

Titre: Architecture et modèle collaboratif multi-agent pour une plate-forme de laboratoires virtuels
Title: Architecture and collaborative multi-agent model for a virtual laboratories platform

Auteur: Ilior Abari
Author:

Date: 2002

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Abari, I. (2002). Architecture et modèle collaboratif multi-agent pour une plate-forme de laboratoires virtuels [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/7019/>
Citation:

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7019/>
PolyPublie URL:

Directeurs de recherche: Samuel Pierre, & Hamadou Saliah-Hassane
Advisors:

Programme: Génie électrique
Program:

UNIVERSITÉ DE MONTRÉAL

ARCHITECTURE ET MODÈLE COLLABORATIF MULTI-AGENT POUR UNE
PLATE-FORME DE LABORATOIRES VIRTUELS

ILIOR ABARI

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

DÉCEMBRE 2002



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-81531-5

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ARCHITECTURE ET MODÈLE COLLABORATIF MULTI-AGENT POUR UNE
PLATE-FORME DE LABORATOIRES VIRTUELS

Présenté par : ABARI Ilior

En vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

A été dûment accepté par le jury constitué de :

Mme BOUCHENEH Hanifa, Ph.D., présidente

M. PIERRE Samuel, Ph. D., membre et directeur de recherche

M. SALIAH-Hassane Hamadou, Ph. D., membre et codirecteur

M. VAZQUEZ-ABAD Jésus, Ph. D., membre

à ma famille ...

REMERCIEMENTS

Je remercie sincèrement mon Directeur de recherche M. Samuel Pierre, pour m'avoir conseillé, encouragé puis soutenu financièrement et moralement pendant mes recherches. Merci aussi à mon co-directeur M. Hamadou Saliah-Hassane, pour sa disponibilité et son soutien appréciable. Ensemble, ils ont su orienter mes travaux.

Grand merci également à mon épouse et à mon fils pour leur patience et leur compréhension. Sans cela, ces études de maîtrise n'aurait jamais été possibles.

Enfin, j'exprime toute ma reconnaissance et ma gratitude à mes parents qui m'ont toujours soutenu et encouragé depuis ma tendre enfance.

Je ne saurais terminer, sans savoir gré à tous mes amis, qu'ils soient membres du laboratoire de recherche en réseautique et informatique mobile (LARIM) de l'École Polytechnique de Montréal ou pas, de m'avoir soutenu et encouragé.

RÉSUMÉ

Après avoir marqué notre manière de communiquer et de faire des affaires, l'Internet influence notre façon d'apprendre. Au même moment, l'enseignement à distance a évolué du schéma classique basé sur l'envoi de contenus pédagogiques par voie postale vers l'apprentissage électronique (e-learning) fondé sur les réseaux de télécommunications.

En sciences et génie, tout enseignement théorique est généralement complété par une pratique en laboratoire. De ce fait, pour les environnements de télé-apprentissage destinés à ces domaines, les laboratoires virtuels sont une composante essentielle.

Dans ce mémoire, nous nous sommes intéressés à une plate-forme de laboratoires virtuels répartis offrant des équipements partagés à des groupes d'apprenants qui effectuent des travaux pratiques. Ainsi, chaque groupe est une communauté d'apprentissage qui repose sur les interactions entre ses différents membres afin de réussir le travail pratique qui lui a été assigné. À l'intérieur d'un groupe, les coéquipiers collaborent. Mais, lorsqu'ils appartiennent à des groupes différents, les apprenants coopèrent et s'entraident, à l'image de ce qui se fait dans un laboratoire conventionnel.

Après avoir proposé une architecture générale de la plate-forme, nous étudions en détail deux de ses composantes: un modèle collaboratif multi-agent et un répertoire de cahiers électroniques. Le modèle aborde le problème de la gestion d'une classe d'apprenants géographiquement répartis et scindés en groupes de travaux pratiques. Le répertoire est basé sur les objets d'apprentissage (learning objects) et il s'appuie sur les standards de métadonnées LOM (Learning Object Metadata) et LIP (Learner Information Package).

Lorsqu'on fait des expériences de laboratoire dans un environnement réparti, l'un des défis majeurs consiste à savoir comment collecter, communiquer, partager et gérer des données hétérogènes (résultats intermédiaires, analyses, images, observations, etc.) manipulées concomitamment par des participants

géographiquement dispersés mais œuvrant en équipes virtuelles. Le cahier électronique est une réponse à cette préoccupation.

Dans le répertoire que nous proposons, chaque groupe utilise un cahier électronique par séance de travail pratique. Ces cahiers sont classés par disciplines, elles-mêmes subdivisées en domaines d'enseignement tenant compte du cursus de formation. L'implémentation a été faite sous la forme d'une base de données d'objets d'apprentissage (learning objects). La raison de ce choix est bien simple car, de par leur granularité et les standards de méta-données qui les portent, les objets d'apprentissage favorisent la réutilisation, l'interopérabilité et l'échange de contenus éducatifs. L'analyse des résultats que nous avons obtenus, par des tests effectués au laboratoire de recherche LARIM, montre que les temps de réponse sont satisfaisants. Les délais les plus longs sont observés lorsqu'il y a des modifications simultanées et concurrentes d'un paragraphe du cahier. Or, la probabilité de les rencontrer est d'autant plus faible que les paragraphes seront plus granulaires, ce qui correspond parfaitement à la philosophie des objets d'apprentissage.

Outre le temps de réponse, la sécurisation des accès au répertoire par un système à clés asymétriques (par exemple) et la flexibilité de choisir le standard de méta-données qu'on désire (LOM, SCORM ou autre) par l'implémentation de tables de transcodage entre standards, constituent des directions de recherches futures.

ABSTRACT

After changing our way to communicate, Internet influences how we learn things. At the same time, traditional distance learning has moved from sending learning content by postal service to electronic learning (or e-learning). This relies on telecommunication networks.

In all scientific and engineering fields, theoretical lectures are generally complemented by practical laboratory sessions. Hence, in telelearning environments, virtual laboratory is very important to address these domains.

In this thesis, we study questions related to a distributed virtual laboratory platform. This platform offers equipment shared by learner groups during a session. Each group member interacts with his teammates to achieve the given laboratory assignment. Within a group, members coordinate their actions and collaborate when working. But when composing separate groups, learners cooperate and help classmates, like in a conventional laboratory session.

After designing a global architecture of the platform, we study in detail two of its components: a collaborative multi-agent model and an electronic laboratory notebook repository. The model tackles how to manage a classroom subdivided into geographically distributed groups. In the same laboratory session, groups may work on different assignments. The repository is based on learning objects using LOM (Learning Object Metadata) and LIP (Learner Information Package).

During virtual laboratory experiment sessions, a major problem is how to collect, store, manage and share heterogeneous data (intermediate results, analysis, annotations, ...) manipulated simultaneously by geographically distributed attendees composing each team. The electronic notebook is aimed to deal with.

In our repository, each learner group uses a different electronic notebook for each laboratory session. According to the curricula, these notebooks are classified by teaching domains and subdomains. We have implemented the repository as a learning object database. This solution is good, because learning objects facilitate reuse, interoperability

and learning content exchange among learning management systems due to their granularity and the metadata standards on which they are based. The results we have obtained after tests conducted in our research laboratory (LARIM), show that the response time is accurate. The slowest case occurs when we have simultaneous and concurrent modifications on the same notebook paragraph. However, the probability to get this case is low when the paragraph granularity level is high, and vice versa. This fits properly with the learning object concept philosophy.

Besides response time, secured repository access by mean of asymmetric keys cryptographic system and flexibility in choosing the desired meta-data standard (LOM, SCORM or other) through transcoding tables represent further researches directions.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX.....	xii
LISTE DES FIGURES.....	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xv
LISTE DES ANNEXES.....	xvi
CHAPITRE I : INTRODUCTION	1
1.1 Définitions et concepts de base	1
1.2 Éléments de la problématique	3
1.3 Objectifs de la recherche	5
1.4 Plan du mémoire	6
CHAPITRE II : LABORATOIRES VIRTUELS, COLLABORATION ET STANDARDISATION DES MÉTA-DONNÉES	7
2.1 Évolution de l'informatique dans l'éducation	7
2.2 Modèle d'exploitation des laboratoires virtuels	10
2.3 Typologie des laboratoires virtuels	11
2.3.1 Les simulations	11
2.3.2 Les télelaboratoires	15
2.4 Environnements de développement des laboratoires virtuels	17
2.4.1 L'environnement graphique VLab	17
2.4.2 L'environnement OPEA-MAT	18
2.4.3 L'environnement LabVIEW	19
2.4.4 Limites de ces environnements	23
2.5 L'espace collaboratif dans d'autres domaines informatiques	24
2.5.1 En intelligence artificielle distribuée	24

