

Spécifications et architecture d'un environnement Web supportant des activités coopératives d'apprentissage

Thierry Nodenot, Christophe Marquesuzaà, Pierre Laforcade, Marie-Noelle Bessagnet, Christian Sallaberry

▶ **To cite this version:**

Thierry Nodenot, Christophe Marquesuzaà, Pierre Laforcade, Marie-Noelle Bessagnet, Christian Sallaberry. Spécifications et architecture d'un environnement Web supportant des activités coopératives d'apprentissage. Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie, Nov 2002, Villeurbanne, France. pp.253-262. edutice-00000663

HAL Id: edutice-00000663
https://edutice.hal.science/edutice-00000663

Submitted on 13 Oct 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Spécifications et architecture d'un environnement Web supportant des activités coopératives d'apprentissage

Thierry Nodenot [(1)], Christophe Marquesuzaa [(1)], Pierre Laforcade [(2)], Marie-Noëlle Bessagnet [(2)], Christian Sallaberry [(2)]

Laboratoire d'Informatique de l'UPPA (LIUPPA)

[(1)]IUT de Bayonne Pays Basque
Château-Neuf, Place Paul-Bert
64100 Bayonne – France

[(2)]FDEG et IAE
Avenue du Doyen Poplawski
64000 Pau – France

contact : Thierry.Nodenot@iutbayonne.univ-pau.fr - *http://liuppa.univ-pau.fr*

Résumé

Nous présentons dans cet article nos travaux sur la spécification d'environnements permettant à un pédagogue de mettre en œuvre des situations coopératives d'apprentissage. Après avoir décrit les enjeux, les acteurs et les principes de mise en œuvre d'activités coopératives d'apprentissage, nous focalisons notre attention sur la définition d'un système (appelé Learning Management System) capable de gérer ce type d'activités pour une communauté d'utilisateurs (apprenants et moniteurs). Nous montrons ainsi qu'une pédagogie à base d'activités coopératives est envisageable sur la base des travaux de normalisation menés par les consortiums internationaux centrés sur l'E-Learning.

Abstract

In this paper, we present our works dedicated to the specification of environments enabling a community of users to do some cooperative learning activities. We first describe the goals, key players and principles of such cooperative and learning activities. Then, we focus the problem of defining the characteristics of a system (called a Learning Management System) that could manage such cooperative learning activities. By the way, we argue that a constructivist way of learning is reachable on the basis of normalisation works that are led by the international consortiums centred on e-learning.

Introduction

L'ingénierie éducative a évolué ces dernières années dans sa *dimension technologique*. Nous en voulons pour preuve les travaux de recherche actuels pour la formation ouverte à distance. Ceux-ci sont centrés sur l'établissement de standards pour les méta données et les objets d'apprentissage. Des organismes comme [LTSC, 2001], [LOM, 2000], [PROMETEUS, 2001] ou [IMS, 2001], des sociétés commerciales comme Asymetrix, Oracle ou encore Microsoft, visent une meilleure interopérabilité entre applications éducatives et une meilleure réutilisation d'applications et services « électroniques ».

D'autres recherches s'intéressent davantage à la *spécification pédagogique* (aspect modélisation) comme point central d'un environnement d'apprentissage. L'objectif est de proposer des langages de haut niveau permettant de guider les pédagogues concepteurs dans la spécification d'un tel environnement. De nombreux travaux s'accordent pour considérer qu'un environnement d'apprentissage est avant tout un Système d'Informations [Rodriguez-Artacho & all, 1999], [Koper, 2001], [Nodenot, 2001]. Il s'agit donc d'étudier un tel environnement dans sa complexité en prenant en compte ses spécifications (pédagogiques et fonctionnelles) mais aussi sa dynamique, sa mise en œuvre et son exploitation.

Dans cet article, nous allons étudier ces deux dimensions, technologique et pédagogique, sur un cas particulier : *le développement d'applications éducatives à base d'activités coopératives entre apprenants*. Nous décrirons dans un premier temps ce que nous entendons par activité coopérative d'apprentissage en centrant notre propos sur les fonctions que doit proposer un environnement support pour ce type d'activités (dimension pédagogique). Dans une seconde partie, nous décrirons comment un tel environnement peut être implémenté en respectant notamment les principes proposés par les principaux organismes de normalisation en matière d'E-Learning (dimension technologique).

1. Spécifications fonctionnelles d'un environnement support pour des activités coopératives d'apprentissage

Préparer, enseigner, évaluer sont les trois étapes primordiales dans la mise en œuvre d'une situation pédagogique. Ce processus doit prendre en compte divers paramètres comme les conditions d'enseignement, les objectifs pédagogiques, la rétroaction nécessaire,

l'adaptation à une modification de la situation pédagogique, etc. Quand la situation pédagogique est « médiatisée » via un réseau d'ordinateurs, l'enseignant doit garder un rôle prépondérant d'animation, de guidage, et ce, même si les conditions d'exploitation changent :

- Les apprenants ne sont pas forcément physiquement dans le même lieu,
- L'activité des apprenants est perçue au travers d'une médiatisation par l'outil informatique,
- L'activité de l'enseignant dans ses rôles de suivi, de maîtrise, de contrôle et d'orientation des activités des apprenants est également médiatisée par l'outil informatique.

Par contraste avec l'apprentissage traditionnel, basé sur la notion de cours, de lectures et d'exercices, l'apprentissage collectif (qu'il soit coopératif ou collaboratif) [George, 2001] impose que les apprenants participent activement à la production, à l'accès et l'organisation de l'information. L'instructeur structure soigneusement les activités des apprenants, met l'accent sur un contenu particulier et contrôle le travail de l'étudiant. La construction de connaissances est donc perçue comme un processus de résolution progressive de problèmes, qui encourage des étudiants à être novateurs, à créer des aptitudes intellectuelles et à développer et acquérir de l'expertise. Ceci est démontré par [Harasim, 1999] qui dresse un bilan d'exploitation d'un tel système : Virtual-U [Sawyer, 2000].

Dans nos travaux, le point de départ d'une activité coopérative est la situation-problème (PBLS) [Develay, 1993], [Meirieu, 1994]. [Suthers, August 2001] note : "Collaborative learning based on the notion of PBLS is an interesting approach and quite different from more classic sequences: courses, exercises, evaluation". Cet objet pédagogique inclut tous les éléments nécessaires pour que le groupe classe (et ses sous-groupes constitutifs) se mette en situation de projet avec :

- Un objectif global, des critères de réussite fonctionnel et formel,
- Des matériels et autres outils facilitant la résolution de la situation problème,
- Un guide décrivant certaines modalités et autre répartition des tâches entre acteurs (et groupes d'acteurs).

