

# Faciliter la création d'environnements virtuels d'apprentissage s'inspirant des jeux vidéo

Emmanuel Blanchard, Claude Frasson

Laboratoire HERON, Département d'Informatique et Recherche Opérationnelle, Université de Montréal.  
{blanchae, frasson}@iro.umontreal.ca

## Résumé

Motiver les apprenants à utiliser des Systèmes Tutoriels Intelligents (STI) est un problème de plus en plus important pour la communauté STI. De nombreuses théories modernes de motivation parmi lesquelles la Théorie de l'Autodétermination (TA) prônent l'encouragement de l'autonomie des personnes dans une tâche comme méthode d'amélioration de la motivation. Par ailleurs, le fort pouvoir d'addiction des jeux vidéo semble notamment provenir de leur capacité à encourager le sentiment d'autonomie de leurs utilisateurs. S'inspirant de tout ceci, nous développons actuellement MOCAS (MOTivational and Culturally Aware System). Ce système a pour but d'étudier et clairement déterminer l'impact du soutien à l'autonomie sur la motivation des apprenants au cours de tâches d'apprentissage, dans des applications hybrides mêlant jeux vidéos et STI. Dans cet article, nous présentons les éléments fondamentaux derrière le projet MOCAS.

**Mots clés :** motivation, soutien à l'autonomie, culture, environnement 3D, système multi agents.

## Abstract

Motivating learners to use Intelligent Tutoring Systems (ITS) is becoming a key issue for the ITS community. Self Determination Theory (SDT) among other modern theories of motivation argues that fulfilling the need for autonomy of people that are performing a task will enhance their motivation. Furthermore, the strong addictive effect of video games seems to be partly related to the autonomy support they provide to their users. Following these points, we are currently developing MOCAS (MOTivationally and Culturally Aware System). The aim of our research is to study and clearly determine the motivational effect of autonomy support during learning tasks performed through a hybrid ITS/video game application. In this paper, we present the basics aspects of the MOCAS project.

**Keywords:** motivation, autonomy support, culture, 3D environment, multi-agents system.

## Introduction

Motiver les apprenants à utiliser des *Systèmes Tutoriels Intelligents* (STI) est un problème qui prend de plus en plus d'importance au sein de la communauté STI

[1,5,6,15]. Dans les dernières années, un certain nombre de recherches concernant les STI ont cherché à tirer profit de l'effet d'addiction bien connu des jeux vidéo sur leurs utilisateurs [10, 18] afin de concevoir des systèmes d'apprentissage motivants. Cependant, la raison pour laquelle les jeux vidéo ont un tel pouvoir de motivation n'est pas aussi claire que cela et on ne rapporte pas systématiquement d'effet motivateur suite à l'utilisation d'un jeu vidéo pour l'apprentissage.

Les théories modernes de motivation telles que la *Théorie de l'Autodétermination* (TA, [13]) ont prouvé l'impact motivationnel positif du soutien à l'autonomie, c.-à-d. le fait d'encourager la personne à décider volontairement de son action, à être elle-même l'agent qui réalise cette action et au final, à assumer entièrement cette action [13]. Il est alors facile de penser que le soutien à l'autonomie que fournissent les jeux vidéo pourrait expliquer en partie leur fort effet motivateur.

De manière à vérifier une telle affirmation dans un contexte d'apprentissage, nous développons actuellement MOCAS (MOTivational and Culturally Aware System), un STI multi agents. Son but est d'améliorer/maintenir la motivation des apprenants en leur permettant d'interagir dans un environnement d'apprentissage virtuel, l'« orientation jeux vidéos » de MOCAS visant à encourager l'autonomie des apprenants [1].

Par ailleurs, un but secondaire pour notre système est rapidement apparu : si nous voulions correctement améliorer/maintenir la motivation de nos apprenants, il était nécessaire de considérer les spécificités culturelles de ces apprenants [2]. Nous avons donc proposé une méthodologie d'adaptation culturelle [3] qui se trouve insérée dans MOCAS.