2.5.2 Dans les logiciels de travail en équipe	26
2.5.3 Dans les grappes d'ordinateurs	27
2.6 La standardisation des méta-données.....	29
2.6.1 Le Learning Object Metadata (LOM)	32
2.6.2 Le Learner Information Package (LIP)	34
CHAPITRE III : ARCHITECTURE ET MODÈLE COLLABORATIF.....	36
3.1 Architecture générale de la plate-forme	36
3.1.1 Banque de laboratoires virtuels spécifiques et interface d'accès	38
3.1.2 Mécanismes d'accès aux ressources génériques	38
3.1.3 L'intégrateur des profils et préférences	40
3.1.4 Logique de contrôle et pilotage de la plate-forme (LCP)	44
3.1.5 Outils logiciels de soutien.....	44
3.1.6 Le gestionnaire de la collaboration	45
3.1.7 Le gestionnaire des équipements génériques	46
3.1.8 Le répertoire de cahiers électroniques de laboratoire	48
3.1.9 Autres composantes de l'architecture	52
3.2 Modèle collaboratif multi-agent.....	53
3.2.1 Contexte d'utilisation, hypothèses et idées de base	54
3.2.2 Description des éléments du modèle collaboratif	57
3.2.3 Initialisation et arrêt	64
CHAPITRE IV : IMPLÉMENTATION, MISE EN ŒUVRE ET RÉSULTATS	66
4.1 Justification du choix du répertoire de cahiers électroniques	66
4.2 Choix d'outils pour l'implémentation.....	67
4.3 Détails de l'analyse et de la conception	69
4.3.1 Diagrammes des cas d'utilisation	69
4.3.2 Diagrammes des classes.....	70
4.3.3 Diagrammes d'activités.....	76
4.4 Mise en œuvre	80
4.4.1 Connexion au répertoire.....	80

4.4.2 L'éditeur de paragraphe	81
4.5 Expérimentation et analyse des résultats.....	86
4.5.1 Critère expérimental.....	87
4.5.2 Hypothèses et scénarios de test	87
4.5.3 Résultats et analyses.....	89
CHAPITRE V : CONCLUSION	96
5.1 Synthèse des travaux	96
5.2 Limitations des travaux	98
5.3 Recherches futures	99
BIBLIOGRAPHIE	101
SITES INTERNET	105

LISTE DES TABLEAUX

Tableau 2.1 Évolution de l'informatique dans l'éducation	9
Tableau 2.2 Catégories du LOM	34
Tableau 2.3 Catégories du LIP	35
Tableau 3.1 Le niveau local du modèle collaboratif	63
Tableau 3.2 Le niveau central du modèle collaboratif	63
Tableau 4.1 Résultats du test 1	90
Tableau 4.2 Résultats du test 2	91
Tableau 4.3 Résultats du test 3	93
Tableau 4.4 Rapprochement des résultats des trois tests	94

LISTE DES FIGURES

Figure 1.1 Modèle conceptuel de la plate-forme de laboratoires virtuels	4
Figure 2.1 Modèle d'exploitation des laboratoires virtuels	10
Figure 2.2 Fonctionnement d'un laboratoire virtuel basé sur la simulation	11
Figure 2.3 Les deux ondes avant l'interférence	12
Figure 2.4 Les trois ondes après l'interférence	13
Figure 2.5 Tir de canon en cinématique	15
Figure 2.6 Architecture simplifiée d'un télelaboratoire en mode client / serveur	16
Figure 2.7 Panneau avant d'un instrument virtuel	21
Figure 2.8 Diagramme d'un instrument virtuel	21
Figure 2.9 Évolution de la standardisation des méta-données	32
Figure 3.1 Architecture générale de la plate-forme	38
Figure 3.2 L'intégrateur des profils et préférences	41
Figure 3.3 Adaptation multi-terminale avec XML/XSL	43
Figure 3.4 Structure arborescente du répertoire de cahiers de laboratoire	50
Figure 3.5 Face avant d'un instrument virtuel LabVIEW	55
Figure 3.6 Modèle collaboratif multiagent	58
Figure 3.7 Interactions entre les différents agents	62
Figure 4.1 Interactions de l'apprenant avec le système	71
Figure 4.2 Interactions de l'encadreur avec le système	72
Figure 4.3 Diagramme des paquetages du système	74
Figure 4.4 Diagramme des classes du répertoire	75
Figure 4.5 Diagramme des classes LIP	77
Figure 4.6 Diagramme des classes du LOM	78
Figure 4.7 Diagramme d'activités pour les interactions Apprenant / Système	79
Figure 4.8 Connexion au répertoire	80
Figure 4.9 L'éditeur des paragraphes	82
Figure 4.10 Méta-données associées à un paragraphe	83

Figure 4.11 Écran principal du serveur	84
Figure 4.12 Crédit d'un apprenant	85
Figure 4.13 Méta-données sur les contacts d'un acteur	86
Figure 4.14 Graphe de scénarios	89
Figure 4.15 Temps de réponse du test numéro 1	91
Figure 4.16 Temps de réponse du test numéro 2	92
Figure 4.17 Temps de réponse du test numéro 3	94
Figure 4.18 Synthèse des résultats	95

LISTE DES SIGLES ET ABRÉVIATIONS

API	Application Programming Interface
ARIADNE	Alliance of Remote Instructional Authoring and Distribution Networks for Europe
CENSA	Collaborative Electronic Notebook Systems Association
DCMI	Dublin Core Metadata Initiative
EAO	Enseignement Assisté par Ordinateur
EIAO	Enseignement Intelligemment Assisté par Ordinateur
HTML	Hyper Text Mark-up Language
IEEE	Institute of Electrical and Electronics Engineers
IMS	Instructional Management Systems
ISO	International Standards Organization
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LIP	Learner Information Package
LOM	Learning Object Metadata
PDA	Personal Digital Assistant
SCORM	Sharable Content Object Reference Model
STI	Système Tutoriel Intelligent
UML	Unified Modeling Language
XMI	XML Metadata Interchange
XML	Extensible Mark-up Language
XSL	Extensible Stylesheet Language
WML	Wireless Mark-up Language

LISTE DES ANNEXES

Annexe A	107
Annexe B	109

CHAPITRE I

INTRODUCTION

Les télécommunications et les réseaux ont ouvert plus que par le passé les applications de la science au grand public. Au cours des dernières années, d'intenses travaux de recherche ont été consacrés au rôle de l'informatique dans l'éducation. De cet engouement est né un nouveau style d'enseignement à distance appelé apprentissage électronique (*e-learning*). Celui-ci fait l'objet du présent mémoire. Cette évolution marquante qui vise à vaincre la distance et l'isolement pour promouvoir le travail collaboratif s'est opérée dans un contexte où l'efficacité est le souci majeur de chacun. En outre, elle a été largement confortée par le développement spectaculaire de l'Internet et l'émergence de technologies permettant la création, la gestion et la distribution de contenus dynamiques via des pages Web.

Dans ce chapitre d'introduction, nous présentons : les définitions et concepts de base, les éléments de la problématique, les objectifs de la recherche et le plan du mémoire.

1.1 Définitions et concepts de base

Le terme *télé-apprentissage* désigne un mode d'apprentissage basé sur l'utilisation des nouvelles technologies pour apprendre par les réseaux informatiques. Les dispositifs qui permettent cette formation dite en ligne favorisent l'interactivité et la personnalisation des interfaces utilisateurs. Aussi, le processus d'apprentissage est indépendant de l'heure et du lieu (Site #2). Les termes anglais *e-learning* et *virtual learning* en sont équivalents.

Il n'existe pas de définition normalisée pour les vocables *laboratoire virtuel* et *télélaboratoire*. Le premier fait référence à une expérience de laboratoire réalisée par l'entremise d'un environnement logiciel approprié, basé sur des nouvelles technologies telles que l'Internet et les réalités virtuelles. On parle de télélaboratoire (*remote laboratory*) lorsqu'il y a interaction avec un équipement réel distant. Sans équipement

réel, il peut s'agir de simulation. Ainsi donc, laboratoire virtuel signifie télélaboratoire ou simulation.

Dans un contexte de laboratoire virtuel, la *collaboration* dénote la possibilité offerte aux participants à une séance de laboratoire virtuel d'interagir, de discuter et d'échanger leurs points de vue sur la réalisation des tâches, en utilisant des moyens tels que le courrier électronique, le chat ou la vidéoconférence.

Au sein d'un groupe de travaux pratiques (TP), il y a unicité d'objectif et absence de divergence quant à la stratégie à adopter pour accomplir les tâches. De ce fait, il s'agit d'une relation de collaboration. Mais entre deux groupes, les démarches peuvent être différentes et la relation relève de la coopération puisqu'il s'agit essentiellement de s'entraider; chacun pouvant avoir un motif distinct d'y adhérer. De même, deux équipes peuvent travailler des TP différents tout en coopérant.

Le vocable *plate-forme* recouvre plusieurs sens selon le domaine dans lequel il est utilisé (secteur informatique, pétrolier, spatial, politique ou autre) (Site #2). Retenons simplement qu'une plate-forme est un schéma qui présente les principes et les propositions visant à mettre en place une solution. Elle supporte la solution et lui fournit les ressources nécessaires à son fonctionnement.

