

Assistance pédagogique et didactique en environnement virtuel de formation

Nicolas Marion*

Centre Européen de Réalité Virtuelle

Cédric Buche†

Centre Européen de Réalité Virtuelle

Ronan Querrec‡

Centre Européen de Réalité Virtuelle

Keywords: Réalité Virtuelle, Formation, Tuteur Intelligent, Didactique.

1 INTRODUCTION

De nombreux domaines de formation, tels que la conduite automobile ou la formation des pompiers professionnels, nécessitent la mise en situation des apprenants ; ceux-ci doivent acquérir non seulement des connaissances, mais encore de véritables compétences. Pour devenir progressivement efficace en situation, l'apprenant doit apprendre par l'action [11]. Or, la mise en situation des apprenants peut s'avérer coûteuse (d'un point de vue matériel) ou risquée (d'un point de vue humain). C'est le cas lorsqu'il s'agit d'apprendre à agir et à réagir face à des accidents, des événements peu prévisibles ou des dysfonctionnements.

La simulation informatique permet d'immerger les apprenants dans les environnements où ils peuvent essayer, choisir, prendre des initiatives, échouer et recommencer. Les environnements informatiques de formation utilisant la technologie de la *réalité virtuelle* (RV, cf. figure 1) sont particulièrement appropriés dans la mesure où ils proposent des situations à la fois réalistes, complexes et incertaines. L'ensemble de ces environnements virtuels peut être regroupé sous l'acronyme de EVF (Environnement Virtuel de Formation).



Figure 1: EVF pour la formation des pompiers (a), ou pour la formation du personnel sur porte-avions (b).

Dans le cadre de la formation professionnelle, l'instructeur n'est généralement pas un pédagogue mais un

*e-mail: marion@enib.fr

†e-mail: buche@enib.fr

‡e-mail: querrec@enib.fr

expert du domaine. Nous proposons donc de coupler l'EVF avec un tuteur intelligent (ITS pour *Intelligent Tutoring System*) qui aura pour but d'assister le formateur dans ses interventions pédagogiques lors de l'exercice.

En effet, la qualité de la construction des connaissances de l'apprenant dépend non seulement de la situation de simulation, mais également de la relation entre le formateur et l'apprenant. La figure 2 (tirée de [15]) présente les activités que peut avoir le formateur avant et pendant l'apprentissage. Comme nous pouvons le voir sur cette figure, les activités du

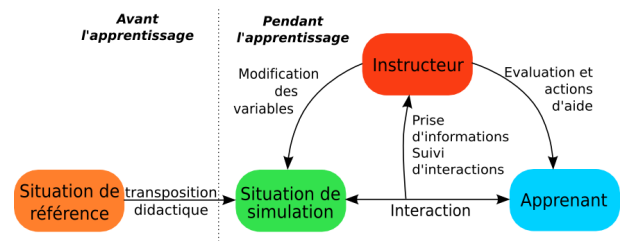


Figure 2: Activités du formateur pour la formation.

formateur se décomposent en deux principales catégories :

- Pendant l'apprentissage : le formateur analyse les interactions entre l'apprenant et la situation de simulation, puis, pour améliorer ces interactions, il peut soit modifier la simulation pour s'adapter à l'apprenant, soit aider l'apprenant pour qu'il s'adapte à la simulation.
- Avant l'apprentissage : le formateur (ou un didacticien) crée l'exercice en fonction des compétences visées par l'apprentissage

Dans la première partie de cet article, nous décrivons le modèle que nous utilisons pour représenter des environnements virtuels qui possèdent des connaissances qui leur permettent de raisonner sur la simulation (*Environnements Virtuels Informés*).

Dans la deuxième partie, nous présentons l'architecture du système tuteur intelligent développé au sein de notre équipe, capable de prendre des décisions pédagogiques pendant la simulation.

Dans la troisième partie, nous expliquons comment la didactique peut nous aider à créer un ITS capable d'assister le formateur dans la création de l'exercice.

2 ENVIRONNEMENT VIRTUEL INFORMÉ

La représentation de l'environnement virtuel que nous proposons réifie les éléments manipulés par un programme informatique pour effectuer un raisonnement. L'environnement virtuel contient alors des informations exploitables, il devient un “*Environnement Virtuel Informé*”.

Dans notre cas, l'environnement virtuel est un environnement virtuel de formation destiné à l'apprentissage de tâches procédurales et collaboratives. Le modèle doit donc non seulement réifier les connaissances portant sur l'environnement, mais également celles portant sur la tâche à effectuer par l'apprenant (connaissances sur la réalisation de procédures en équipe).

2.1 Représentation de l'environnement virtuel (VEHA)

Le modèle VEHA (Virtual Environment supporting Human Activity) actuellement développé au sein de l'équipe SPI (Simulation Participative et Immersive) du CERV permet de représenter explicitement un sous ensemble du méta-modèle UML [1] au sein de la simulation en réalité virtuelle. L'objectif de ce modèle est d'ajouter dans l'environnement les informations nécessaires aux comportements des agents comme dans les *Smart objects* [9] et permettant l'analyse de l'activité. Ainsi, les objets explicitent les notions :

- de géométries;
- d'actions (appelées dans notre modèle opérations);
- de points informés, utilisés par les opérations;
- de variables sous la forme d'attributs et de relations;
- de comportements, sous la forme de machines à états, qui utilisent les actions.