[Marquesuzaà, 1998] a décrit cet objet d'apprentissage particulier qu'est la situation-problème et proposé un environnement d'outils permettant à un enseignant de créer et manipuler un tel objet. [Wiley, 2000] résume de façon consensuelle ce qu'est un objet d'apprentissage (dans la littérature, certains auteurs emploient le terme d'objet pédagogique [De La Passardière & Giroire, 2001] ou encore « *learning object* ») : « [il] *représente toute ressource numérique (quelle que soit sa taille et son type) pouvant être réutilisée comme support d'apprentissage (à travers le réseau ou non)* ».

Partant de ces travaux, nous proposons un environnement au sein duquel l'enseignant a la possibilité de définir un projet global éducatif à base de situations-problèmes :

- Plusieurs groupes d'apprenants disposant de rôles distincts doivent coopérer pour résoudre une situation problème et atteindre les objectifs pédagogiques attachés à cette situation.
- Le système intègre un autre type d'acteur : un moniteur qui est chargé de suivre et d'assister les apprenants dans leur résolution de problème. Ce moniteur peut être humain (personne physique) ou automatique. Il dispose donc de ressources et de services idoines et peut interagir avec le système :

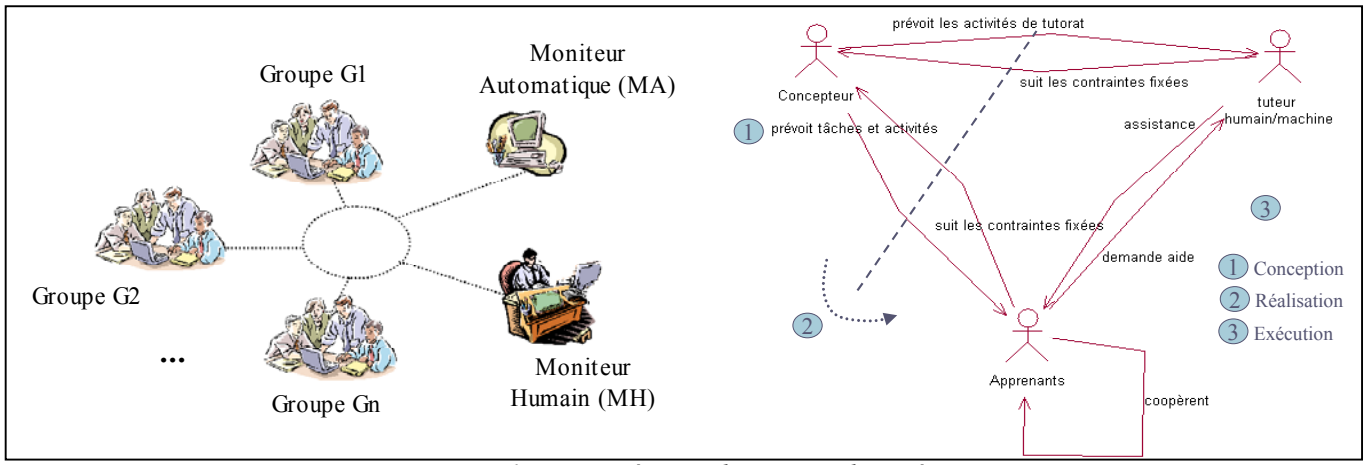


Figure 1 : La coopération des acteurs du système

Chacun des groupes d'acteurs présenté mène, au service de la collectivité, un certain nombre d'activités au sens de [Bourguin, 2000]. Ces activités dont les caractéristiques sont définies en phase de conception pédagogique sont enchaînées par les acteurs de manière synchrone ou asynchrone [Laforcade, 2002]:

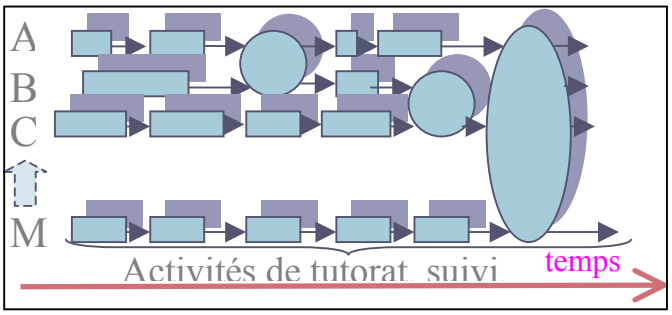


Figure 2 : La synchronisation des activités menées par les acteurs du système

Ces activités se caractérisent par un certain nombre d'invariants :

- Toute activité met à disposition des acteurs qui la mènent un ensemble de ressources en lecture et définit un ensemble de ressources à produire,
- Toute activité met à disposition des acteurs qui la mènent un certain nombre de fonctionnalités d'outils. Ces fonctionnalités ne sont disponibles qu'en fonction du contexte de la tâche menée par l'acteur considéré. Pour des activités coopératives d'apprentissage, on distingue trois types d'outils :
  1. *Des outils de production* qui sont utiles aux activités individuelles (pour la production des ressources requises par l'activité).
  2. *Des outils de coordination* qui offrent un point de vue dynamique sur les activités menées par les uns et les autres et sur la façon dont elles sont synchronisées.
  3. *Des outils de communication* qui permettent l'échange d'information entre apprenants ou entre apprenants et moniteurs.

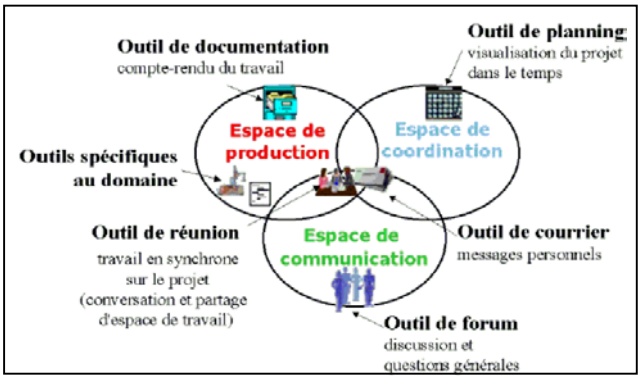


Figure 3 : Les trois espaces des outils d'un environnement collectif [George, 2001]

Dans la seconde partie de cet article, nous allons focaliser notre étude sur le problème de la mise en œuvre d'un environnement-support capable de gérer des activités coopératives d'apprentissage compatibles avec les objectifs énoncés ci-dessus. Un exemple d'utilisation illustrera nos propos.