Habituellement, le *cahier de laboratoire* est un document de travail qui fournit au départ l'énoncé du travail pratique. Durant la séance, le groupe de travail y inscrit ses résultats au fur et à mesure. Sous sa forme électronique, il supporte la collaboration et le travail d'équipe grâce au partage d'informations.

Les *objets d'apprentissage* sont des objets granulaires destinés à encapsuler un contenu d'enseignement. Ils sont décrits par un ensemble de métadonnées (données sur des données) conformément à un standard. Le but est d'en faire ressortir les caractéristiques techniques et éducationnelles les plus pertinentes, facilitant ainsi l'interopérabilité, la recherche et la découverte d'opportunités de télé-apprentissage.

Un *agent logiciel* est une entité virtuelle (i.e un programme), possédant des ressources propres, capable de percevoir son environnement, d'agir sur lui et de communiquer avec d'autres agents. Son comportement vise à satisfaire un certain

nombre d'objectifs. L'agent possède des compétences et offre des services. Il est dit mobile s'il se transfère de nœud en nœud dans un réseau durant son exécution (Site #2).

1.2 Éléments de la problématique

Il est communément admis qu'en sciences et technologie, les travaux pratiques constituent un complément indispensable de tout enseignement théorique. En outre, l'Internet a rendu possible l'interaction entre personnes situées en des endroits géographiques différents mais travaillant sur le même sujet. Alors, pourquoi ne pas s'en servir pour créer des espaces virtuels partagés et y conduire des expériences de laboratoire ?

Une recherche rapide sur le Web montre l'existence de nombreux prototypes de laboratoires virtuels qui s'appuient sur une applet Java pour reproduire un phénomène naturel. Par exemple, la loi des gaz parfaits en thermodynamique : $PV = nRT$ (Site #1).

Dans ce type de laboratoire, l'essentiel de l'effort est consacré à la programmation de l'applet. Cette approche est très simplificatrice puisqu'elle n'aborde que certains aspects de la question complexe de l'expérimentation à distance via un réseau.

En réalité, malgré cette profusion apparente, les laboratoires virtuels émanent d'initiatives de chercheurs isolés ou de responsables de cours qui agissent en pionniers. L'offre n'est pas structurée autour d'une plate-forme et la dimension collaboration n'est pas toujours traitée. Or, dans un laboratoire conventionnel, on travaille en équipe et les équipes s'entraident. Ceci est essentiel d'un point de vue purement pédagogique et la réduction de l'écart entre le virtuel et le réel est un critère fondamental de qualité. À ce jour, il n'existe pas de réalisation connue qui repose sur une démarche globale couvrant tous les aspects du problème : difficultés liées aux réseaux, considérations architecturales, support intelligent à l'expérimentation, gestion de la collaboration, maintien de la sécurité, méthodologie d'analyse. L'objectif final est d'avoir une banque d'objets interopérables servant à constituer des laboratoires virtuels spécifiques autour de la même plate-forme.

La démarche du groupe de recherche LVEST (Laboratoires Virtuels pour l'Éducation en Sciences et Technologie), qui accueille ce mémoire, s'inscrit dans cette logique. Dans un travail antérieur, Kassouf avait proposé un modèle conceptuel pour une plate-forme de télécommunications pour des laboratoires virtuels distribués (Kassouf, 1999 ; Pierre et al., 2001). Celui-ci est représenté à Figure 1.1.

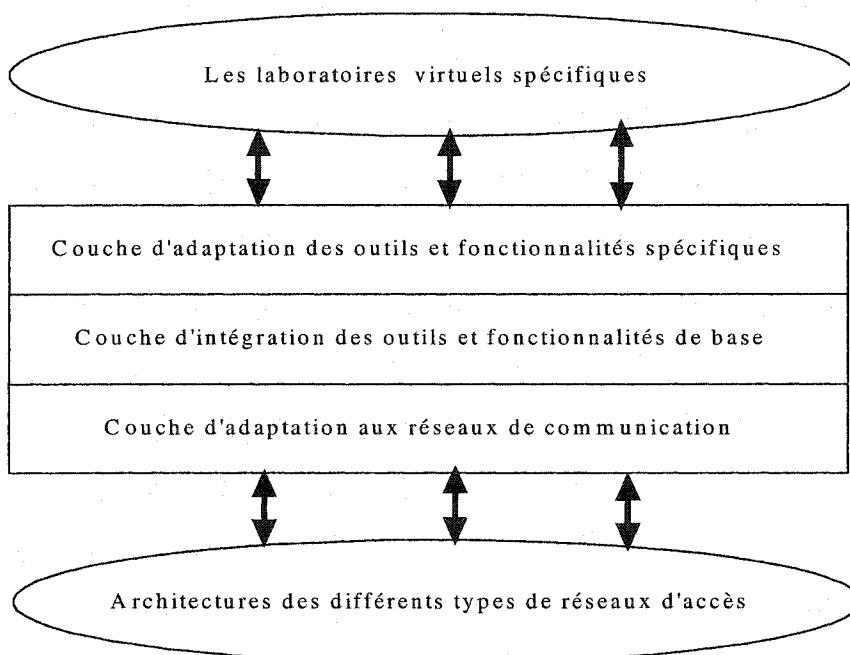


Figure 1.1 Modèle conceptuel de la plate-forme de laboratoires virtuels

Pour la suite, nous numérotions ces couches comme suit:

Couche 1 = couche d'adaptation aux réseaux de communication ;

Couche 2 = couche d'intégration des outils et fonctionnalités de base ;

Couche 3 = couche d'adaptation des outils et fonctionnalités ;

Couche 4 = les laboratoires virtuels spécifiques.

Alors, plusieurs questions fondamentales se posent:

Comment passer du modèle conceptuel en 3 couches à une plate-forme offrant aux développeurs qui le souhaitent un cadre intégré permettant de créer des applications de laboratoires virtuels spécifiques, réparties et collaboratives partageant les mêmes ressources matérielles ? En d'autres termes, comment passer du conceptuel à l'opérationnel ? De même, comment exploiter la plate-forme et subdiviser une classe en groupes de travaux pratiques pour mener des expériences à l'image de ce qui se fait dans un laboratoire conventionnel ?

Une telle évolution ne se fera probablement pas en une seule étape. Elle nécessitera de nombreux travaux et le présent mémoire se propose d'y apporter sa contribution. Nous porterons notre attention sur les couches 2 et 3 qui constituent le socle sur lequel repose la plate-forme d'expérimentation. C'est aussi la limite entre l'infrastructure réseau et les laboratoires virtuels spécifiques.

1.3 Objectifs de la recherche

L'objectif principal de ce mémoire est de concevoir une architecture capable de prendre en compte l'ensemble des fonctionnalités requises d'une plate-forme de laboratoires virtuels répartis. Une de ces fonctionnalités est la prise en compte d'équipes de travail dont les membres doivent collaborer alors qu'ils ne résident pas nécessairement au même endroit. De manière plus spécifique, nous visons les objectifs suivants :

- spécifier les fonctions et les moyens de mise en œuvre des composantes d'une telle architecture ;
- proposer un modèle collaboratif basé sur la plate-forme et permettant de scinder une classe en groupes effectuant des travaux pratiques en équipe ;
- spécifier et implémenter le concept de répertoire de cahiers électroniques de laboratoire basés sur les objets d'apprentissage (learning objects). En effet, par la collecte et le partage d'informations au sein de l'équipe, le cahier constitue un support à la collaboration ;

- concevoir des scénarios d'expérimentation permettant d'évaluer les performances de la collaboration à travers le répertoire.

1.4 Plan du mémoire

Ce mémoire comprend cinq chapitres: outre le chapitre 1, le chapitre deux fait une revue de la littérature scientifique sur les laboratoires virtuels, la collaboration et la standardisation des méta-données (objets d'apprentissage et description de l'apprenant). Le chapitre trois présente l'architecture de la plate-forme d'expérimentation et le modèle collaboratif. Le chapitre quatre est consacré à l'implémentation et à la mise en œuvre du répertoire de cahiers électroniques. Enfin, le chapitre cinq présente en guise de conclusion une synthèse des travaux et suggère quelques idées pour des recherches futures.

CHAPITRE II

LABORATOIRES VIRTUELS, COLLABORATION ET STANDARDISATION DES MÉTA-DONNÉES

Le développement du multimédia et les progrès réalisés en télécommunications fixe et mobile ont créé un contexte favorable à l'expansion des systèmes répartis. Parallèlement, la vie moderne soumet l'être humain à une pression socio-économique constante qui l'oblige à adapter sa manière de vivre, d'apprendre et de travailler. De ce fait, la "virtualisation" croissante et la mobilité accrue sont devenues deux caractéristiques marquantes de notre époque. Ce chapitre présente les différentes classes de laboratoires virtuels et les environnements qui permettent de les produire. Ensuite, le problème de l'espace collaboratif et la standardisation des méta-données qui favorise la production, la réutilisation et l'échange de contenus éducatifs interopérables sont abordés.