Finalement, nous sommes conscient que développer un tel système est très demandant, aussi bien en terme de temps de développement que de connaissances techniques. Nous cherchons donc à concevoir, en parallèle à MOCAS, un certain nombre d'outils pour faciliter la création d'environnements virtuels d'apprentissage, pour une variété de domaines d'apprentissage aussi grande que possible.

Dans cet article, nous présentons les travaux actuels que nous effectuons au sein du projet MOCAS. Nous présentons les principes de base selon lesquels est développé MOCAS et nous décrivons les différents outils que nous développons pour faciliter la création d'environnements virtuels d'apprentissage au sein de MOCAS.

## MOCAS : Un système motivationnel et culturellement conscient

### Un système motivationnel

L'une des approches actuelles les plus populaires pour augmenter la motivation de personnes pour une tâche est de chercher à améliorer le soutien à l'autonomie que vont ressentir ces personnes durant l'exécution de cette tâche [13]. Comme on l'a dit précédemment, il s'agit de l'un des points principaux de la T.A. Également selon la T.A., faciliter la prise de choix par les personnes durant une tâche est une des méthodes les plus courantes de soutien à l'autonomie. A contrario, toute action qui vise à contrôler ou à restreindre les comportements d'une personne nuira au sentiment d'autonomie de cette même personne, ce qui aura un effet négatif sur sa motivation à effectuer la tâche. MOCAS a été développé de manière à encourager l'autonomie des apprenants, ce qui devrait résulter en une augmentation de leur motivation.

Dans un article précédent, nous avons mis en lumière que, dans le cadre d'apprentissage en ligne, utiliser des *Environnements Virtuels Ouverts* (EVO, c.-à-d. des environnements où les apprenants peuvent naviguer et interagir librement, un concept fréquemment utilisé dans les jeux de rôle par ordinateur) facilitait l'autonomie et était donc une pratique recommandée si on voulait encourager la motivation des apprenants. Cependant, malgré tous les aspects motivationnels positifs des EVO, un apprenant se retrouvant dans un environnement où il est libre d'entreprendre de multiples actions peut parfois en oublier ses objectifs d'apprentissage (ce qui est appelé l'effet « free rider »). De ce fait, encadrer l'apprenant demeure un besoin important de toute tâche d'apprentissage, y compris dans le cadre d'un EVO.

Nick Jennings énonce que les systèmes multi agents peuvent être utilisés pour présenter un même problème sous de multiples perspectives [9]. En suivant cette affirmation, nous avons donc décidé d'intégrer un ensemble d'agents pédagogiques dans notre EVO. Le but de ces agents est de fournir aux apprenants de l'encadrement et des conseils de manière à les tenir

concentrés sur la tâche et le contenu d'apprentissage. Dans cet objectif, différents rôles peuvent être attribués à nos agents (par exemple, dans une simulation médicale, les agents pourraient avoir les rôles de médecin, d'infirmière, de patient...) et différentes attitudes. Par ailleurs, nos agents ne peuvent enseigner que les concepts qui sont rattachés à leur rôle (une infirmière n'ayant pas les mêmes connaissances qu'un médecin), ce qui signifie qu'ils auront accès à toutes les ressources didactiques reliées à ces concepts. Ils pourront également référer l'apprenant à un autre agent dont les connaissances (les concepts auxquels il a accès) correspondront mieux au besoin de l'apprenant. Ces agents, tout comme les apprenants seront représentés par un avatar au sein de l'environnement d'apprentissage. On peut noter que ce type d'agent est fréquemment employé dans les STI. Lester et ses collègues ont d'ailleurs mis en évidence l'impact motivationnel qu'ils peuvent avoir en interagissant avec l'apprenant [12].