2. Spécifications techniques d'une architecture support pour les activités coopératives d'apprentissage.

2.1. Existant en matière de normes et architectures-support

L'industrie du logiciel a investi depuis maintenant plusieurs années le domaine de l'E-Learning. Les principaux acteurs militent pour la définition de standards qui facilitent la compatibilité des développements d'applications éducatives. Disons clairement qu'il n'existe pas de standards définissant comment les développeurs doivent bâtir des cours en ligne de telle manière que des cours mis en œuvre par différentes compagnies puissent interagir. Il y a en revanche des groupes de travail définissant des spécifications, des protocoles qui sont proposés à l'industrie du logiciel éducatif : citons parmi les groupes non européens les travaux complémentaires ou concurrents menés par the IEEE's Learning Technology Standards Committee (IEEE LSTC) [LTSC, 2001], the IMS Global Learning Consortium [IMS, 2001], the Aviation Industry CBT Committee (AICC) [AICC, 2000], the Advanced Distributed Learning with its Sharable Content Object Reference Model (ADL-SCORM) [SCORM, 2000]. En Europe, les consortiums ARIADNE [ARIADNE, 2001] et PROMETEUS [PROMETEUS, 2001] sont les principaux acteurs.
Un certain nombre d'invariants ressortent de tous ces travaux :

- La prédominance des architectures client-serveur dans la mesure où le déploiement des contenus éducatifs se fait via le Web. Afin de dépasser les limites du langage HTML (trop tourné vers la présentation de contenus), les travaux se sont tournés vers le langage XML comme support de transfert des contenus entre le serveur Web et les clients (postes de travail des apprenants),
- La recherche d'une meilleure réutilisabilité des contenus éducatifs. L'idée générale est d'extraire des contenus éducatifs (Learning Contents) deux types d'éléments :
  1. Les éléments de présentation de ces contenus aux apprenants (aspects liés à l'interface usager),
  2. Les éléments de navigation qui définissent l'orientation proposée aux apprenants d'un contenu éducatif à un autre. Cette orientation dépend en général du comportement constaté de l'apprenant au niveau de l'un ou d'une liste de contenus éducatifs dans lesquels il a navigué.

Pour répondre à ces préoccupations, différents types de systèmes sont proposés [Chapman & Hall, 2001]:

- Les Learning Management Systems (ou LMS) assurent

le déploiement des contenus éducatifs aux apprenants. Ils gèrent l'accès des usagers aux différents modules de formation, permettent aux pédagogues de suivre l'avancement des apprenants dans les modules (temps écoulé, historique des réponses, …) et de gérer l'orientation de ces apprenants d'un module à un autre. De exemples de LMS sont WebCT, Ingenium, WBT Manager, …

- Les Learning Content Management Systems (ou LCMS) permettent à des experts d'un domaine, à des développeurs, de coopérer (via le Web) pour créer des contenus éducatifs (ou Learning Contents) aussi réutilisables que possible. ASPEN est un exemple d'environnement entrant dans cette catégorie [Rengarajan, 2001].

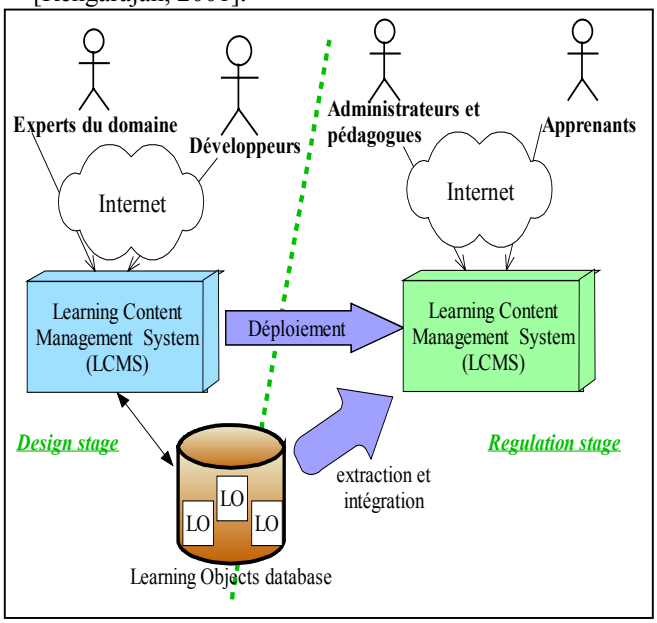


Figure 4 : Le rôle du LMS et du LCMS dans la diffusion de contenus éducatifs pour le Web

[Chapman & Hall, 2001] note qu'à ce jour, des travaux importants sont engagés et doivent être poursuivis afin d'améliorer les capacités de navigation entre contenus éducatifs offerts par ces systèmes : ces capacités se bornent à des orientations basiques d'un contenu à un autre, ce qui limite l'adaptabilité des modules de formation offerts aux comportements réels des apprenants.

Nous rajouterons que ces LMS nous semblent plus pensés pour transmettre des savoirs sous forme de pages et exercices proposés aux apprenants que pour mettre en œuvre des activités d'apprentissage (plus basées sur des savoir-faire) qu'elles soient individuelles ou coopératives. Ces constatations nous conduisent à proposer, sur la base des travaux menés par les grands consortiums du domaine, un certain nombre d'adaptations que nous décrivons dans le prochain paragraphe.

2.2. Eléments d'une solution technique pour les activités coopératives d'apprentissage

Notre objectif est ici de décrire le fonctionnement général d'un Learning Management System (LMS) capable de prendre en compte des activités coopératives telles que celles décrites dans la partie 1. Le rôle de ce LMS est donc de permettre le suivi des apprenants, tant dans leur progression et leurs performances dans les activités coopératives proposées que dans le niveau de tutorat qui leur est apporté.

Le point de départ de notre analyse est le suivant. L'architecture client-serveur qui est la base du Web et sur laquelle s'appuient tous les développements en Formation Ouverte et A Distance (FOAD) actuels (notamment tous ceux des consortiums cités dans le paragraphe précédent :

- ne peut être remise en cause sous peine que nos travaux se coupent de toutes les retombées positives en matière de normalisation qui découlent de l'activité de ces consortiums,
- pose cependant de nombreuses difficultés dans la mesure où elle place toute l'intelligence du système au niveau du serveur (machine sur laquelle se situe le LMS). La notion d'activité perd selon nous une partie de son sens dans ce type d'architecture puisque l'apprenant interagit avec des pages qui lui sont envoyées les unes après les autres par le LMS. La notion d'activité n'a donc pas d'existence sur le poste de l'apprenant, chaque action de l'apprenant devant être envoyée au LMS pour interprétation et rétroaction sous forme d'une nouvelle page Web (cf figure 5.1).