2.1 Évolution de l'informatique dans l'éducation

Les années 1960 consacrent l'arrivée de l'informatique dans le monde de l'enseignement et de la formation. C'était l'époque de l'EAO (Enseignement Assistée par Ordinateur). Au cours de la décennie 80, une première évolution entraîne le passage de l'EAO à l'EIAO (Enseignement Intelligemment Assistée par Ordinateur). Elle a été favorisée par les progrès réalisés en intelligence artificielle, notamment dans le domaine des systèmes experts. Le point focal des recherches était le système tutoriel intelligent (STI) qui a pour ambition de fournir à chacun un enseignement qui lui est individuellement adapté. C'était, entre autres, l'idée de la machine à enseigner de Skinner (Tadié, 1998). On parlait alors beaucoup de modèles :

- le modèle de l'apprenant à qui on adapte le contenu de l'enseignement ;
- le modèle du tuteur (l'expert) qui adapte l'enseignement en fonction de la cible ;
- le modèle de la matière qui est la chose qu'on adapte.

Il y eut beaucoup de débats car un modèle peut être difficilement complet et satisfaisant pour intégrer toute la complexité du processus de transmission des connaissances. Les laboratoires virtuels étaient synonymes de simulations.

Dans les années 1990, avec la monté en puissance des réseaux et la popularisation de l'Internet, on passe de l'EIAO au "e-learning" (electronic learning). Ce nouveau concept met l'accent sur le télétravail collaboratif et la flexibilité spatio-temporelle. Alors, on assista à une profusion d'offres de formation en ligne. Au même moment, apparaissent des laboratoires virtuels utilisant la télémesure/télémanipulation. En effet, l'instrumentation à distance accroît la capacité technologique et donc l'efficacité économique. La prochaine étape de cette course qui cherche à « amener l'école à l'élève » et non l'élève à l'école, est le passage du "e-learning" au "m-learning" (mobile learning). Elle vise à associer les téléphones cellulaires et autres assistants personnels (PDA) aux ordinateurs personnels et aux portables (laptops) pour en faire un nouveau canal d'accès au savoir. Mais, de nombreux problèmes, tant techniques que pédagogiques, restent à résoudre.

Sur le plan pédagogique et organisationnel, le passage de l'EIAO au e-learning s'est traduit par une évolution d'un mode totalement autonome (EIAO) à une forme distribuée, en réseau et contrôlée centralement (e-learning). Le travail collaboratif entre apprenants situés en des endroits géographiques différents devient alors possible. Ce qui est plus conforme à nos habitudes sociales. En outre, les solutions e-learning sont généralement bâties autour d'une base de données centrale qui conserve la trace des apprentissages, les détails des cours suivis, les durées d'utilisation et les résultats des exercices traités. Les responsables et les formateurs disposent ainsi d'un moyen de suivi. Si cette centralisation présente des avantages, il faut éviter un retour en arrière (EAO) du fait d'un contenu générique, standard, hébergé sur un serveur unique et destiné à un public large à qui reviendrait la charge d'y trouver ce qu'il recherche. Le Tableau 2.1 résume l'évolution de l'informatique destinée à l'éducation dans les quarante dernières années.

Tableau 2.1 Évolution de l'informatique dans l'éducation

Avancée	Point focal des recherches	Laboratoires virtuels	État actuel
De l'EAO à l'EIAO	Le système tutoriel Intelligent (S.T.I) - S'adapter à la "learning curve" de l'apprenant	Simulations	Déjà mature
De l'EIAO au e-learning	Apprentissage collaboratif - Flexibilité spatio-temporelle	Apparition de la télémesure / télémanipulation	En pleine croissance
Du e-learning au m-learning	Granularité des contenus - Continuité des communications	Non étudié à ce jour	Embryonnaire

Actuellement , on observe les tendances suivantes:

- on fait de plus en plus appel au *Soft computing* et aux agents mobiles pour prendre en compte toute la complexité de l'enseignement, notamment dans les STI;
- avec le passage du e-learning au m-learning, la mobilité change de nature. D'itinérante ou résidentielle (l'usager en perdait le privilège pendant son déplacement entre ses différents lieux de travail), elle deviendra ambulatoire du fait de la continuité des communications grâce aux équipements portatifs. Cette tendance est inspirée par le *pervasive computing* (informatique diffuse);
- pour la présentation multi-support et l'échange d'informations, XML/XSL voire XMI, s'impose en lieu et place de HTML, apportant une plus grande flexibilité ;
- pour la production de contenus, les métadonnées apparaissent comme un thème stratégique. De nombreux travaux sont consacrés à leur normalisation dans l'optique de l'émergence d'un marché mondial du savoir comprenant des fournisseurs de contenus et des consommateurs ;
- quant aux laboratoires virtuels sans lesquels il est difficile d'imaginer une formation de qualité en sciences et technologie, les prototypes basés exclusivement sur la simulation ont montré leurs limites.

2.2 Modèle d'exploitation des laboratoires virtuels

Avant d'établir une typologie des laboratoires virtuels, voyons comment on peut les exploiter. Le modèle de la Figure 2.1 a été tiré de celui proposé par Christian et al. (1999).

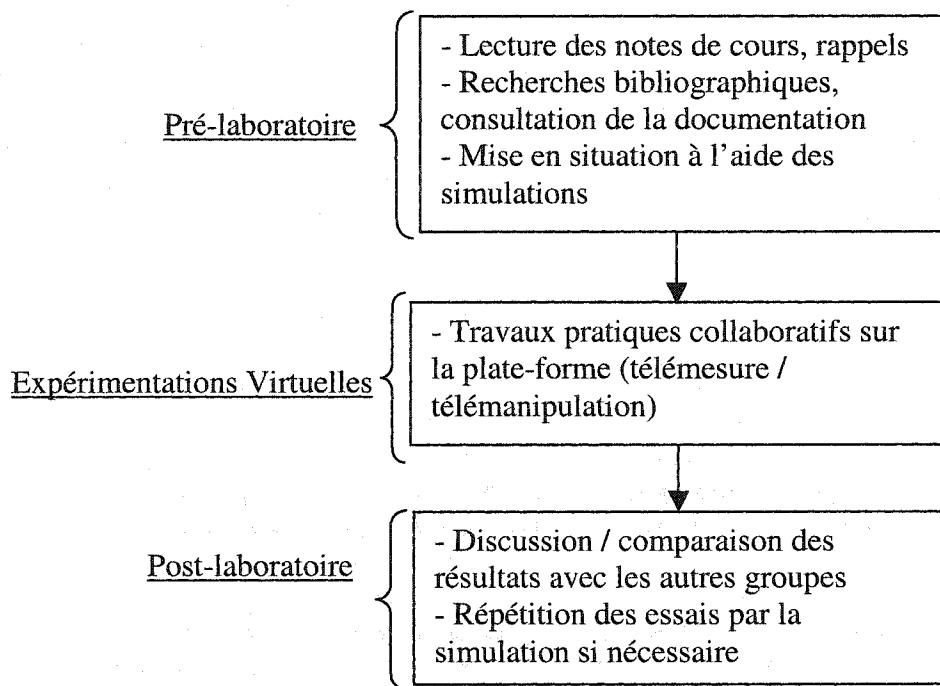


Figure 2.1 Modèle d'exploitation des laboratoires virtuels

Dans ce modèle, l'apprenant s'inscrit sur la plate-forme et réserve une plage horaire. De ce fait, il connaît les membres de l'équipe dans laquelle il fera son travail pratique. Puis, il prépare sa séance de laboratoire en lisant ses notes de cours, en faisant des recherches bibliographiques et en se mettant en situation grâce à la simulation. Après la séance, il peut en débattre avec ses autres collègues et, au besoin, répéter les essais à l'aide de la simulation pour mieux assimiler. À cette fin, il peut sauvegarder ses résultats pendant l'expérimentation et les réutiliser plus tard.

2.3 Typologie des laboratoires virtuels

Dans cette section, nous abordons les différents types des laboratoires virtuels puis passons en revue quelques exemples. Ensuite, nous examinerons les principaux environnements qui permettent de les produire et les raisons pour lesquelles ils ne sont pas satisfaisants. Globalement, on peut dire qu'il existe trois types de laboratoires virtuels :

- les simulations (ou laboratoires virtuels basés sur la simulation) ;
- les télelaboratoires (remote laboratories) ;
- les télelaboratoires avec simulation.

2.3.1 Les simulations

Historiquement, les premiers laboratoires virtuels étaient des simulations. Aujourd’hui encore, c’est de loin la catégorie la plus répandue sur le Web. L’idée de base était d’utiliser la capacité graphique des ordinateurs et l’interactivité des programmes informatiques pour reproduire des phénomènes réels. Les progrès réalisés dans les réalisations virtuelles et l’avènement des applets Java ont été des éléments favorables. Malgré tout, construire un bon système de simulation, aussi proche que possible de la réalité, est une tâche ardue qui implique la prise en compte d’un grand nombre de paramètres. De ce fait, le temps de calcul peut être long, ce qui ne favorise pas l’interactivité requise dans un contexte d’apprentissage.