La figure 1 illustre l'interface client dont dispose un apprenant dans MOCAS. Cette interface comporte trois parties : l'interface pour l'environnement 3D (où les avatars des apprenants et des agents pédagogiques coexistent et interagissent), l'interface de communication (pour permettre à plusieurs apprenants humains connectés au même moment de communiquer) et l'interface pour le contenu d'apprentissage (où les informations reliées au cours sont affichées d'une manière ou d'une autre dépendamment du profil culturel de l'apprenant). On pourra par ailleurs avoir à disposition plusieurs modules différents pour permettre l'affichage de contenus divers et variés, sous différentes formes (page web, interface d'évaluation, interface pour contrôler un objet...) de manière à pouvoir traiter de problèmes didactiques propres au domaine d'enseignement.

### Un système culturellement conscient

Lorsque l'on s'intéresse aux recherches interculturelles, il devient rapidement clair que les STI et l'enseignement en ligne en général peuvent tirer un grand profit de

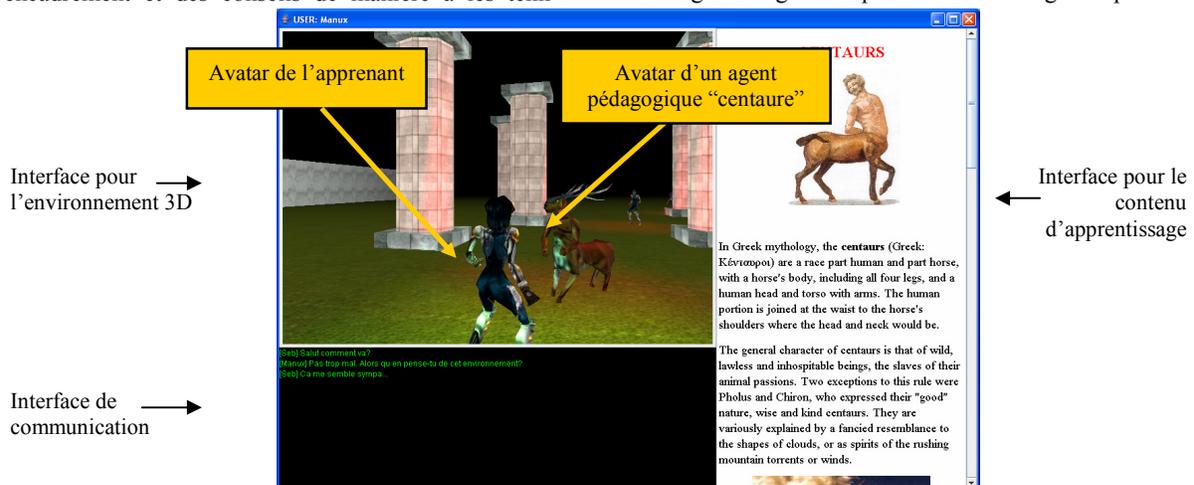


Figure 1 : Interface de l'application client dans MOCAS

traiter les différences culturelles des apprenants. De fait, le profil culturel affecte plusieurs éléments qui sont fortement étudiés et pris en compte dans les recherches sur les STI comme par exemple, les émotions [11, 17], les stratégies pédagogiques à choisir préférentiellement à d'autres [2], les significations données à des concepts ou symboles [8], la motivation et le soutien à l'autonomie ([4, 14] qui est, rappelons-le, la raison principale du développement de MOCAS). Ces exemples mettent en évidence la nécessité de doter les STI et les systèmes de formation en ligne d'une forme d'Intelligence Culturelle décrite par Earley et Mosakowski comme *la capacité apparemment naturelle d'interpréter les comportements non familiers et ambigus d'une personne de la manière dont les compatriotes de cette personne le feraient* [7].

Ainsi nous croyons qu'un système culturellement conscient tel que MOCAS doit avoir les capacités de *comprendre culturellement* (c.-à-d. interpréter culturellement les comportements/sentiments/résultats d'un apprenant) et de *s'adapter culturellement* (c.-à-d. afficher différentes interfaces et/ou mettre en action différentes stratégies dépendamment de la culture de l'apprenant). Tout ceci a mené à la mise au point d'une méthodologie d'adaptation culturelle [3].