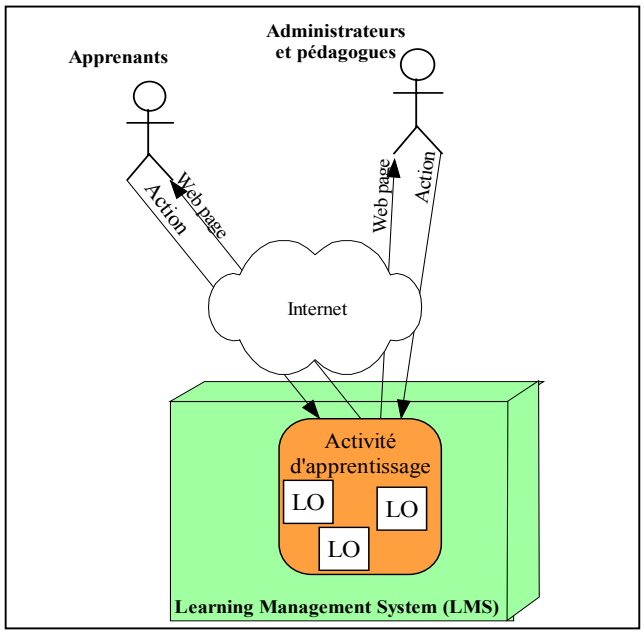


Figure 5.1 : Une activité d'Apprentissage conforme avec les modèles classiques de diffusion par le Web

Notre proposition (cf figure 5.2 ci dessous) *ne se contente pas de changer la granularité des ressources transmises* aux apprenants (Page Web / Activité d'Apprentissage).

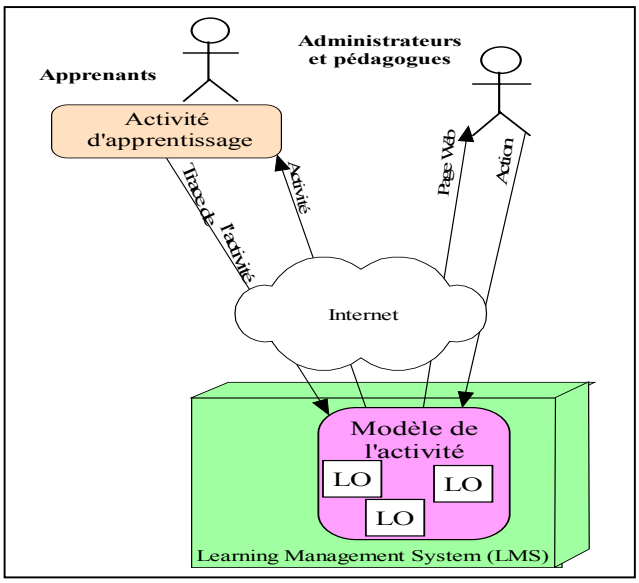


Figure 5.2 : Notre proposition pour l'exploitation via le Web d'Activités d'Apprentissage

Une page Web est une entité de présentation d'informations qui fournit des capacités d'intéraction limitées ; une activité d'apprentissage intègre des éléments de présentation et d'intéraction mais aussi des outils permettant à l'apprenant d'agir pour résoudre le problème posé par l'activité (notion de micro-monde qu'on retrouve dans la figure 3 : outils de production, de communication, de coordination).

Notre proposition propose aussi une distribution de la connaissance sur l'activité menée par les apprenants entre le serveur et le client. Cette distribution est basée sur les éléments suivants :

- Le poste serveur héberge un modèle de données lui permettant de disposer, en temps réel de toute la connaissance sur l'activité concrète menée par chaque apprenant (que nous appellerons « Domaine de l'activité ») ainsi que sur l'activité coopérative globale (que nous appellerons « Organisation de l'activité »). Les détails des informations relevant du domaine de l'activité et de l'organisation de l'activité seront donnés dans le paragraphe 2.3),
- Chaque poste client récupère du serveur toute l'information décrivant :
  1. L'activité à mener par l'apprenant présent sur ce poste client : cette information correspond à un sous-ensemble des informations du domaine et de l'organisation de l'activité.
  2. Mais aussi les interfaces-usagers (écrans, outils et fonctionnalités) mises à sa disposition.

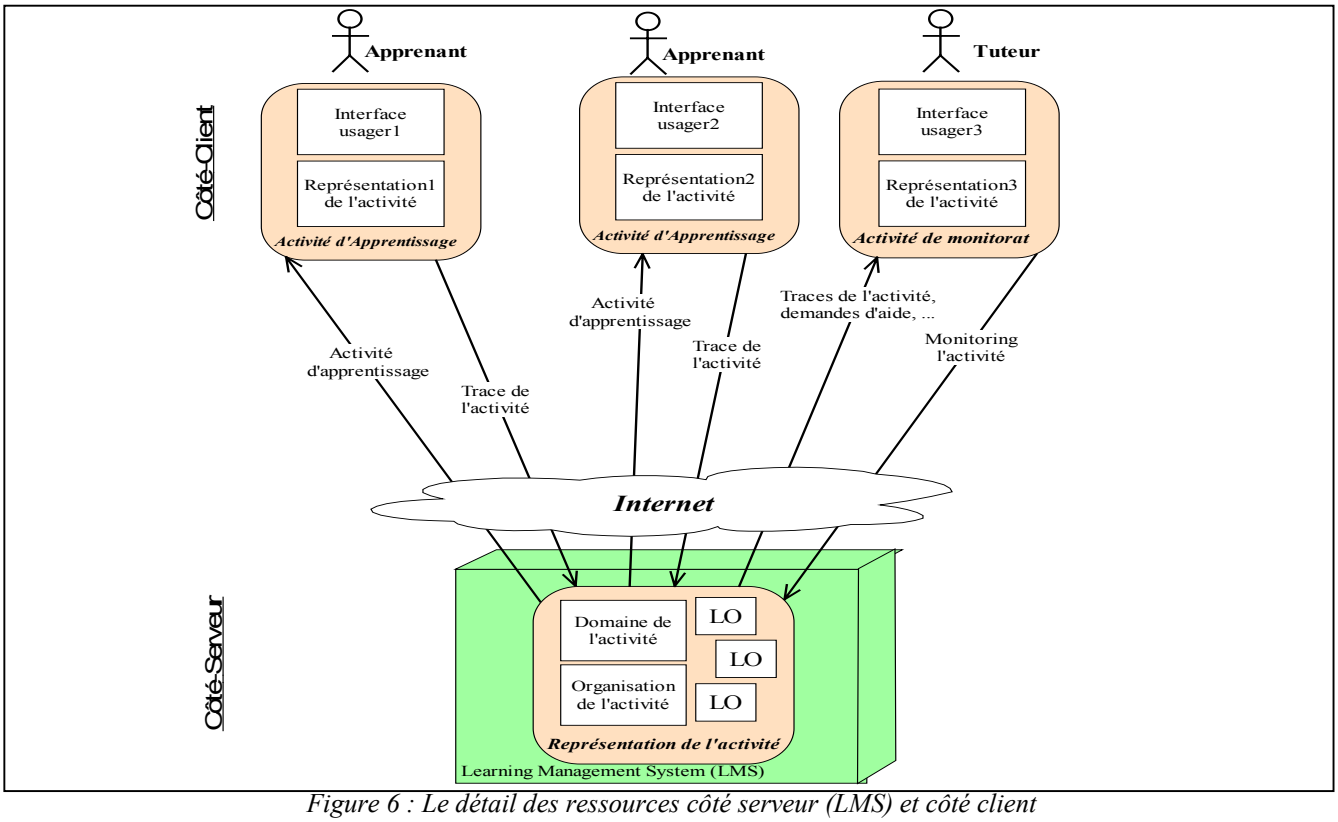


Figure 6 : Le détail des ressources côté serveur (LMS) et côté client

Chaque apprenant, au travers de ses interactions, met à jour certaines valeurs du domaine de l'activité ainsi que celles relatives à l'organisation de l'activité (ce qui permettra, non seulement de synchroniser les actions d'un apprenant avec les activités d'autres apprenants mais aussi de synchroniser les actions de tutorat avec les activités d'apprentissage menées).