2.1.1 Structure de l'environnement (classes, objets...)

Le modèle VEHA réifie la notion de classe, c'est à dire qu'il définit la classe *Classe*. Une classe est composée d'attributs, d'opérations et de machine à états. VEHA réifie également les relations (héritages et associations) entre les classes. Il est donc possible de raisonner sur la notion de type d'objet indépendamment des objets eux-mêmes. Une classe ne sert qu'au raisonnement des agents, ce sont les instances de la classe qui constituent effectivement l'environnement.

Une classe dispose également d'opérations. Typiquement, une opération est qualifiée par son prototype et ses pré et post conditions. Cette représentation suffit aux agents pour la manipulation des opérations. Cependant, le modèle VEHA permet de représenter l'exécution de l'opération par un diagramme d'activités UML. Cela rend accessible dans l'environnement des informations permettant un raisonnement plus fin sur le déroulement d'une opération et donc une meilleure compréhension du rôle des opérations.

Nous avons expliqué précédemment que le modèle décrit en UML était interprété automatiquement pour créer l'environnement virtuel. Cette interprétation s'appuie sur le format XMI¹. La plupart des modeleurs UML permettent d'exporter le modèle créé en XMI (qui est un document XML décrivant le modèle UML). Le rôle de VEHA est simplement de traiter automatiquement ce fichier pour créer l'environnement. Cependant, le modèle créé représente la structure de la simulation (les classes et non les objets). Nous utilisons donc des outils externes permettant de placer à la souris les objets de l'environnement. Actuellement un outil de création des objets en liens avec le modèle est en cours de réalisation. De même un plugin pour 3DSMAX permettant d'exporter un fichier XML décrivant la liste des objets est également en cours de réalisation.

2.1.2 Dynamique de l'environnement (machines à états)

Nous venons de voir que le modèle VEHA permet de représenter la structure “statique” du système. Mais son rôle est également de décrire la “dynamique” de l'environnement. En effet, à chaque classe peut être attribuée une ou plusieurs machines à états. Cela signifie qu'il est possible de donner à une entité (à un objet) un moyen de décrire l'ensemble de ses états et les conditions de transitions entre ses états et que cette machine sera automatiquement interprétée dans la simulation. Cela permet, par exemple, de décrire les comportements réactifs des objets de l'environnement.

Dans VEHA comme en UML, une machine à états est composée d'états et de transitions. Un état peut désigner l'opération qui doit s'exécuter lorsque l'état devient actif ou lorsqu'il est actif et qu'il devient inactif. Une transition peut posséder une garde et être sensible à un évènement. Une garde est une expression booléenne qui sera évaluée en cours de simulation. VEHA (comme UML) propose un interpréteur de contraintes OCL pour décrire les gardes. Les évènements sont de quatre types : *callEvent* (une opération a été appelée), *signalEvent* (un signal a été reçu), *timeEvent* (un laps de temps s'est écoulé) et *changeEvent* (une condition est devenue vraie). Ces quatre types d'évènement sont gérés dans VEHA. Le fait que les états et les transitions soient explicitement représentés dans la simulation, permet aux agents de raisonner sur la manière de manipuler un objet. Il sera possible pour un agent de connaître la condition qui permet à un objet de transiter d'un état à un autre afin de réaliser une opération, à la charge ensuite au comportement de l'agent de modifier l'environnement pour que cette condition devienne vraie (par un chaînage arrière sur les pré et post conditions des opérations par exemple).

2.2 Représentation de la tâche (Procedural)

Le modèle VEHA présenté précédemment s'appuie sur le méta-modèle UML pour représenter l'environnement dans

¹XML Metadata Interchange : <http://www.omg.org/technology/documents/formal/xmi.htm>

lequel les procédures sont exécutées. Il est alors possible de décrire les procédures en parlant explicitement des objets de l'environnement (les ressources de la procédure par exemple). La description de ces procédures et des équipes (ou organisations) s'appuie sur le même principe d'une modélisation UML. Cependant, si les notions réifiées par VEHA existent déjà dans le méta-modèle UML (classes, associations...), ce n'est pas le cas des notions que l'on veut expliciter dans le cadre du travail procédural et collaboratif (procédures, rôles...). Nous proposons pour cela le modèle MASCARET [13] (MultiAgent System for collaborative, Adaptive, Realistic Environment for Training). Ce modèle décrit les concepts relatifs aux organisations et aux procédures en interprétant la sémantique du méta-modèle UML (cf. [10]).

Les organisations sont vues comme des structures organisationnelles qui fixent le cadre de réalisation des actions des agents autonomes par le biais des rôles que ces derniers jouent dans l'organisation. Le modèle d'organisation de MASCARET précise et rend implémentable des patterns de conception tel que le modèle AGR proposé par Ferber [6]. Les quatre concepts importants du modèle d'organisation sont : l'agent, l'organisation, le rôle et l'action. Ces quatre concepts sont représentés sur la figure 3 tirée de [13].