Dans cette méthodologie, des *Règles Culturelles* sont déduites de la littérature interculturelle. Par exemple, nous pouvons utiliser le *système de valeurs de Hofstede* [8] ou *l'inventaire de valeurs de Schwartz* [16]. Ces deux études ont abouti à une représentation des cultures nationales sous forme d'un certain nombre de dimensions (ou *valeurs*) auxquelles sont associés des scores. Notre méthodologie utilise ces données comme des *Faits Culturels* de manière à initialiser un système à base de règles. Ce système a pour but de déterminer des poids de certitudes pour des *Attributs Pédagogiques* (par exemple l'intérêt des apprenants pour les tâches collaboratives). Pour résumer, un groupe culturel est décrit par un vecteur d'attributs auxquels sont associés des poids de certitude. Un tel vecteur est appelé *Vecteur de Poids de Règles* (VPR). Chaque apprenant dispose également d'un VPR qui lui est propre et qui est initialisé en fonction de son profil culturel (obtenu suite à un questionnaire succinct). L'appartenance d'un apprenant à chaque groupe culturel appelé *Score d'Appartenance* est également obtenu en calculant une distance normalisée entre le VPR de l'apprenant et le VPR de chacun des groupes culturels. Durant le processus d'enseignement et dépendamment des succès et échecs de l'apprenant, les poids contenus dans les VPR des apprenants évoluent et affectent à leur tour les VPR des groupes culturels ainsi que les Scores d'Appartenance. Finalement, toutes les ressources et stratégies pédagogiques sont elles aussi évaluées dynamiquement de manière à représenter l'intérêt de les utiliser avec des apprenants d'un groupe culturel donné. On obtient donc un *Score d'Intérêt Culturel*. Quand un apprenant a besoin de se former à un concept, on sélectionnera la ressource et la stratégie pédagogique à utiliser (parmi celles disponibles pour ce concept) en fonction des Scores d'Appartenances de cet apprenant vis-à-vis des différents groupes culturels et également

des Scores d'Intérêt Culturel des ressources et stratégies disponibles pour l'étude de ce concept.

## Une architecture multi agents

On trouve trois types d'agents dans MOCAS :

- Une multitude d'*agents pédagogiques* : ils sont en charge de mettre en application les stratégies pédagogiques et, si nécessaire, ils collaborent avec l'agent des apprenant de manière à fournir les leçons du cours de la manière qui a été préalablement déterminée.
- Un *agent personnel pour chaque apprenant* : ce sont des agents d'interface. Chacun d'entre eux scrute les interactions de l'apprenant auquel il est attaché avec l'interface client. Ils ont également la capacité de modifier cette interface de manière à fournir un soutien d'apprentissage culturellement adapté.
- Un *agent de monde* : il a pour charge de gérer l'ensemble des activités au sein d'un monde virtuel mis en place dans le cadre de MOCAS. Il peut ainsi mettre en action des stratégies qui impliqueraient plus d'un apprenant et/ou plus d'un agent.

Chacun de ces agents peut accéder directement au curriculum de la matière enseignée et à l'ensemble des modèles d'apprenants. La figure 2 résume cette architecture multi agents de MOCAS.