L'enjeu est donc de mettre en œuvre un LMS capable de supporter des modalités de fonctionnement compatibles avec le schéma précédent. L'analyse du fonctionnement d'un tel système nous oriente vers un pattern de type Modèle-Vue-Présenteur (MVP) comme support de mise en œuvre. Ce modèle permet en effet d'établir un pont d'informations (bi-directionnel) entre un serveur et des clients Web. Le principe du pattern MVP vient du projet IBM-Taligent [Potel, 1996]. Il s'agit d'une extension du pattern Model-View-Controler (MVC) qui a fait la popularité de Smalltalk et qui est largement utilisé pour tous les développements d'Interfaces-Usagers (notamment Java-Swing mais aussi des projets du consortium Apache tels que STRUTS cf http://jakarta.apache.org/struts/) :

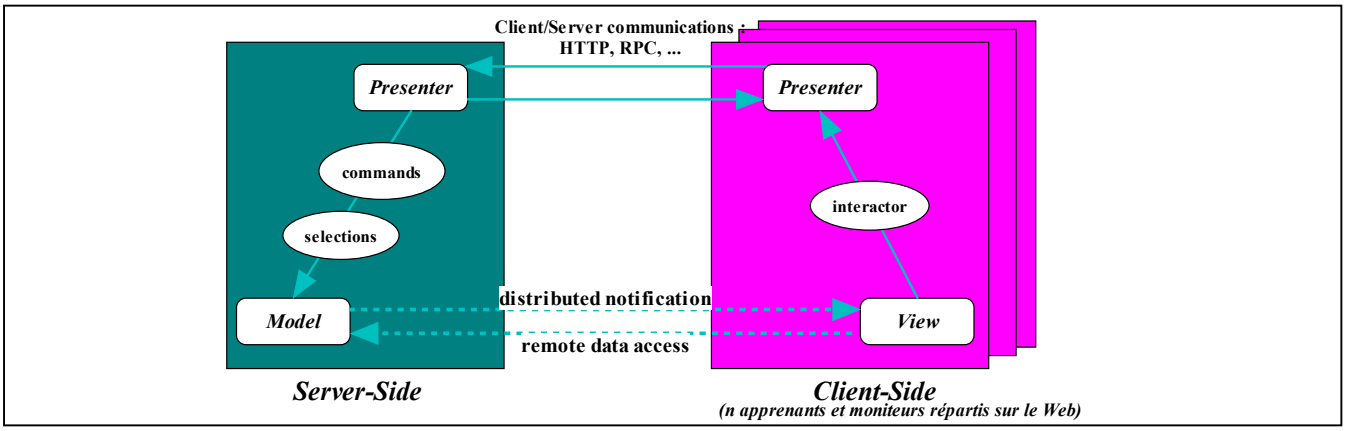


Figure 7 : Le pattern Model-View-Presenter dans un contexte réparti

Côté serveur, ce pattern permet de définir des vues partielles sur un modèle (structure de données et méthodes) ; ces vues étant obtenues grâce à des restrictions sur certaines valeurs stockées dans le modèle ou des projections sur certaines propriétés du modèle.

Côté client, ce pattern permet de lier l'interface-usager offerte à un acteur (représentée dans le schéma par la classe View) au modèle sur lequel agit cette interface (représenté par la classe Model) via le présenteur du modèle. Le pattern fonctionne comme si le modèle et la vue étaient en liaison directe et assure ainsi leur synchronisation.

Dans notre contexte, ce pattern va permettre de gérer la distribution des responsabilités entre les apprenants ou les moniteurs (côté client) et le LMS (côté serveur). L'idée est de placer :

- Au niveau de la classe Model les objets représentant le domaine de l'activité et l'organisation de l'activité,
- Au niveau de la classe View les objets permettant d'offrir une interface-usager susceptible de permettre à l'apprenant de suivre l'activité coopérative proposée par le LMS,
- La classe Presenter a pour rôle de synchroniser ce qui touche à la représentation de l'activité et ce qu'en perçoit un utilisateur (apprenant ou moniteur).

Remarquons ici, que plusieurs clients ayant leurs propres présenteurs (issus de la classe Presenter) et leurs propres vues (issues de la classe View) peuvent synchroniser leurs activités grâce à ce modèle ; le pattern MVP se charge d'assurer cette synchronisation pour la communauté des utilisateurs manipulant la classe Model (nous expliquerons comment dans le paragraphe suivant).

Dans le prochain paragraphe, nous allons présenter un exemple simplifié d'application éducative coopérative qui nous permettra de décrire comment un tel pattern MVP pourrait être mis en œuvre par un LMS.

2.3. Un exemple de mise en œuvre

Un pédagogue a conçu une situation dont l'objectif est que deux groupes d'apprenants coopèrent pour comprendre les effets de la radioactivité sur l'organisme. Il s'agit d'une situation simplifiée qui permettra à des enfants d'école primaire de percevoir les effets sur l'organisme en fonction de la quantité de matière radioactive, de la durée de contact avec la source radioactive, de la distance entre une personne et cette source radioactive, du calfeutrage de la source radioactive (capacité à isoler la source de son environnement immédiat). L'objectif est que les apprenants se meuvent dans un espace à proximité de la bombe, fassent des hypothèses et découvrent peu à peu les caractéristiques définissant le degré de contamination engendré par cette charge. Les enfants doivent échanger des informations sur

leurs découvertes et être en mesure à la fin de remplir un formulaire permettant de valider leurs acquis.