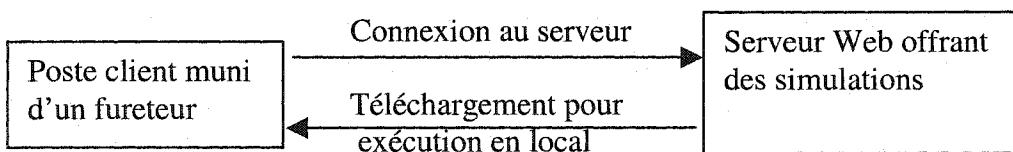


Figure 2.2 Fonctionnement d'un laboratoire virtuel basé sur la simulation

Lorsque le phénomène étudié ne peut être décrit selon des règles précises et/ou une formulation mathématique, la simulation permet néanmoins d'en comprendre les principes généraux. Sa principale limite est qu'elle est artificielle. En effet, ce n'est pas la réalité mais une représentation voire une interprétation de celle-ci.

La Figure 2.2 montre le fonctionnement d'un laboratoire virtuel basé sur la simulation. Le téléchargement peut également être remplacé par une applet Java. C'est d'ailleurs la forme la plus répandue. Examinons quelques exemples de simulations.

Interférence d'ondes électromagnétiques

Cette simulation apparaît à la Figure 2.3 et elle est actuellement disponible sur le site de l'université nationale de Taiwan (Site #5).

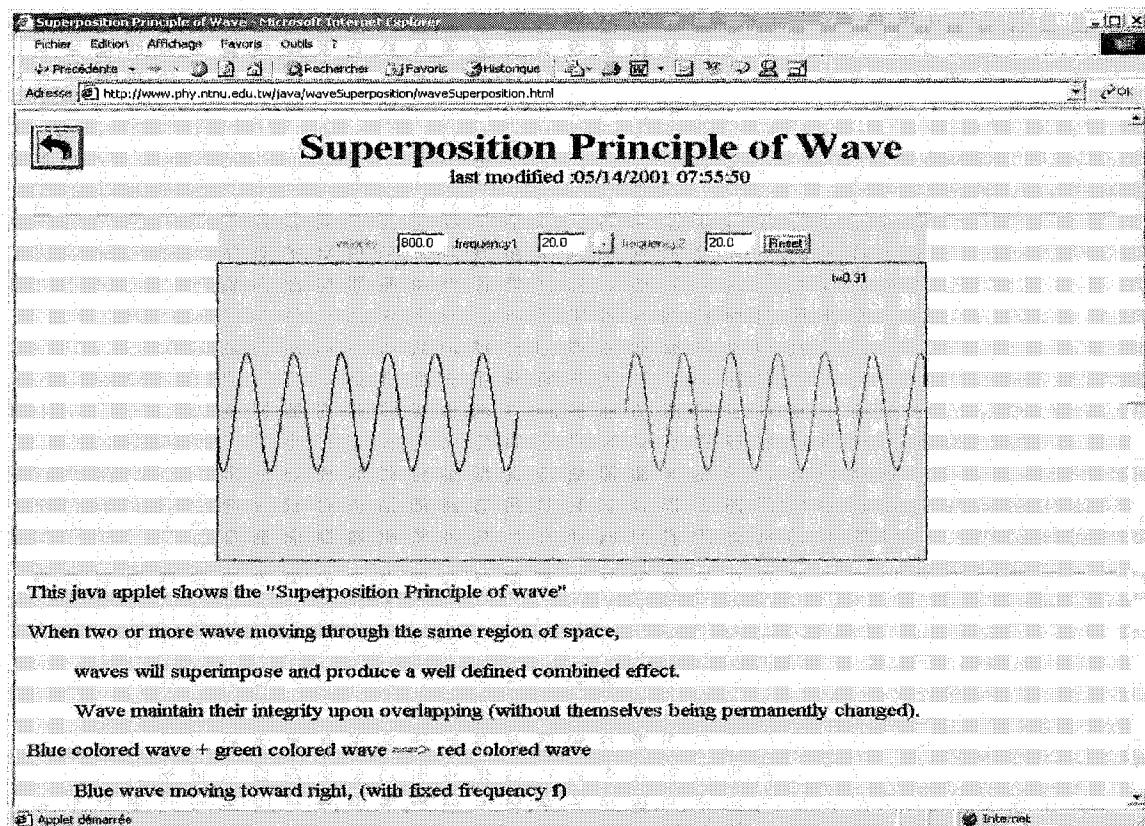


Figure 2.3 Les deux ondes avant l'interférence

Il s'agit d'étudier le principe de l'interférence des ondes électromagnétiques. Deux ondes se propagent l'une en direction de l'autre dans des espaces initialement disjoints. Lorsqu'elles se rencontrent, elles donnent naissance à une troisième onde du fait de l'interférence. Celle-ci coexistera avec les deux premières comme l'illustre la Figure 2.4. L'animation procurée par l'applet Java permet de voir comment tout cela fonctionne. L'apprenant peut faire varier la fréquence des deux signaux initiaux ou leur vitesse commune de propagation et en observer directement les effets sur son écran. Justement, l'interactivité et la possibilité de répéter plusieurs fois chaque essai sont deux des avantages attendus des laboratoires virtuels basés sur la simulation.

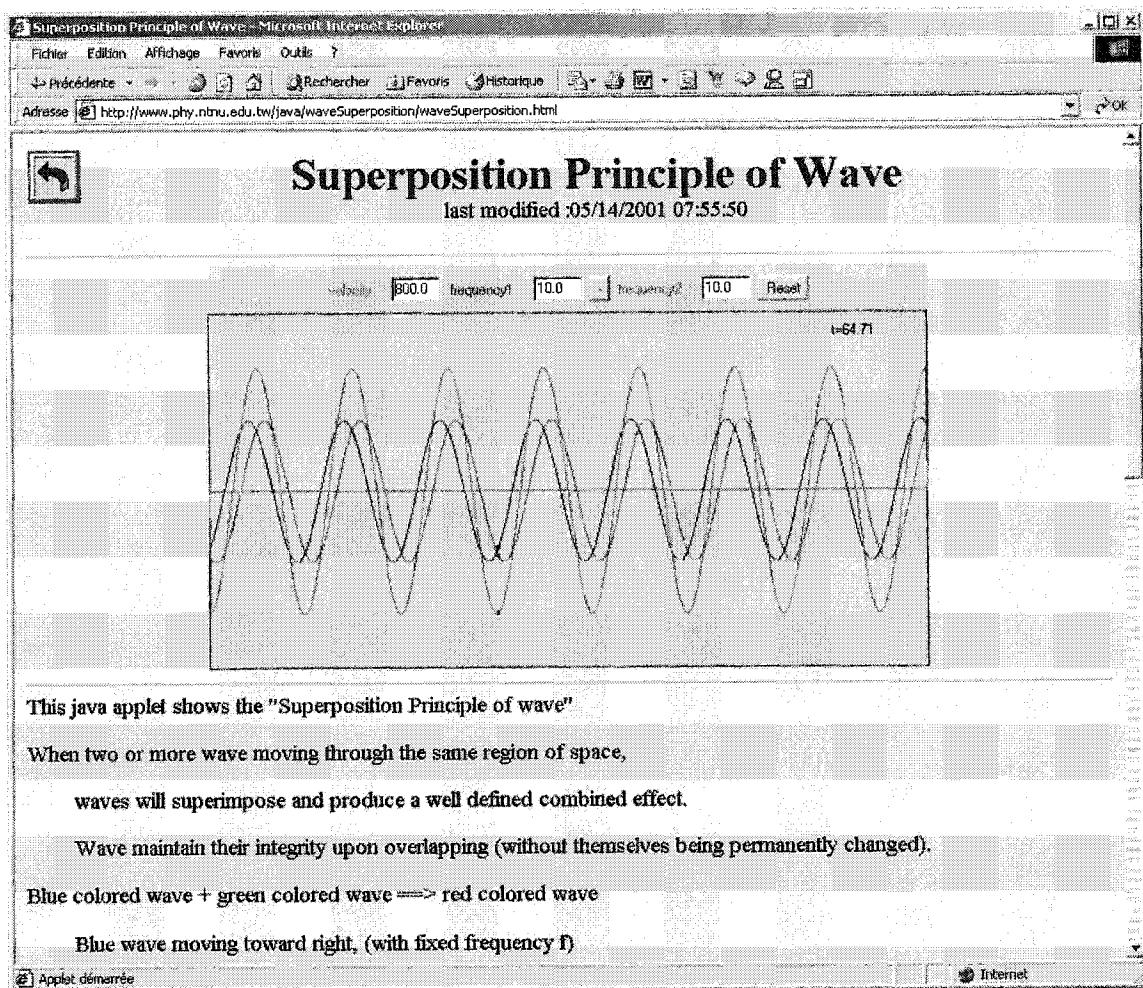


Figure 2.4 Les trois ondes après l'interférence

Ce phénomène d'interférence résulte de la superposition des vibrations ayant la même longueur d'onde mais des phases différentes en un signal uniforme. Il obéit à des lois bien connues en acoustique. Elle est dite constructive lorsque les amplitudes s'ajoutent (c'est le cas Sur la Figure 2.4) ou destructives quand elles se retranchent. L'onde peut être n'importe quelle vibration électromagnétique : son, lumière ou autre.