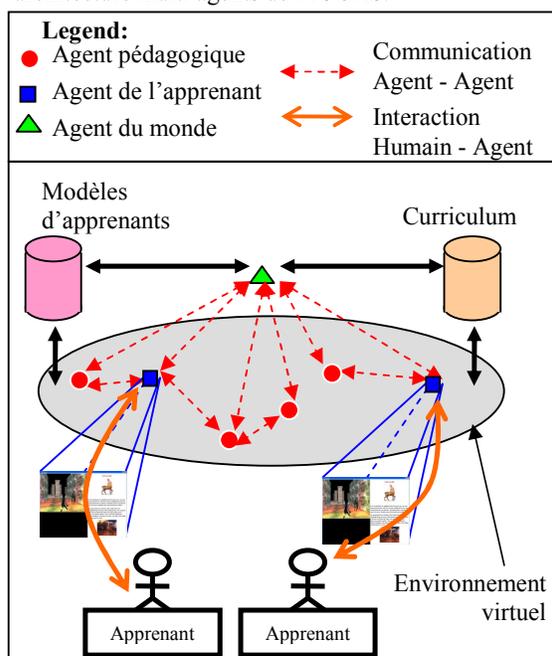


Figure 2 : Architecture multi agents de MOCAS

## Considérations techniques

MOCAS est entièrement développé avec de la technologie JAVA, ce choix n'étant pas innocent. Lorsque l'on regarde la littérature sur les STI, on remarque que le langage JAVA est prédominant pour les développements dans notre communauté. JAVA offre de nombreux avantages qui expliquent cet état de

fait tels que sa portabilité, la facilité et la rapidité d'obtention de prototypes ainsi qu'un ensemble très complet d'API facilitant le développement d'applications en tout genre. Cependant, jusqu'à récemment, JAVA péchait par excès de lenteur, rendant tout développement d'application 3D plus que problématique. Ceci n'est plus un problème avec l'apparition de « binding » OpenGL tels que LWJGL (<http://www.lwjgl.org>) ou encore JOGL (<https://jogl.dev.java.net>) permettant d'utiliser l'accélération matérielle propres aux cartes graphiques supportant le standard OpenGL.

Vu les difficultés propres au développement d'environnements virtuels 3D pour l'apprentissage, notre idée est donc de rendre accessible dans un futur proche l'API que nous développons dans le cadre de MOCAS (une fois que nous en aurons obtenu une version plus stable et avancée). C'est une attitude courante dans le domaine des jeux vidéo. Cela permet à des groupes de développeurs indépendants de créer des applications originales appelées MODS (pour Modifications). Ces MODS sont des dérivés d'une application mère qui vont se concentrer sur et/ou personnaliser un point particulier de cette application mère.

Ainsi la plupart des API utilisées dans MOCAS sont bien connues de la communauté STI. Nous utilisons entre autres la plate forme multi agents JADE (<http://jade.tilab.com>) pour le développement des agents, de leurs comportements et des communications entre ces agents, profitant en cela du support au *Agent Communication Language* fourni par JADE. Le système à base de règles JESS (<http://herzberg.ca.sandia.gov>) est utilisé pour la gestion des règles culturelles. Enfin, nous développons et gérons la partie 3D de notre système avec le moteur 3D JPCT (<http://www.jpct.net>, qui est basé sur LWJGL).

### Créer des cours dans le cadre de MOCAS

Comme on l'a remarqué auparavant, le développement d'environnements virtuels pour l'apprentissage, inspiré des jeux vidéo, est un processus particulièrement compliqué. De manière à réduire la quantité de travail nécessaire au développement de cours originaux au sein du système MOCAS, nous développons donc en parallèle un ensemble d'outils auteur que nous avons regroupé au sein du système IBIS : *Intuitive Builder for Intelligent Systems*. Le mot *intuitif* est particulièrement important ici. En effet, comme on l'a dit un peu plus tôt, la population d'auteurs cibles peut comprendre des individus qui auraient peu ou pas de connaissance en informatique (c'est encore fréquemment le cas d'enseignants de premier et second cycle). L'objectif est donc de concevoir un outil facile d'utilisation qui ne rebuterait pas de telles personnes. L'approche WYSIWYG est donc recherchée dans le développement de IBIS. Nous nous concentrons donc au final sur la création d'un système auteur simple (possédant sans doute moins de fonctionnalités que ses concurrents) plutôt qu'un système complet mais complexe. Cependant, il n'est pas exclu qu'une autre version de

notre système propose dans le futur plusieurs niveaux de complexité pouvant être sélectionnés par l'auteur.