Pour décrire cette activité d'apprentissage, il nous faut décrire les éléments du système d'informations qui vont se répartir au niveau serveur et au niveau client (cf figure 7). Les informations doivent être stockées au niveau de la classe Model (côté serveur) afin d'être partagées et mise à disposition des différents acteurs. Par rapport au pattern MVP, la partie Model de notre exemple se décrit de la manière suivante :

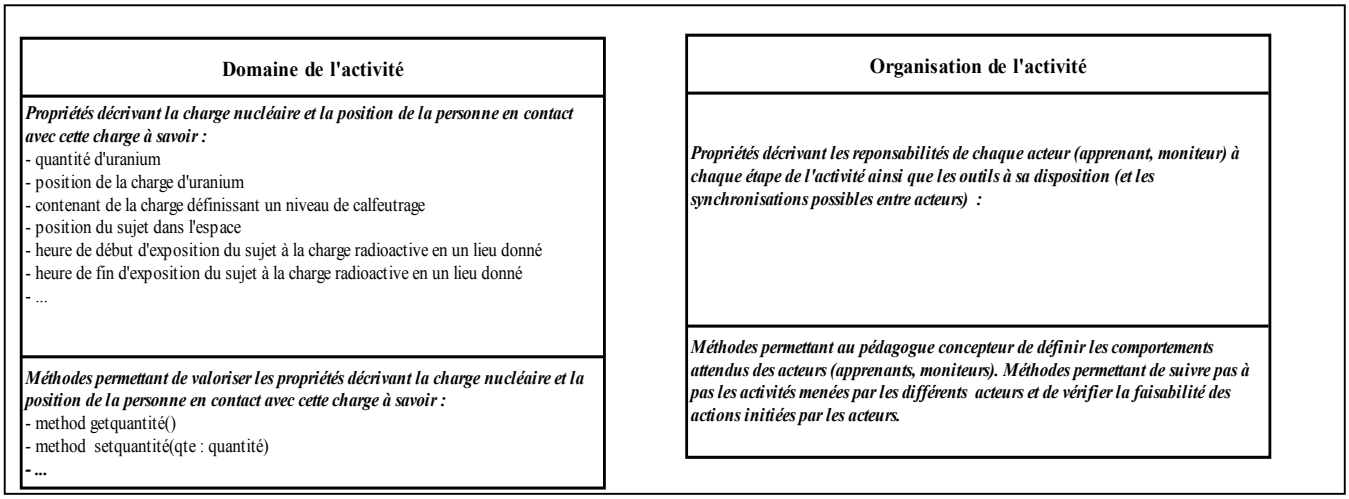


Figure 8 : Description générale des deux classes permettant de modéliser une activité

La description en UML du détail des classes utiles pour mettre en œuvre ces deux éléments du modèle dépassent le cadre de cet article. Nous donnons ci-dessous le modèle complet en UML pour la partie du modèle intitulée : Organisation de l'activité. Cet ensemble de classes UML permet de décrire à la fois les activités prévues pour chaque acteur et de suivre les actions réellement menées durant l'activité par chaque acteur. Le lecteur trouvera les explications détaillées relatives à ce modèle dans [Sallaberry, *et al.*, 2002] :

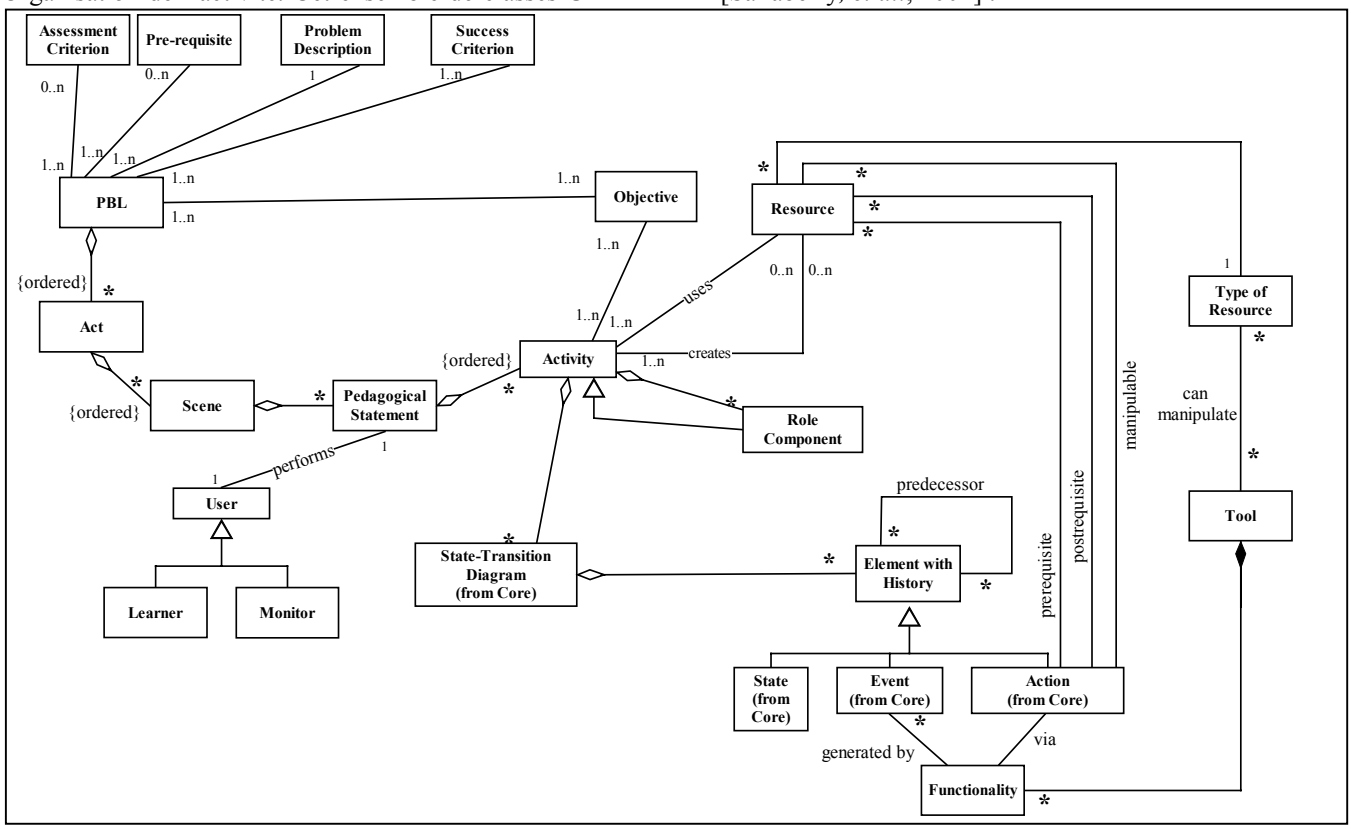


Figure 9 : Le détail des classes décrivant l'organisation d'une activité coopérative d'apprentissage

Côté View du pattern MVP, les apprenants (côté client) vont percevoir le modèle de la figure 8 au travers de l'activité qu'ils mènent :

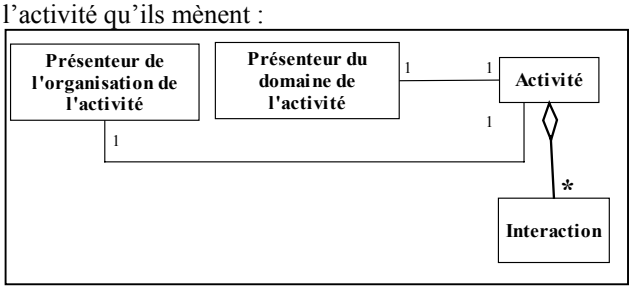


Figure 10 : La perception de l'activité par les apprenants (côté client)