Étude du tir de canon

Nous avons trouvé ce laboratoire virtuel sur le site de l'université d'Oregon (Site #1). L'objectif est d'étudier la trajectoire d'un obus tiré par un canon. C'est un problème classique de cinématique et il est bien connu des élèves du secondaire. À l'aide du prototype présenté ici, l'élève peut tour à tour faire varier l'angle de tir ou la vitesse du projectile (et donc l'énergie cinétique) afin d'atteindre la cible qui est matérialisée par un petit singe virtuel. Pour chacun de ces paramètres, l'usager peut faire jusqu'à six essais au maximum. En cas de réussite avant l'épuisement de ces six possibilités, le programme passe automatiquement à la série suivante en grisant les paramètres pour lesquels l'essai vient d'être réussi. Dans la partie supérieure gauche de l'écran, un référentiel indique les positions successives de l'obus en fonction du temps durant son parcours. Ceci est illustré à la Figure 2.5.

Limites de ces laboratoires virtuels

Ces simulations sont basées sur des applets Java. L'essentiel de l'effort de développement est mis sur la programmation de l'applet, alors que plusieurs aspects essentiels de l'expérimentation à distance sont occultés:

- il n'y a pas de partage de ressources avec d'autres usagers ;
- il y a absence d'interaction avec des équipements réels, même à distance ;
- l'apport du réseau est marginal puisqu'il ne sert que dans la phase de téléchargement.

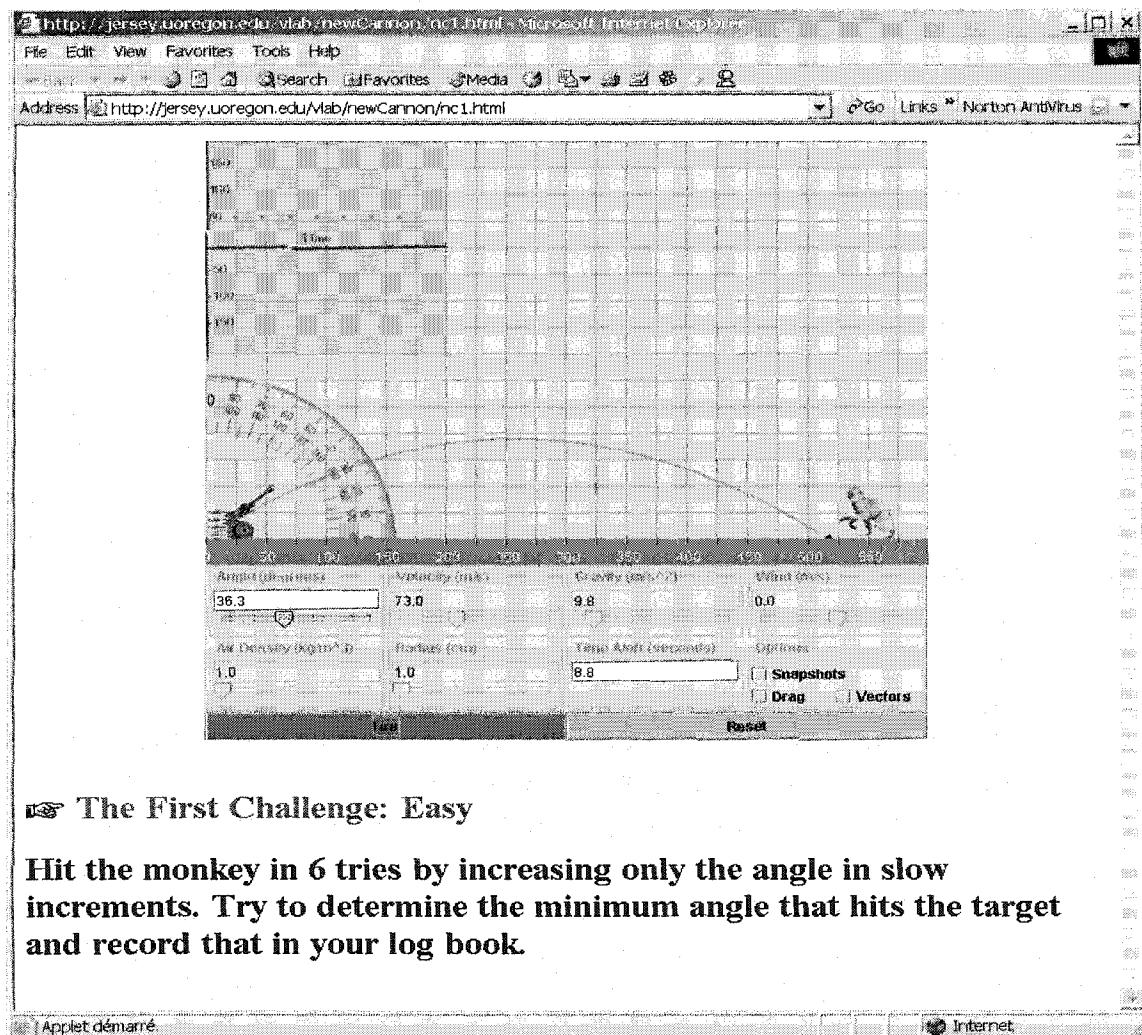


Figure 2.5 Tir de canon en cinématique

La simulation procure des avantages certains mais au regard des limitations qui précèdent, on peut se demander si elle est toujours suffisante quand on l'emploie toute seule. De ce fait, dans bien de cas elle est combinée à la télémesure/télémanipulation.

2.3.2 Les télélaboratoires

S'il existe beaucoup de simulations, les télélaboratoires sont encore relativement rares. Or, l'un des objectifs des laboratoires virtuels est de permettre le partage de ressources et d'accorder ainsi ceux qui n'ont pas facilement accès à certains

équipements (utilisateurs en région éloignée ou matériel sophistiqué, donc rare). Les télélaboratoires reposent sur la télémesure et la télémanipulation. La Figure 2.6 présente l'architecture simplifiée d'un télélaboratoire (Saliah et al., 2000).

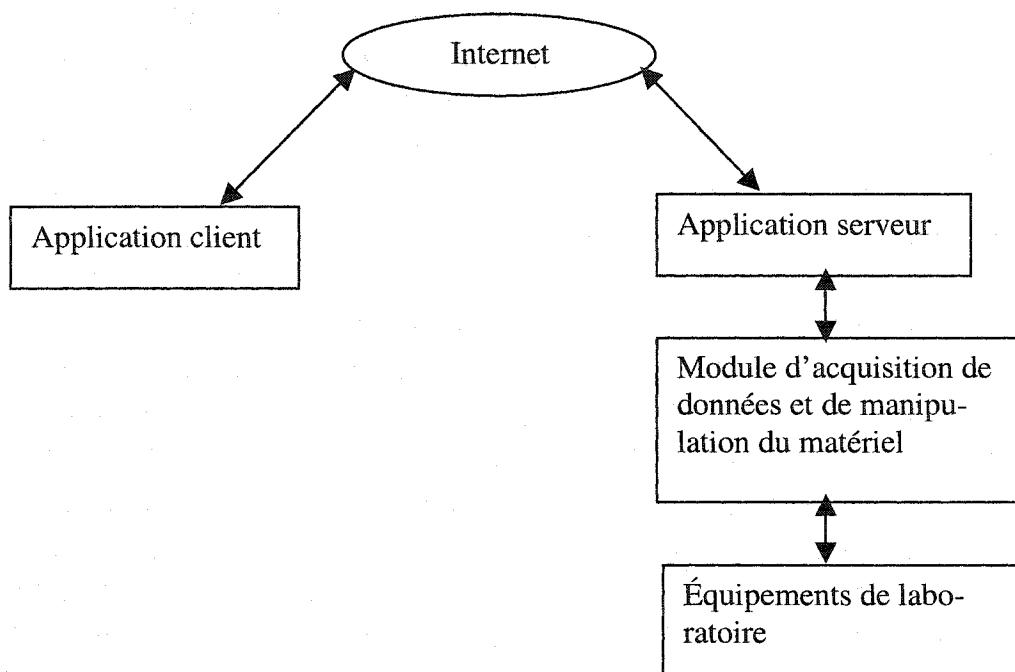


Figure 2.6 Architecture simplifiée d'un télélaboratoire en mode client / serveur

Le *module d'acquisition de données et de manipulation du matériel* comprend des capteurs, des pilotes et des cartes d'interface servant à connecter des équipements de laboratoire (oscilloscope, ampèremètre, etc.) aux serveurs pour les commander à distance. L'originalité des télélaboratoires repose sur l'acquisition des données. Un tel système opère en cinq phases :

- 1- mesure d'une grandeur physique (température, pression ou autre) par des capteurs;
- 2- conversion de la grandeur physique mesurée en un signal électrique approprié (par exemple dont la tension est comprise entre 0 et +5 volts) ;

- 3- numérisation du signal électrique et stockage des valeurs dans une mémoire ou un fichier ;
- 4- traitement et analyse des valeurs numériques obtenues ;
- 5- affichage du résultat sur l'écran de l'apprenant avec possibilité de changer la valeur des paramètres expérimentaux à l'aide du clavier et de la souris.