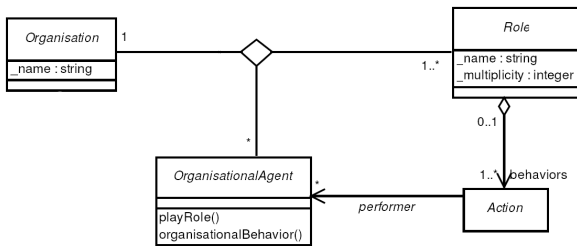


Figure 3: Diagramme de classes du modèle MASCARET.

L'agent est une entité de l'environnement (au même titre que les objets de l'environnement) douée de capacités de communication et d'un ensemble de comportements. Il n'est toutefois pas possible d'autoriser l'interaction avec un agent de la même manière qu'avec un objet, c'est à dire qu'on ne doit pas provoquer directement la réalisation d'une opération ou la modification d'un attribut de l'agent, seule la communication avec l'agent peut lui faire effectuer ces modifications. L'organisation est explicitée pour permettre le raisonnement des agents sur la structure et permet de différencier une structure organisationnelle d'une entité organisationnelle (instance d'une structure organisationnelle). Un rôle est un ensemble d'actions que l'agent peut avoir à effectuer s'il joue ce rôle. Enfin, une action est, dans le cadre d'un environnement virtuel, la réalisation d'une opération sur l'agent lui-même ou sur un objet de l'environnement, la réalisation d'une animation et/ou d'un son ou uniquement la consommation de temps.

Une équipe (classe dérivant d'organisation) est décrite par sa structure organisationnelle (les différents rôles qui composent l'équipe), et de l'ensemble des procédures qu'elle peut réaliser (catapultage d'un RAFALE par exemple). Une procédure est décrite et interprétée à l'aide des diagrammes d'activités d'UML. Le modèle MASCARET propose donc un modèle de procédure s'appuyant sur le modèle des diagrammes d'activités d'UML (cf. [10]). Ce type de diagramme est suffisamment expressif pour décrire l'ensemble des cas d'agencement possibles et suffisamment clair pour servir au dialogue entre l'informaticien et l'expert. Une procédure est structurée en rôles (correspondant aux rôles de l'organisation), ces rôles peuvent être joués par des types d'agents. Les rôles de la procédure sont déduits des couloirs d'activités décrits dans le diagramme d'activités. Au cours de ces procédures, des ressources externes interviennent (l'aéronef à catapulter, le holdback à utiliser...) ces ressources sont représentées par des objets de l'environnement et sont également déduites de l'interprétation du diagramme d'activités (*ObjectNodes*). Une procédure agence les actions des rôles, c'est à dire qu'elle décrit l'enchaînement logique et temporel des actions. Pour cela, plusieurs opérateurs sont utilisés dans un diagramme d'activités tels que le parallélisme, la synchronisation ou les branchements conditionnels. Tous ces opérateurs sont gérés dans le modèle MASCARET.

Dans cette section, nous avons vu comment notre modèle permet de réifier les notions de la simulation pour permettre le raisonnement sur celles-ci. Dans la section suivante, nous expliquons comment un système tuteur intelligent peut utiliser ces informations pour simuler un raisonnement pédagogique.

3 SYSTÈME TUTEUR INTELLIGENT

L'objectif ici est de proposer un système tuteur intelligent (ITS) qui fournit une aide pédagogique à l'apprenant et une assistance pédagogique au formateur.

Nous soutenons ici qu'il est possible d'intégrer un ITS qui soit :

- *générique*. Plus précisément, notre modèle propose une représentation abstraite, indépendante de l'exercice à réaliser. Cette caractéristique est primordiale puisque nous ne voulons pas redévelopper une application pour chaque exercice (les environnements de réalité virtuelle sont complexes et nécessitent des temps de développement considérables).
- *adaptatif*. Le modèle s'adapte au couple apprenant-formateur en modifiant son comportement pédagogique. Ainsi, le système propose des assistances pédagogiques (mise en transparence d'objets, ajouts d'explications...) en tenant compte des expériences passées. Une assistance pédagogique non pertinente pour une situation donnée sera dévaluée et vice versa.

Nous utilisons ici les modèles de l'environnement informé définis dans la partie précédente. Les connaissances représentées par ces modèles sont manipulables, elles peuvent alors être utilisées pour la pédagogie. Il reste à représenter les connaissances qui portent sur l'apprentissage lié aux activités de l'apprenant. Ces connaissances sont alors utilisées pour simuler un comportement pédagogique et proposer des assistances pédagogiques au formateur.