### Créer des environnements 3D

Dans le processus de développement d'un jeu vidéo, l'un des premiers éléments conçu est bien souvent un éditeur de cartes. Les cartes sont cruciales dans les environnements virtuels car les informations qu'elles fournissent vont non seulement permettre de décrire les spécificités des environnements où se déroulera l'action de jeu mais elles seront aussi utilisées pour nombre les processus liés au fonctionnement du jeu (comme par exemple la fonctionnalité de recherche de chemin dont disposeront les différents agents présents dans le jeu). Lorsque l'on crée des environnements 3D, bien souvent les éditeurs de cartes se trouvent être en 2D. En suivant cette constatation, nous utilisons actuellement des programmes grand public de création d'images bitmap en tant qu'éditeurs de carte (voir figure 3).

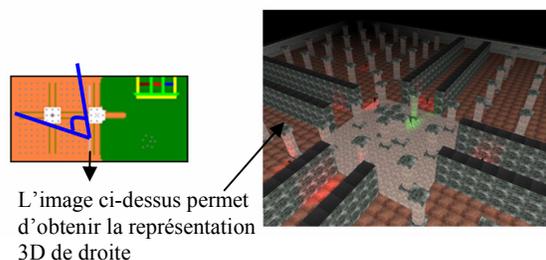


Figure 3: Création des environnements 3D

Les images classiques RGB proposent seize millions de combinaisons possibles par pixel. Un certain nombre de ces combinaisons sont déterminées comme codantes. A ce moment là, lorsque notre programme lit la valeur codante d'un pixel, il peut déterminer la représentation en 3D associée au code de ce pixel ainsi que le statut « peut être traversé / ne peut être traversé » de cette position dans l'environnement 3D (ce qui est par la suite utilisé par notre algorithme de recherche de chemin).

En définitif, une image bitmap est passée en paramètre lors de la création de l'agent de monde. Celui-ci génère alors l'environnement et transmet les informations obtenues aux agents d'interface qui disposent alors d'une représentation 3D de cet environnement.

### Créer le curriculum

Créer des environnements virtuels dans un but d'enseignement présente certaines particularités. La plus grande d'entre elles concerne sans doute la représentation des connaissances du domaine. Comment au juste représenter le curriculum ?

Comme on l'a dit auparavant, l'audience visée par IBIS peut comporter des personnes ayant peu ou pas de connaissances en informatique. En prenant en compte ce point, nous avons déterminé que la méthode des graphes conceptuels était une technique acceptable pour créer nos curriculum. Les graphes conceptuels ont l'avantage d'être une technique très graphique et assez intuitive. Ils présentent néanmoins un certain nombre limitations comparativement aux réseaux bayésiens ou aux

ontologies. Comme on l'a dit, dans la première version de notre système auteur IBIS, on cherche avant tout la simplicité d'utilisation. Cependant, notre méthodologie de développement aussi bien pour MOCAS que pour IBIS étant basée sur une approche par module, rien ne nous empêche d'envisager d'utiliser d'autres techniques de représentation des connaissances dans le futur. La figure 4 présente l'outil que nous avons développé.

Il permet deux vues différentes du curriculum :

- La vue « *graphe* » (à gauche de l'interface) montre une représentation du sous graphe du concept actuellement sélectionné. C.-à-d. que le graphe affiché contient tous les nœuds représentant les sous concepts qui pourraient être abordés lorsque l'on voudrait étudier le concept principal sélectionné.
- La vue « *arbre* » (en bas à droite de l'interface) montre une représentation globale des concepts définis dans le cadre de ce curriculum. Les concepts présents dans le sous graphe d'un autre concept y sont ici visibles comme étant les fils de ce concept. Par ailleurs, comme on va le voir dans ce qui suit, seuls les feuilles de l'arbre peuvent contenir les ressources pédagogiques utilisées dans les cours pour illustrer le concept parent de ces feuilles.