Les dernières phases de ce processus constituent un instrument virtuel. Celui-ci peut être réalisé avec un logiciel d'applications d'instrumentation comme LabVIEW.

2.4 Environnements de développement des laboratoires virtuels

Jusque-là, il existe deux types d'environnement de développement de laboratoires virtuels : Java (avec les applets) et les environnements de programmation graphique. Dans cette section, nous présentons trois environnements : VLab, OPEA-MAT et LabVIEW.

2.4.1 L'environnement graphique VLab

VLab est un environnement interactif pour la création de laboratoires virtuels simulés. Il a été développé à l'Université de Calgary (Site #6). VLab comprend: un programme de simulation lié au domaine, des objets appelés *unités expérimentales* encapsulant les fichiers de données du laboratoire et des outils de manipulation d'objets. Ces outils sont indépendants de tout domaine. Il s'agit de :

- **Object Browser** qui permet de parcourir l'arborescence des objets avec la possibilité de les copier, déplacer, renommer ou supprimer ;
- **Lab Table** qui est une mémoire temporaire dans laquelle on stocke les objets manipulés pendant l'expérimentation. À la fin de la séance, l'usager peut les rendre persistants en les sauvegardant ;
- **Object Manager** offre des opérations de manipulation des fichiers constituant les objets. Même si l'usager n'a pas une connaissance parfaite de l'application, il peut s'en servir. En effet, dès qu'on invoque un objet, les fichiers qui l'implémentent sont systématiquement copiés dans l'espace de travail Lab Table de manière à préserver les

versions originales. Par la suite, il reviendra à l'usager de sauvegarder ou non les copies sur lesquelles il vient de travailler. Dans l'affirmative, elles remplaceront les versions originales ;

- **Panel Manager** est le gestionnaire de la barre d'outils. C'est une interface graphique permettant de modifier les paramètres de l'expérience, ceux-ci étant décrits dans les fichiers de données qui implémentent les objets ;

- **HyperBrowser** facilite la présentation des objets dans différents contextes. Il offre un système de menu déroulant avec des fonctions applicables aux objets contenues dans des bases de données d'objets : Get object, show text, Next, Prior, etc ;

- **VLab daemon** assure la communication entre les composants et lance l'exécution des objets.

L'utilisateur d'un laboratoire créé avec VLab peut également accéder à des objets distants et leur appliquer les mêmes opérations, dès lors qu'il en a acquis préalablement la permission. Pour cela, il se sert du **remote access server daemon**.

2.4.2 L'environnement OPEA-MAT

OPEA-MAT est l'abréviation de **Outil Pédagogique pour l'Enseignement et l'Apprentissage des MATHématiques**. Il a été conçu à l'Université Laval pour l'enseignement et l'auto-apprentissage des mathématiques appliquées. Nous en parlons ici pour deux raisons :

- il intègre un outil de simulation qui peut être utilisé comme démonstration dans une classe (laboratoire virtuel simulé) ;
- des considérations architecturales : OPEA-MAT a une organisation modulaire dont on peut tirer plusieurs enseignements, notamment pour l'architecture de la plate-forme de laboratoires virtuels que nous allons proposer. OPEA-MAT qui a obtenu plusieurs distinctions, comprend trois modules (Bastien et al., 1999) : un tutoriel, un exerciceur et un démonstrateur.

Le tutoriel : présente des résumés de cours et des rappels. Son rôle est de compléter le cours suivi en classe. Il offre également des quiz (questions simples), des exemples, des contre-exemples ainsi que des notions avancées qui sont des contextes d'utilisation du concept étudié. Pour chaque question, la réponse est fournie sur demande, permettant ainsi une auto-évaluation de l'usager (Bastien et al., 1999). Le tutoriel peut servir de pré-laboratoire dans l'architecture d'une plate-forme de laboratoires virtuels.

L'exerciceur : propose à l'usager des exercices classés en trois niveaux de difficulté. L'apprenant est libre de choisir le niveau qui lui convient. Le niveau 1 représente les exercices de base, le niveau 2 ceux d'une difficulté intermédiaire et le niveau 3 les exercices comparables à un examen formel pour ce niveau d'étude. Pour chaque question, l'exerciceur offre, en plus de la réponse juste, une série de réponses erronées fréquemment rencontrées avec les explications y afférentes. Ces explications varient selon le niveau de difficulté et le nombre de tentatives de résolution. L'exerciceur peut servir de post-laboratoire afin de tester l'assimilation.

Le démonstrateur : permet d'associer par la simulation un phénomène physique à la notion mathématique étudiée. Exemple : les équations différentielles qui régissent les vibrations d'un ressort soumis à l'effet d'un poids. L'élève peut observer graphiquement l'effet de chacun des paramètres apparaissant dans les différentes équations. Ainsi, le démonstrateur permet de choisir une combinaison de facteurs, d'expérimenter puis d'établir un lien avec des concepts de mathématique appliquée. Dans une plate-forme de laboratoires virtuels, il peut tenir lieu de pré-laboratoire.

OPEA-MAT a été utilisée à l'Université Laval et les sondages effectués ont montré une grande satisfaction des apprenants (80% selon Bastien et al., 1999).

2.4.3 L'environnement LabVIEW

C'est un produit commercial de *National Instruments Corporation* qui s'est imposé comme le principal outil de développement de télélaboratoires (remote laboratories).

LabVIEW est l'abréviation de **Laboratory Virtual Instruments Engineering Workbench**. C'est un environnement de programmation graphique, basé sur le langage G. Il permet d'une part le contrôle et l'acquisition des données et d'autre part la présentation et l'analyse des résultats. Malgré sa puissance, LabVIEW nécessite peu d'effort de développement. Il procède d'une logique séquentielle basée sur le contrôle des flots de données entre une source et une destination. LabWindows en est une variante.

Pour comprendre la logique d'un programme LabVIEW, il faut suivre le cheminement des données (i.e le flot) et non l'ordre des instructions. Il n'y a pas de syntaxe à apprendre pour programmer en LabVIEW. Dans un programme LabVIEW, on peut appeler du code LabVIEW s'exécutant sous un autre environnement, ajouter des contrôles *activeX* ou appeler un autre code stocké sous forme de DLL. D'où un environnement totalement ouvert. Plusieurs algorithmes de traitement de signaux et d'analyse arrivent sous forme d'instruments virtuels directement incorporés dans le logiciel. On peut citer : les transformées rapides de Fourier, les fonctions d'analyse de spectres, les transformées de Hilbert, les régressions, les filtres, etc. Les cartes d'acquisition de données vendues dans le commerce sont "plug and play" et intègrent généralement des convertisseurs analogique/numérique et numérique/analogique, ce qui permet l'échange de données entre un PC et un instrument de mesure.

Avec LabVIEW, on crée des programmes appelés *instruments virtuels* (VI). Ensuite, ces derniers peuvent être utilisés seuls ou couplés à un module d'acquisition des données. Dans ce cas, une fois l'acquisition effectuée, les données peuvent être exploitées indépendamment. Comme tout instrument réel, le VI a un panneau avant (Figure 2.7) avec des boutons et/ou des interrupteurs qui permettent de le manipuler. Les sorties sont représentées par des indicateurs lumineux (LED), des graphes ou des diagrammes. Outre les entrées/sorties de l'usager, l'application LabVIEW peut lire directement sur les ports série et/ou parallèle les mesures prises par des équipements. Outre le panneau avant, un VI comprend un diagramme (block diagram) comme celui de la Figure 2.8. Celui-ci décrit le fonctionnement interne du VI. On peut imbriquer des VI les uns dans les autres.