Les connaissances qui portent sur l'apprentissage lié aux activités de l'apprenant sont contenues dans des modèles inspirés de ceux qui sont utilisés classiquement dans un ITS [16]. Notre proposition définit les composantes d'un ITS sous la forme d'un système multi-agents [2]. Les agents ont la responsabilité des modèles classiques des ITS. Ainsi, *ExpertAgent*, *LearnerAgent*, *ErrorAgent*, *TeacherAgent* et *InterfaceAgent* ont respectivement la charge du modèle du domaine, de l'apprenant, des erreurs, du formateur et d'interface. Le système est un processus en cinq étapes

Étape 1. *Observer* :

Utilisant le modèle d'interface, le système analyse les activités de l'apprenant. Les éléments pertinents pour la formation sont fournis au modèle de l'apprenant. Ces informations concernent les actions de l'apprenant, les éléments observables par l'apprenant et les déplacements de l'apprenant.

Étape 2. *Détecter et identifier une erreur* :

Le système analyse les actions de l'apprenant (modèle de l'apprenant) et les compare aux actions à effectuer (modèle du domaine). Cette confrontation permet de détecter une éventuelle erreur. Si une erreur a été détectée, un mécanisme d'identification de l'erreur est mis en place (en utilisant le modèle des erreurs).

Étape 3. *Proposer des assistances pédagogiques* :

Utilisant le modèle de l'apprenant (caractéristiques, activités, erreur, *etc.*) et le modèle du domaine (connaissances sur les structures organisationnelles), un mécanisme simulant un raisonnement pédagogique préconise les assistances pédagogiques adaptées à la situation. Cette étape n'est pas conditionnelle, elle intervient même si aucune erreur n'est détectée.

Étape 4. *Choisir une assistance pédagogique* :

Le formateur peut sélectionner une assistance pédagogique parmi celles qui ont été préconisées.

Étape 5. *Représenter l'assistance pédagogique* :

L'assistance pédagogique sélectionnée est représentée dans l'environnement virtuel.

Ce système multi-agents fournit une représentation des activités de l'apprenant. Ainsi, le système construit dynamiquement un ensemble d'informations relatives à la tâche à effectuer en considérant l'avancée de

l'apprenant dans la procédure (actions réalisées, erreurs, objets perçus...). L'ensemble de ces connaissances (relatives à l'apprenant et à son avancée dans la procédure) forme la "situation pédagogique". Cette dernière sert de base de connaissances au raisonnement effectué par l'agent pédagogique (*PedagogicalAgent*).

La situation pédagogique offre la possibilité de déclencher des assistances pédagogiques sur les éléments qu'elle contient, elle fournit alors les sorties possibles de la prise de décision pédagogique. Il s'agit à présent de définir un modèle simulant le comportement décisionnel de l'agent pédagogique qui fournit les propositions d'assistance pédagogique, c'est à dire un modèle qui fait le lien entre les connaissances et les propositions d'assistances. Nous devons considérer

- l'hétérogénéité et la nature des connaissances impliquées (connaissances issues de la pédagogie fondamentale jusqu'à la réalité virtuelle);
- les capacités d'adaptation (le raisonnement doit s'auto modifier pour prendre en compte les expériences passées);
- le raisonnement doit pouvoir être spécifié *a priori* (la spécification initiale peut alors être réalisée par un pédagogue).

Après nous être intéressés aux familles d'architectures comportementales existantes (connexionistes, à base d'automates, à base de règles), nous avons opté pour celles qui utilisent des règles. Plus précisément, nous avons choisi les systèmes de classeurs [3] qui répondent au mieux aux critères que nous avons retenus dans le cadre de notre étude (expressivité, hiérarchisation, modularité, réactivité et adaptabilité). Il s'agit d'une architecture réactive et adaptative, fondée sur des règles conditionnelles.

Nous proposons un modèle basé sur un système de classeurs hiérarchiques. Il organise les connaissances en considérant l'abstraction des données mises en jeu. Il structure les connaissances sur trois niveaux, allant des règles basées sur des connaissances abstraites de la pédagogie (les démarches pédagogiques) jusqu'aux règles fondées sur des connaissances concrètes de la réalité virtuelle (les techniques pédagogiques), en passant par un niveau intermédiaire (les attitudes pédagogiques).

Chaque niveau d'abstraction contient des ensembles qui regroupent plusieurs règles. Un ensemble représente une façon d'aborder une démarche, une attitude ou une technique pédagogique. Les règles sont conditionnées par les éléments de la situation pédagogique et ont pour effet de favoriser des ensembles du niveau inférieur. Le système utilise alors un mécanisme de diffusion dans les trois niveaux qui considère les règles appariées par la situation pédagogique. Il aboutit à une liste qui ordonne les propositions d'assistance pédagogique.