Lorsque l'on crée un nœud pour représenter un concept principal (représenté sous la forme d'un rectangle dans la fig. 4), deux nœuds enfants sont créés directement, ce qui signifie qu'un concept principal ne sera jamais une feuille (et ne contiendra donc jamais directement de ressource pédagogique). Le « *point d'entrée* » (ovale orange ou plus clair) sera le nœud de départ dans l'étude du concept principal tandis que le « *point de fin* » (ovale rouge ou plus foncé) sera le nœud à atteindre pour déterminer que l'étude du concept principal est complétée. Ces deux nœuds, les points

d'entrée et de fin, seront systématiquement des feuilles et pourront donc contenir des ressources didactiques (la plupart du temps respectivement pour introduire et conclure l'étude du concept). Ces ressources pourront par la suite être sélectionnées selon notre méthodologie d'adaptation culturelle lorsque viendra le temps de faire une session de cours concernant le concept principal. Si l'auteur veut disposer de nœuds intermédiaires contenant des ressources, il pourra créer des « *points de validation* » (représentés sous la forme d'hexagones dans la fig. 4). Bien entendu, comme on l'a signifié auparavant, on pourra également ajouter des nœuds représentant des sous concepts (qui eux même contiendront au minimum un *point d'entrée* et un *point de fin*).

De manière à mettre en évidence les capacités de navigation nœuds à nœuds au sein du sous graphe d'un concept, nous avons définis deux types de liens.

- Un *lien direct* (flèche verte ou plus claire) allant d'un nœud A à un nœud B signifie que le nœud B peut être étudié dès que l'étude du nœud A est complétée (sans autre pré requis).
- Dans le cas où un *lien secondaire* (flèche bleu ou plus foncée) relie un nœud C à un nœud D, la situation sera différente. Même si l'étude du nœud C est complétée, on devra s'assurer que l'étude des concepts de tous les autres nœuds aboutissant au nœud D par l'intermédiaire de liens secondaires aura été complétée avant d'entreprendre l'étude de D.

Notre outil permet de générer un fichier XML qui décrit la structure de notre curriculum. Celui-ci est utilisé d'une part par l'agent de monde et les agents pédagogiques pour déterminer et structurer la prochaine étape dans la session d'apprentissage. Ce fichier sert également à initialiser la partie du modèle de l'apprenant reliée aux connaissances de ce dernier,

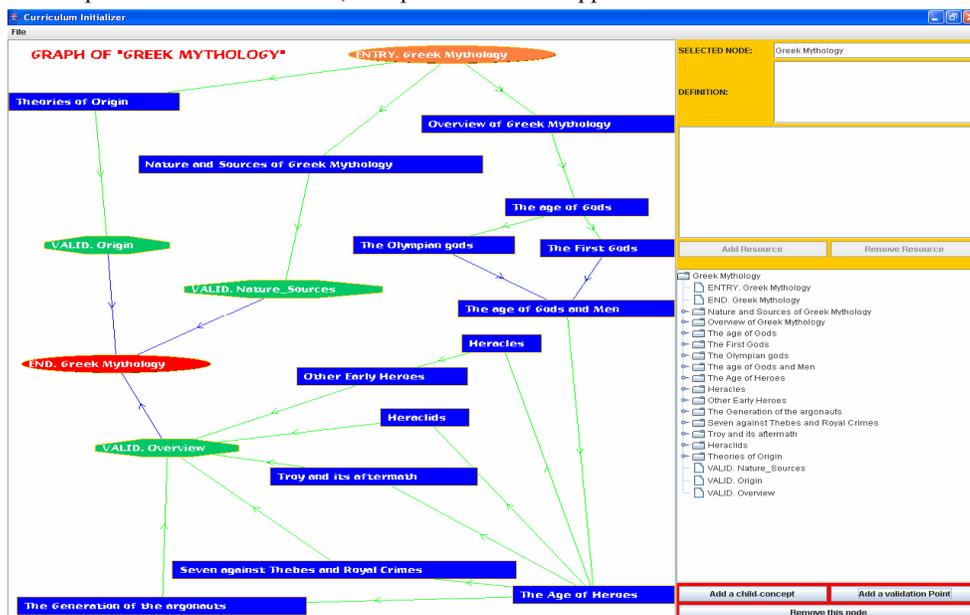


Figure 4 : Outil pour la création du curriculum

utilisant pour cela la *méthodologie de recouvrement* fréquemment utilisée dans les STI.