L'activité de l'usager est contrainte par l'organisation de l'activité qui lui est proposée et par les éléments du domaine qu'on lui permet de manipuler (il s'agit bien de présenteurs car deux apprenants différents vont avoir sur la même activité coopérative des droits et des vues différentes). Cette activité de l'usager va se manifester par des interactions de celui-ci avec un environnement informatique (au travers d'une certaine interface-usager qui lui offrira certaines fonctionnalités en liaison avec son rôle dans l'activité). Ainsi, une activité va par exemple offrir à un apprenant un environnement d'outils lui permettant d'agir dans un micro-monde dédié. Le schéma ci-dessous présente des exemples d'interactions en relation avec le thème « Danger radioactif » :



Figure 11 : Exemple d'interactions présentées à un apprenant

L'écran présenté ci-dessus est un exemple d'interaction dans lequel l'apprenant se retrouve dans une partie de la ville (ici la zone industrielle) et perçoit le phénomène radioactif au travers d'un certain nombre de capteurs : le niveau de radioactivité actuelle, l'état physique de l'enquêteur se trouvant dans la zone industrielle, … On trouve aussi dans cette interaction :

- Des objets que peut analyser l'enquêteur pour déterminer si la charge radioactive se situe ou pas dans cette zone (taux de radioactivité instantanée, état physique de l'enquêteur).
- Des boutons permettant à l'enquêteur d'interagir avec d'autres enquêteurs (pour demander des renseignements sur les autres zones de la ville, pour renseigner d'autres enquêteurs, pour visualiser les lieux de la ville où se trouvent les différents enquêteurs).

D'autres interactions pour cet apprenant sont rendues disponibles par l'activité : une interaction lui permettant d'évaluer sa compréhension du phénomène radioactif observé est ainsi accessible à travers le bouton «Tu as trouvé ?».

Les autres enquêteurs disposent d'interfaces et de fonctionnalités d'outils spécifiques pour coopérer avec cet enquêteur selon un modèle de coopération défini au niveau de l'organisation de l'activité et sur la base d'informations partagées qui sont issues du domaine de l'activité (cf figure 8).

Les présenteurs (côté client et côté serveur) ne sont pas perçus par les acteurs ; ils permettent de gérer les synchronisations entre les actions des usagers et les valeurs prises par les propriétés du modèle. Cette synchronisation est basée sur le pattern Sujet-Observateur [Gamma, *et al.*, 1995]. Le pattern Sujet-Observateur assure le fonctionnement général du LMS puisque c'est lui qui assure la synchronisation entre les valeurs des propriétés gérées par les différentes classes d'objets (cf figure 12). Ce pattern fonctionne selon le principe suivant :

- Un objet qui souhaite observer des éléments d'un sujet donné va s'attacher à ce sujet (méthode publique attache() de la classe Sujet)
- A un instant donné (lorsqu'une valeur des propriétés d'un sujet est modifiée par un acteur externe ou par le sujet lui-même), le sujet invoque sa méthode privée notifie() qui va se charger de parcourir tous ses observateurs attachés pour les mettre au courant que le sujet a été changé. Chaque activité de l'apprenant, instance d'un observateur, se charge alors de demander la mise à jour des interactions fournies à l'apprenant ; chaque interaction connaît les propriétés du sujet sur lesquelles elle est bâtie. Un rafraîchissement des valeurs de ces propriétés de l'interaction suffit pour assurer la cohérence de cette interaction au regard du sujet observé.

Ainsi, côté client le schéma complet devient :

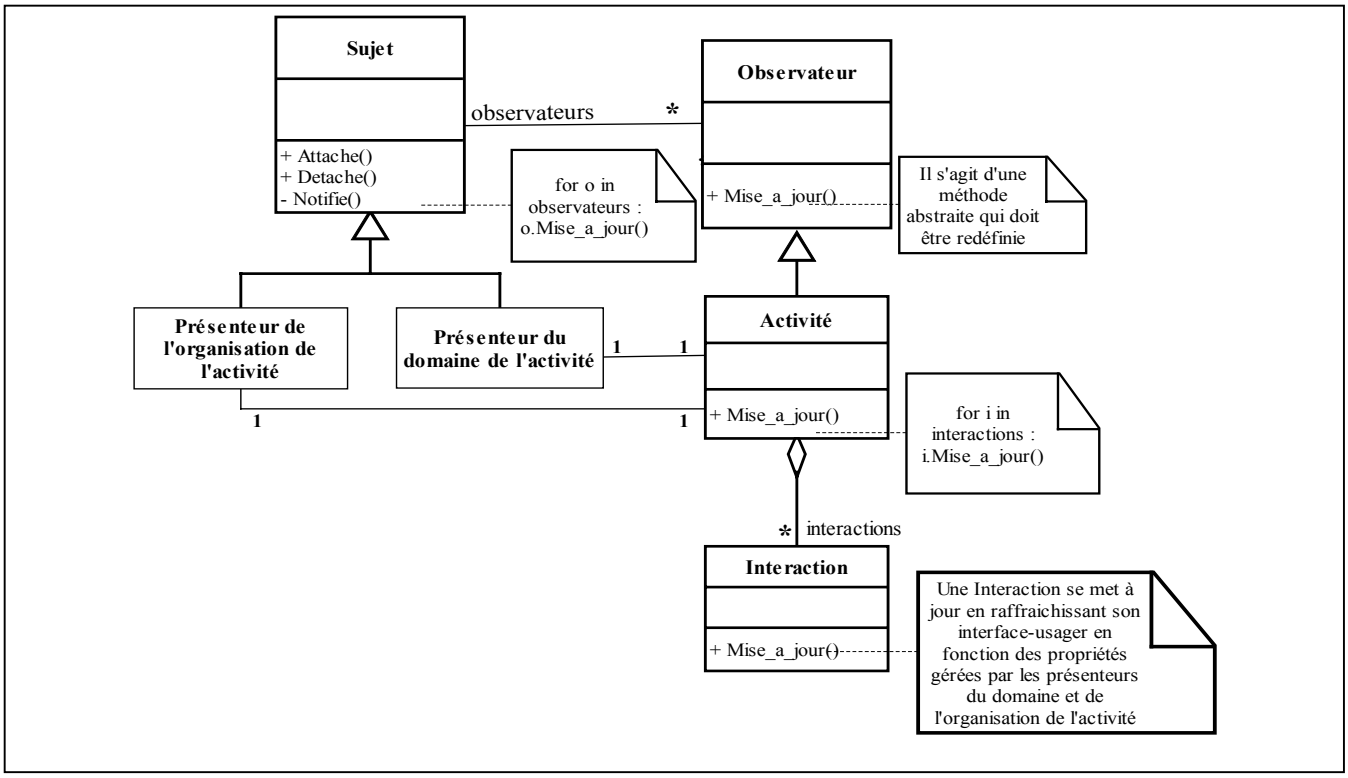


Figure 12 : Application du pattern Sujet-Observateur à la gestion d'une activité distribuée

Côté client (voir schéma ci-dessus), on peut voir comment le système est capable, dès qu'il y a modification d'une information au niveau d'un des présenteurs, de notifier l'activité en cours d'un apprenant pour que celle-ci prenne en compte le contexte nouveau (on rappelle que les activités étant coopératives, une valeur peut être mise à jour par un acteur autre que celui qui agit au niveau d'une interface-usager).