La figure 4 illustre la structure et la dynamique du modèle pédagogique contrôlant le comportement de l'agent

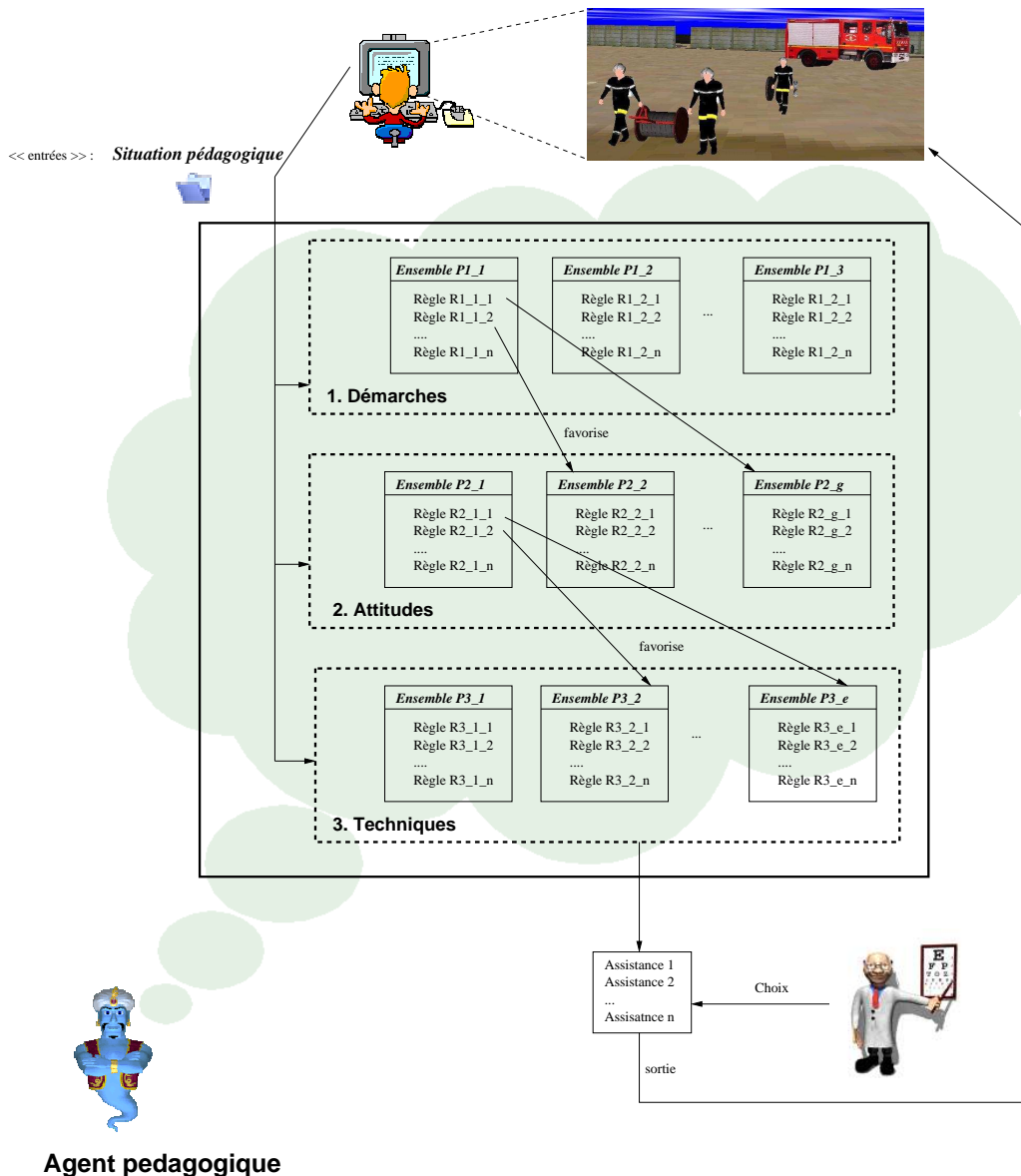


Figure 4: Représentation complète du modèle pédagogique.

pédagogique. Les informations prises en considération dans les parties conditions des règles, sont procurées par notre ITS (situation pédagogique). Ces “entrées” sont disponibles sur les trois niveaux d’abstraction des données (démarches, attitudes et techniques pédagogiques). Les règles dont la partie condition est satisfaite par rapport aux entrées, favorisent certains ensembles de règles pédagogiques du niveau inférieur. Le dernier niveau (techniques) favorise directement des assistances pédagogiques applicables dans l’environnement. Ces dernières sont proposées au formateur qui choisit celle qui est, selon lui, la plus appropriée.

Enfin, nous intégrons un mécanisme d’apprentissage artificiel, de type *bucket brigade*, qui adapte le comportement décisionnel de l’agent pédagogique au couple apprenant-formateur. Il est fondé sur un renforcement provenant des

choix du formateur. Ainsi, au fur et à mesure des exercices, l’agent pédagogique doit proposer des choix de plus en plus en accord avec les décisions du formateur. L’agent pédagogique pourrait alors, dans le cadre d’un apprentissage multi-apprenants, prendre le relais temporairement et appliquer directement les assistances qu’il a choisies.

Simuler un raisonnement pédagogique présente un double intérêt:

1. les formateurs ne sont pas toujours des pédagogues, nous leur proposons alors une assistance pédagogique;
2. les formateurs ne sont pas des experts dans les logiciels de simulation, l’agent pédagogique va proposer à l’apprenant des aides qui exploitent au mieux les capacités de la réalité virtuelle.