### Définir les agents pédagogiques et leur associer des connaissances

Finalement, le dernier élément important dans la création de cours pour MOCAS concerne la définition des agents pédagogiques qui vont peupler notre environnement 3D. La figure 5 présente l'outil que nous avons développé pour répondre à ce besoin.

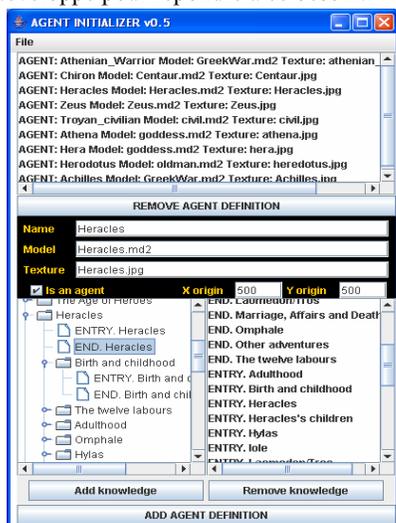


Figure 5 : Outil pour l'initialisation des agents pédagogiques

Dans le haut de l'interface, on trouve la liste des intitulés des définitions d'agents déjà réalisées. À côté du nom de l'agent créé figure le nom du modèle 3D ainsi que la texture qui lui sera appliquée de façon à obtenir l'avatar de l'agent dans l'environnement 3D.

En bas sur la gauche, on retrouve la vue « arbre » du curriculum. L'auteur peut y sélectionner des concepts et les associer à la définition de l'agent en cours de création. Comme on l'avait indiqué plutôt, les agents dans MOCAS se voient attribués des rôles différents, ce qui fait que leurs champs d'expertise diffèrent (ils ne seront donc pas aptes à enseigner tous les concepts du curriculum). Le nom des concepts sélectionnés (et pouvant être enseignés par cet agent) figure dans une liste en bas à droite de l'interface.

### Conclusion et travaux futurs

Dans cet article, nous avons introduit les principaux éléments de MOCAS, un système hybride mêlant STI jeu vidéo développé pour étudier l'effet du soutien à l'autonomie sur la motivation des apprenants. MOCAS présente aussi la particularité de chercher à prendre en compte les spécificités culturelles des apprenants en vue d'adapter la tâche d'apprentissage. Vu la complexité d'une telle application, plusieurs outils auteurs sont développés.

MOCAS est d'autre part développé selon une stratégie à base de modules de manière à faciliter l'intégration de nouvelles fonctionnalités dans le futur. L'amélioration du profil et des réponses culturels, la gestion

émotionnelle et la « scénarisation » des sessions de cours sont notamment à l'étude.

Pour finir, une évaluation de MOCAS dans des conditions réelles d'utilisation est actuellement en cours.

### Références

1. Blanchard, E., Frasson, C. (2004). An autonomy-oriented system design for enhancement of Learner's motivation in eLearning. 7<sup>th</sup> 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*, 2<sup>nd</sup> edition, London: Sage.
9. Jennings N. R. (2001). An introduction to multiagent systems. Tutorial of International Conference on Autonomous Agents, Montréal.
10. Johnson, W. L., Vilhjálmsón, H., Marsella, S. (2005). 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. Laguardia, J. G., Ryan, R. M. (2000). Buts personnels, besoins psychologiques fondamentaux et bien-être: théorie de l'autodétermination et applications. *Revue québécoise de psychologie*, vol 21(2), pp. 281-304.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.