Nous ne présenterons pas les détails du fonctionnement côté serveur dans cet article. Cependant le fonctionnement est similaire côté serveur avec des présenteurs qui jouent le rôle d'Observateurs de sujets que sont le Domaine de l'activité et l'Organisation de l'activité.

3. Conclusions et perspectives

Dans cet article, nous avons montré qu'une pédagogie à base d'activités coopératives est compatible avec les travaux de normalisation menés par les consortiums internationaux centrés sur l'E-Learning. Après avoir décrit les enjeux, les acteurs et les principes de mise en œuvre d'activités d'apprentissage coopératives, nous avons focalisé notre attention sur la définition d'un système (appelé Learning Management System) capable d'assurer la distribution d'activités au service d'une communauté d'utilisateurs (apprenants et moniteurs).
En nous appuyant sur la notion de pattern et sur le langage UML, nous avons proposé une architecture technique permettant de supporter ces activités coopératives d'apprentissage. Au delà de l'exemple présenté dans cet article :

- Cette architecture doit être testée sur des situations d'apprentissage en vraie grandeur. C'est à ce travail que notre groupe de recherche vient de s'engager puisque nous avons acquis les ressources relatives à une situation d'apprentissage réelle [ACTIS, 2001] que nous tentons en ce moment de formaliser et de mettre en œuvre.
- Du point de vue technique, nous avons entamé un travail de développement afin de coder les fonctionnalités d'un LMS simplifié capable de se comporter selon les principes décrits dans cet article. Ce développement s'effectue dans un environnement EJB (Entreprise Java Beans), qui est particulièrement indiqué pour tous les développements Web côté serveur [Brethes, *et al.*, 2000].

Ces deux directions de travail nous permettront de valider les premiers résultats décrits dans cet article.

4. Bibliographie

ACTIS (2001). Smash Scenario. ACTIS Ltd, http://www.projectboxes.co.uk/catalogue/scenarios.html#smash

AICC (2000). Aviation Industry CBT Committee (AICC).

document #CMI001: CMI Guidelines for Interoperability, v3.4; released, www.aicc.org

ARIADNE (2001). Alliance of remote instructional authoring and distribution networks for Europe website [On-line. 2001: ARIADNE, http://ariadne.unil.ch/

Bourguin, Grégory (2000) Un support informatique à l'activité coopérative fondée sur la théorie de l'activité : le projet DARE, Thèse de Doctorat de l'Université de Lille

Brethes, T., Hisquin, F., & Pezziardi, P. (2000). Serveur d'Applications.

Chapman, B., & Hall, B. (2001). Learning Content Management Systems.

De La Passardière, B., & Giroire, H. (2001). XML au service des applications pédagogiques. *Revue Sciences et Techniques Educatives*, 99-112,

Develay, M. (1993). De l'apprentissage à l'enseignement. *Collection Pédagogies*

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns: elements of reusable object-oriented software.

George, S. (2001). Apprentissage collectif à distance. SPLASH : un environnement informatique support d'une pédagogie de projet. Thèse de doctorat de l'Université du Maine

Harasim, L. (1999). A framework for Online Learning: The Virtual-U. *IEEE Computer, 0018-9162/99*, http://www.telelearn.ca/g_access/news/r9044.pdf.

IMS (2001). Instructional management systems project website. http://imsproject.org/

Koper, R. (2001). Modeling Units of Study from a Pedagogical Perspective : the pedagogical meta-model behind EML. Educational Expertise Technology Centre, Open University of the Netherlands

Laforcade, P. (2002). Modélisation des composants éducatifs mis en jeu dans une plate-forme de FOAD. *soumis à la conférence Inforsid 2002*. Grenoble (France)

LOM (2000). LOM working draft v4.1. http://ltsc.ieee.org/doc/wg12/LOMv4.1.htm

LTSC (2001). Learning technology standards committee website. http://ltsc.ieee.org/

Marquesuzaà, C. (1998). OMAGE : Outils et Méthode pour la spécification des connaissances au sein d'un Atelier de Génie Educatif. Doctorat de l'Universite de Pau et des Pays de l'Adour

Meirieu, P. (1994). Apprendre... Oui mais comment? *Collection Pédagogies*

Nodenot, T. (2001). A case for an Adult Educational Technology. *IFIP 7th World Conference on Computers in Education*. Copenhagen (Denmark)

Potel, M. (1996). MVP : Model-View-Presenter : the Taligent Programming Model for C++ and Java. Taligent Incorporated, www.carfield.com.hk/document/software_design/MVC.pdf

PROMETEUS (2001). European Partnership for a Common Approach to the Production of e-learning Technologies and Content website. http://www.prometeus.org

Rengarajan, R. (2001). ASPEN LCMS : Click2Learn's Comprehensive LCMS Solution. Click2learn Inc., http://home.click2learn.com/en/downloads/Aspen_LCMS_white_paper.pdf

Rodriguez-Artacho, M., & all, a. (1999). Using a high-level language to describe and create Web-based learning scenarios. *29th ASEE/IEEE Frontiers in education Conference*. San Juan, Porto Rico

Sallaberry, C., Nodenot, T., Marquesuzaà, C., Bessagnet, M.-N., & Laforcade, P. (July 14th-18th 2002). An attempt to design an Information System supporting Collaborative Problem-Based Learning Situations. *The 6th World MultiConference on Systemics, Sybernetics and Informatics*. Orlando (USA), http://www.iiis.org/sci2002/

Sawyer, B. (2000). Using Virtual U in Consultant and Training Situations:

Tips and advice on creating 1-2 day workshops or seminars using Virtual U., http://www.virtual-u.org/edutools.html

SCORM (2000). The Department of Defense Advanced Distributed Learning -- ADL -- Initiative Releases Version 1.2 of the Sharable Courseware Object Reference Model -- SCORM 1.2.

Suthers, D. (August 2001). Architectures for Computer Supported Collaborative Learning. *IEEE International Conference on Advanced Learning Technologies (ICALT'2001)*. Madison, Wisconcin (USA)

Wiley, D. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. http://reusability.org/read/chapters/wiley.doc