4 ASPECT DIDACTIQUE

Nous venons de voir que l'utilisation d'un ITS permet d'assister le formateur pendant la session d'apprentissage en lui proposant des assistances pédagogiques pertinentes en fonction du contexte. Si l'on reprend la figure 2 vue en introduction, on constate que le tuteur intelligent que nous avons développé est utile pendant l'apprentissage. Ceci suppose que les exercices destinés aux apprenants sont déjà conçus. Or, une partie importante d'un apprentissage consiste pour le formateur à créer des situations d'apprentissage en s'assurant que celles-ci mettent en oeuvre l'acquisition des compétences visées.

4.1 La transposition didactique

Pour présenter la problématique de la création d'une situation d'apprentissage, nous allons commencer par présenter la démarche du didacticien en nous appuyant sur la notion de transposition didactique telle qu'elle est présentée par Perrenoud dans [12].

La transposition didactique, devenue d'usage courant dans les didactiques disciplinaires (notamment dans le domaine des mathématiques, grâce aux travaux de [4]), a pour but de transformer les *savoirs savants* en *savoirs à enseigner*, puis les *savoirs à enseigner* en *savoirs enseignés* (cette dernière étape n'est pas toujours incluse dans le processus même de transposition didactique). Depuis quelques années, plusieurs chercheurs défendent la théorie selon laquelle la transposition didactique n'est pas exclusive aux savoirs savants, et que la théorie de la transposition peut donc être étendue aux savoirs professionnels [14].

Selon Perrenoud, "*la transposition didactique n'est pas un pur parcours du savoir, mais passe par des situations et des pratiques qui ne contiennent pas les savoirs, mais en permettent la reconstruction par chaque apprenant*". La Figure 5 décrit sa vision de la transposition didactique, ou comment, à partir de pratiques, créer des exercices permettant la construction des connaissances-compétences chez l'apprenant. Comme nous pouvons le voir sur cette figure, la toute première étape de la transposition consiste en une analyse précise des situations de travail (ou équivalent) de référence. Cette étape est une des plus délicates à réaliser car les pratiques sont souvent constituées d'éléments implicites (non-dits, censures, embellissements...). À partir de cette description, le didacticien doit déterminer les compétences utilisées dans la situation de référence, puis identifier dans ces compétences les *ressources cognitives mobilisées* et les *schèmes opératoires* mettant en oeuvre ces ressources. Une fois les compétences identifiées, il reste à saisir comment se construisent à la fois les ressources cognitives nécessaires et leurs schèmes de mobilisation. Enfin, les dernières étapes résultent en la création d'un curriculum formel de la formation (objectifs pédagogiques, stratégies de formation...), puis d'un curriculum réel qui décrit les exercices effectivement proposés aux apprenants.

Les deux dernières étapes de cette chaîne de transposition

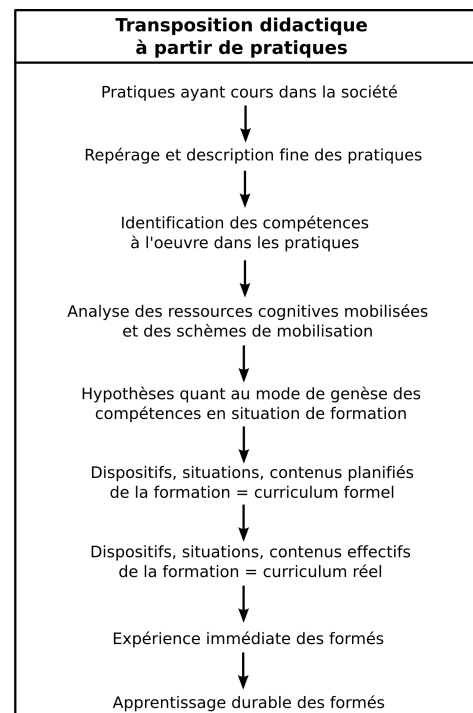


Figure 5: Chaîne de transposition didactique à partir de pratiques.

concernant la construction des connaissances à court et long terme ne nous intéressent pas ici, notre but étant simplement d'arriver à la création d'un curriculum réel.

4.2 Un tuteur intelligent didactique

La démarche de transposition didactique que nous venons de décrire est bien entendu une démarche très spécifique à la formation que l'on souhaite mettre en place. En effet, la mise en place d'une situation d'apprentissage (un curriculum réel) dépend de nombreuses variables (situation de référence, objectif de la formation, cursus des apprenants...) si bien qu'il est nécessaire de recommencer entièrement le processus de transposition dès que l'on veut mettre au point un nouveau dispositif de formation.

L'approche informatique quant à elle, est par nature très générique, le but étant d'automatiser les tâches en limitant les interventions humaines. Automatiser la transposition didactique dans son ensemble apparaît rapidement comme étant quelque chose d'impossible. Il est en effet très peu concevable qu'un programme informatique réunisse toutes les qualités d'analyse nécessaires par exemple à l'identification des compétences mises en oeuvre dans une activité ou encore des ressources cognitives mobilisées. Notre but est donc de créer un tuteur ayant une capacité didactique générique, spécifique à un domaine précis.

