: Comparing Values, Behaviors, Institutions, and Organizations across Nations*, 2nd edition, London: Sage.
9. Jennings N. R. (2001). An Introduction to Multi Agent Systems. Tutorial of International Conference on Autonomous Agents, Montréal, Canada.
10. Johnson, W. L., Vilhjálmsón, H., Marsella, S. (2004). Serious Games for Language Learning: How Much Game, How Much A.I.? International Conference on Artificial Intelligence in Education. Amsterdam, the Netherlands. pp 306-313.
11. Kim-Prieto, C., Fujita F., Diener E. (2004). Culture and structure of emotional experience. Unpublished Manuscript, University of Illinois, Urbana-Champaign.
12. Lester, J. C., Converse, S.A., Kahler, S. E., Todd Barlow, S., Stone, B. A., Boghal, R, S. (1997). The Persona Effect: Affective Impact of Animated Pedagogical Agents. Conference of Human Factors in Computer Systems (CHI'97). Atlanta, GA. pp 359-366
13. Levesque, C., Zuehlke, A. N., Stanek, L. R., Ryan, R. M. (2004). Autonomy and Competence in German and American University Students: A Comparative Study Based on Self-Determination Theory. *Journal of Educational Psychology*, vol 96 (1). pp 68-84.
14. Qu, L., Johnson W.L. (2005). Detecting the Learner's Motivational States in an Interactive Learning Environment. International Conference on Artificial Intelligence in Education. Amsterdam, the Netherlands, pp 547-554.
15. Razaki, R., Blanchard, E., Frasson, C. (2006). On the Definition and Management of Cultural Groups of e-Learners. To appear in the 8th International Conference on Intelligent Tutoring Systems, Jhongli, Taiwan.
16. Ryan, R. M. & Deci, E. L (2000). Self Determination Theory and the facilitation of intrinsic motivation, social development and well-being, *American Psychologist*, 55. pp 68-78.
17. Schwartz, S. H. (1992). Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In M.P. Zanna (Ed), *Advances in experimental psychology*, vol 25. New York: Academic Press. pp. 1-65.
18. Scollon, C. N., Diener, E., Oishi, S., Biswas-Diener, R. (2004). Emotions across Cultures and Methods. *Journal of Cross-cultural Psychology*, 35(3). pp. 304-326.
19. Tan, J., Beers, C., Gupta, R., Biswas, G. (2005). Computer Games as Intelligent Learning Environments: A river Ecosystem Adventure. International Conference on Artificial Intelligence in Education. Amsterdam, the Netherlands, pp 646-653.
20. Reeve, J., Jang, H., Carrell, D., Jeon. S., Barch, J. (2004). Enhancing Students' Engagement by Increasing Teachers' Autonomy Support. *Journal of Motivation and Emotion*, 28(2). pp 147-169.
21. Van Joolingen, W. (1999). Cognitive tools for discovery learning. *International Journal of Artificial Intelligence in Education*, 10. pp. 385-397.
22. Rickel, S., Gratch, J., Hill, R., Marsella, S., Swartout, W. (2001). Steve Goes to Bosnia: Towards a New Generation of Virtual Humans for Interactive Experiences. AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment in at Stanford University.