Des psychologues de l'équipe SARA (Simulation, Apprentissage, Représentation, Action) du CERV travaillent actuellement sur l'analyse de la formation de pompiers professionnels aux risques chimiques et biologiques sur quatre

niveaux différents. Le but de cette analyse est d'identifier les compétences requises et les invariants mobilisés pour ces situations. L'objectif que nous poursuivons est d'abstraire les résultats de ces travaux pour pouvoir les appliquer à toute une classe de situations : la formation à des tâches procédurales et collaboratives. Cela permettrait de capitaliser les résultats pour les quatre premières étapes de la démarche de transposition représentée sur la Figure 5. Nous partons de l'hypothèse que les compétences entrant en jeu sont les mêmes, dans différentes situations procédurales et collaboratives, et donc que la construction de ces compétences peut s'effectuer de manière similaire.

Une fois cette première étude menée à son terme, nous proposerons si les résultats sont concluants, un modèle capable de décrire un objectif pédagogique dans ce cadre de formation ainsi qu'un outil permettant au formateur d'exprimer l'objectif pédagogique qu'il veut atteindre. À partir de là, le tuteur intelligent devra être capable d'assister le formateur dans la création du curriculum réel (contenus effectifs de la formation).

Pour créer les exercices de la formation, cela suppose que l'on dispose d'un modèle capable de décrire ces exercices. Dans la section qui suit, nous allons étudier des modèles existants dans les EIAH pour décrire des exercices de formation.

4.3 Modéliser un exercice

Lorsque l'on veut créer une simulation pour la formation, on doit disposer d'un modèle capable de décrire les exercices qui vont être proposés aux apprenants. Ce modèle est souvent appelé scénario pédagogique, et il a pour but de décrire les activités qui seront proposés aux apprenants sur la simulation et le contrôle pédagogique associé à ces activités. Ces scénarios pédagogiques se présentent souvent sous des formes peu explicites pour les non-informaticiens (fichier XML...). C'est pourquoi la conception d'exercices en utilisant un tel modèle se fait souvent en utilisant un outil-auteur, permettant de concevoir à la fois la partie simulation, la partie pédagogique et le lien entre les deux (cf. [8] pour une présentation et une discussion sur les environnements auteurs). Nous ne parlerons pas ici de la conception d'un environnement auteur dans son ensemble, nous nous intéresserons uniquement au modèle de scénario pédagogique qu'ils utilisent.

Nous décrivons à présent deux modèles de scénario pédagogiques, tout d'abord le modèle utilisé par la norme IMS-Learning Design, puis celui intégré dans l'outil-auteur OASIS [5].

4.3.1 IMS-Learning Design

La spécification IMS-Learning Design (IMS-LD) permet de structurer les situations d'apprentissage en unités d'apprentissage. Ce modèle place en son centre les activités proposées aux apprenants plutôt que les objets manipulés. Il permet de décrire les activités de l'apprenant ainsi que des membres celles de l'équipe pédagogique.

La figure 6 présente schématiquement la structuration du modèle d'IMS-LD. Sur ce schéma sont représentées les

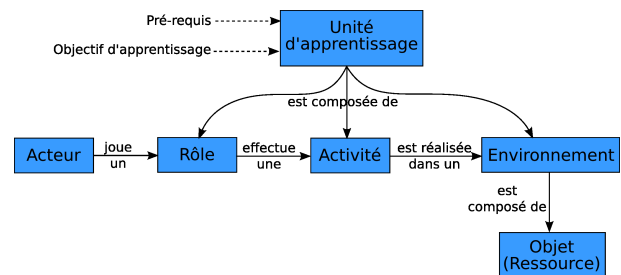


Figure 6: Structuration du modèle d'IMS-LD.

différentes contraintes auxquelles doit répondre une unité d'apprentissage :

- une unité d'apprentissage peut nécessiter un certain nombre de pré-requis;
- une unité d'apprentissage correspond à un objectif pédagogique;
- une unité d'apprentissage est composée d'un ensemble d'activités;
- une activité est réalisée par un ou plusieurs acteurs tenant chacun un rôle;
- une activité est réalisée au sein d'un environnement lui donnant accès à un ensemble de ressources.

Un point fort de ce modèle est qu'il peut aussi bien décrire un apprentissage informatisé ou non. En effet, les ressources de l'environnement peuvent décrire des documents physiques ou numériques, ou encore des outils nécessaires à la réalisation de l'activité.

4.3.2 Scénario pédagogique d'OASIS

OASIS [5] est un outil-auteur utilisé notamment dans le projet FORMID [7]. Le principe d'OASIS est de créer un scénario pédagogique indépendant de l'outil pédagogique interactif (OPI, qui peut être une simulation, un micro-monde...), et qui puisse donc se coupler à n'importe quel OPI grâce à un moniteur de scénario. Ce qui nous intéresse ici est plus précisément la forme que prend ce scénario pédagogique. Tout comme le modèle d'IMS-LD que nous avons vu précédemment, le scénario pédagogique d'OASIS définit l'activité proposée aux apprenants ainsi que le contrôle de la progression des apprenants durant cette activité. De plus, le modèle permet de préciser des assistances pédagogiques fournies automatiquement aux apprenants en fonction de leur progression. Un scénario est défini par son titre, l'énoncé général, et les différentes étapes de résolution (une au minimum). Chaque étape est considérée comme un sous-objectif à atteindre et l'apprenant doit valider successivement tous les objectifs. Un retour d'information global

est prévu en cas de réussite ou d'échec de l'exercice. Chaque étape est définie par :

- un libellé qui rappelle l'objectif principal de l'étape;
- un énoncé qui constitue la consigne spécifique à l'étape;
- un état initial, exprimé en termes de valeurs de variables, qui sert au moniteur de scénario pour initialiser l'OPI;
- un ensemble éventuellement vide de situations à observer, décrites par un ensemble de valeurs de variables ou par une occurrence d'évènement. Ces situations ne constituent pas forcément une réussite ou un échec, mais permettent au formateur de spécifier des états dans lesquels le système doit fournir une assistance automatique à l'apprenant, ou simplement le prévenir que l'apprenant a atteint un certain point.

5 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons présenté notre approche de la conception d'environnements virtuels de formation pour l'apprentissage de tâches procédurales et collaboratives. Dans la première partie, nous avons commencé par décrire VEHA et MASCARET, les modèles qui nous permettent de créer des environnements virtuels dans lesquels toutes les notions de la simulation sont représentées explicitement.

Dans la seconde partie, nous avons proposé un modèle de Système Tuteur Intelligent (ITS) utilisant les modèles vu précédemment. Ce tuteur est capable de proposer des assistances pédagogiques pertinentes au formateur en fonction du contexte de l'apprentissage en cours.

Enfin, dans la dernière partie, nous avons présenté la manière dont la didactique peut permettre à l'ITS d'assister le formateur dans la création de situations de simulation, notamment par la création d'un modèle d'objectif pédagogique.

Actuellement, nos travaux portent principalement sur ce dernier point. À court terme, notre premier objectif sera de créer un modèle de scénario pédagogique (cf. section 4.3). Ensuite, nous analyserons les résultats des travaux de notre équipe de psychologues sur la formation des pompiers afin d'en abstraire un modèle d'objectif pédagogique utilisable par notre ITS.

REFERENCES

- [1] G. Booch, J. Rumbaugh, and I. Jacobson. *The unified modeling language user guide*. Addison-Wesley Reading Mass, 1999.
- [2] C. Buche, R. Querrec, and C. Le Gall. Intelligent tutoring system for procedural and collaborative training. In *8th World Conference on Computers in Education*, 2005.
- [3] C. Buche, C. Septseault, and P. De Loor. Les systèmes de classeurs. une présentation générale. *Revue des Sciences et Technologies de l'Information, série Techniques et Sciences Informatiques (RSTI-TSI)*, 25:963–990, Décembre 2006.
- [4] Y. Chevallard. *La transposition didactique: du savoir savant au savoir enseigné*. Pensée sauvage, 1985.
- [5] G. Cortés and V. Guéraud. Experimentation of an authoring tool for pedagogical simulations. *International Conference on Computer Aided Learning and Instruction in Science and Engineering (CALISCE'98)*, Göteborg, Sweden, 1998.
- [6] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *ICMAS '98*. IEEE Press, 1998.
- [7] V. Guéraud, J.-M. Adam, J.-P. Pernin, G. Calvary, and J.-P. David. L'exploitation d'objets pédagogiques interactifs à distance : le projet formid. *Revue STICEF*, 11:103–163, May 2004.
- [8] M. Joab, V. Guéraud, and O. Auzende. *Environnements informatiques pour l'apprentissage humain*, chapter 13, pages 287–310. Hermès, 2006.
- [9] M. Kallmann. Interaction with 3-d objects. In N. Magnenat-Thalmann and D. Thalmann, editors, *Handbook of Virtual Humans*, chapter 13, pages 303–322. John Wiley & Sons, first edition, 2004.
- [10] N. Marion. Méta-modélisation uml pour les environnements virtuels procéduraux et collaboratifs. Master's thesis, UBO-ENIB, June 2006.
- [11] A. Nguyen-Xuan. Les modèles d'analyse de l'apprentissage. *Revue Française de Psychologie*, (112):57–67, 1995.
- [12] P. Perrenoud. La transposition didactique à partir de pratiques : des savoirs aux compétences. *Revue des sciences de l'éducation*, 24(3):487–514, 1998.
- [13] R. Querrec. *Les systèmes multi-agents pour les environnements virtuels de formation. Application à la sécurité civile*. PhD thesis, UBO-ENIB, 2002.
- [14] J. Rogalski and R. Samurçay. Modélisation d'un "savoir de référence" et transposition didactique dans la formation de professionnels de haut niveau. *La transposition didactique à l'épreuve*, Grenoble, La Pensée sauvage Éditions, pages 35–71, 1994.
- [15] R. Samurçay and J. Rogalski. Exploitation didactique des situations de simulation. *Le travail humain*, 61(4):333 – 359, 1998.
- [16] E. Wenger. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann, Los Altos, California, 1987.