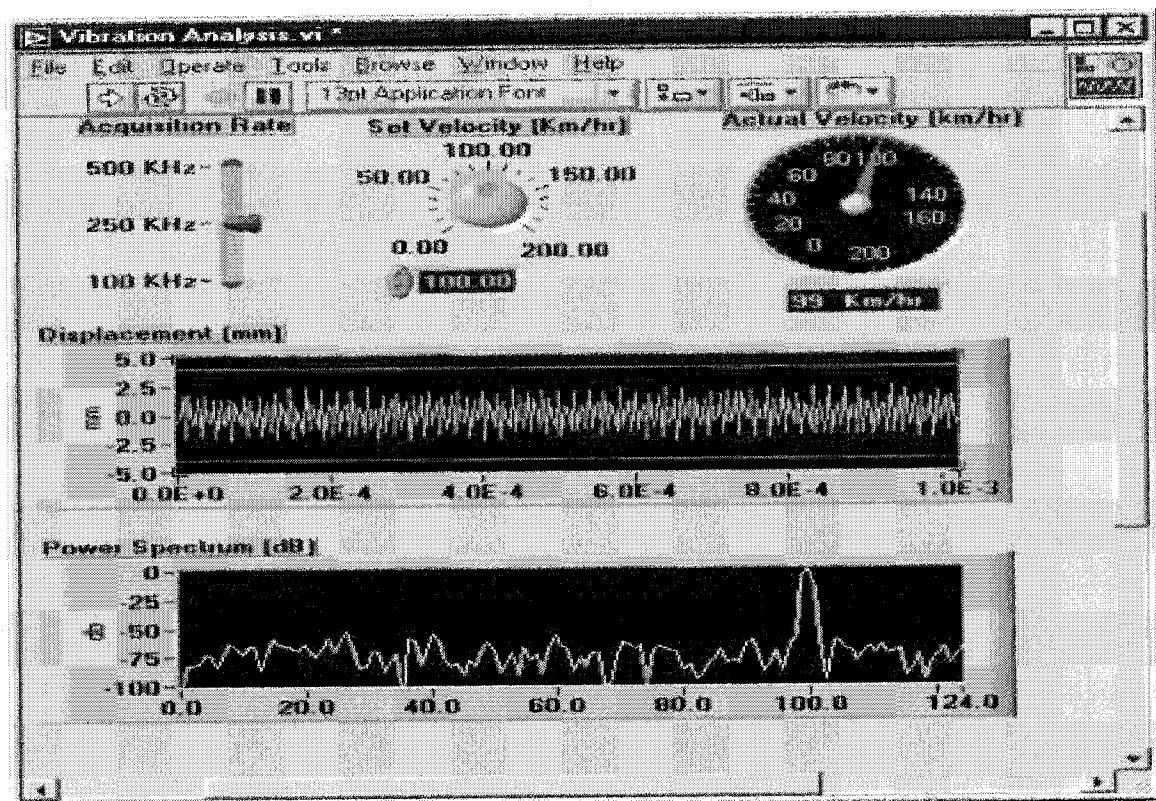


Figure 2.7 Panneau avant d'un instrument virtuel

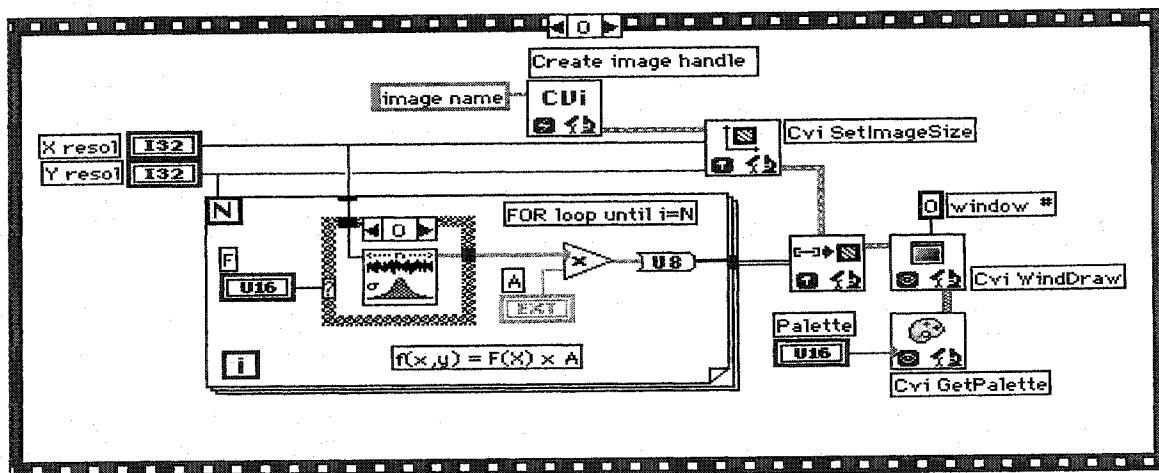


Figure 2.8 Diagramme d'un instrument virtuel

Tout n'est pas de construire des VI. Il faut pouvoir les utiliser à distance. Pour cela, on a plusieurs solutions.

Internet developers toolkit for G (ITK) est une suite de librairies LabVIEW conçue par National Instruments pour opérer des instruments virtuels via un réseau TCP/IP (Internet, intranet, LAN). Ainsi, on peut créer des sockets, les ouvrir, les fermer ou écouter des ports (*port listener*) pour y lire ou écrire des données. D'où la possibilité de télécommander des équipements, d'acquérir et de manipuler des données à distance. D'après Travis (2000), ITK permet :

- la télésurveillance d'un processus qui s'exécute sur un serveur distant à partir d'un poste client. Exemple: observation d'un VI dans un fureteur sur Internet ;
- la télémanipulation (commande d'un processus distant) ;
- le calcul distribué. Exemple : un calcul fastidieux qui s'exécute simultanément sur plusieurs machines reliées par réseau ;
- l'envoi de courriels à partir de l'application LabVIEW ;
- l'échange de fichiers avec des serveurs FTP.

Outre les VI TCP/IP, LabVIEW offre deux autres mécanismes de communication pour l'exploitation de télélaboratoires. Il s'agit de :

- VI server / VI Client ;
- DataSocket server / DataSocket Client. Là, les clients lisent ou écrivent des données sur le serveur (i.e mécanisme publisher/subscriber). Le client et le serveur peuvent être chacun *publisher*, *subscriber* ou les deux à la fois. Le client est une API (DataSocket API for clients).

AppletVIEW est un produit de *Nacimento Company* permettant de visualiser dans un fureteur un VI qui s'exécute à distance. Il est basé sur des applets Java interfacés avec LabVIEW. Un de ses avantages est qu'il utilise les structures de programmation de LabVIEW, ce qui fait que le développeur n'a pas besoin d'apprendre un autre langage.

Avec LabVIEW, l'accès aux ressources peut être sécurisé soit par des mots de passe soit par des "firewalls". Ce qui fait la force de cet environnement, c'est la facilité de

contrôler l'échange de données entre PC et équipements de mesure, le grand nombre d'algorithmes implémentés sous formes de VI qui arrivent avec LabVIEW, la possibilité d'incorporer du code provenant d'autres applications grâce au *code interface node* (CIN) et le fait d'avoir des instruments virtuels utilisables à la place des équipements réels (Gulotta, 1995). Lorsqu'il est couplé à un module d'acquisition de données, le VI est limité essentiellement par la carte d'acquisition des données. LabVIEW peut apporter deux choses dans un contexte d'enseignement : suppléer les équipements obsolètes par des instruments virtuels et simuler ceux qu'on n'a pas.

2.4.4 Limites de ces environnements

Des trois environnements présentés, VLab se destine explicitement à la production des simulations d'expériences, alors que OPEA-MAT est plus un "courseware" (logiciel spécialisé pour l'enseignement d'une discipline) doublé d'un simulateur. LabVIEW est sans doute le plus élaboré d'entre eux. Il permet de concevoir des télélaboratoires où l'on interagit avec un équipement réel, distant ou non, au moins à un moment donné. L'acquisition de données correspond à la télémesure / télémanipulation qui est absente dans les autres environnements. Néanmoins, les problèmes suivants demeurent:

- comment fédérer un grand nombre de laboratoires virtuels spécifiques, partageant les mêmes ressources et en structurer l'exploitation ? En effet, comme nous l'avons vu, l'une des tendances du "e-learning" est d'aller vers la structuration des offres de formation en ligne. Or, sur le Web les laboratoires virtuels émanent actuellement d'initiatives de chercheurs isolés ou de responsables de cours qui agissent en pionniers. D'où la nécessité d'avoir une plate-forme ;
- dans un laboratoire réel, l'expérimentation est généralement une affaire de groupe. Or, travailler ensemble suppose l'existence d'un espace de collaboration où chacun peut voir ce qui se fait, dialoguer avec les autres, coordonner ses actions avec eux et intervenir pour contribuer à la réalisation de la tâche commune (i.e l'expérience). Aussi, des personnes appartenant à des groupes différents pourront-ils s'entraider.

2.5 L'espace collaboratif dans d'autres domaines informatiques

L'espace collaboratif offre des ressources supportant le travail d'équipe au sein d'une classe répartie en groupes effectuant chacun un TP. Plusieurs chercheurs en ont abordé divers aspects sous des angles autres que celui des laboratoires virtuels. On peut citer l'intelligence artificielle distribuée, les logiciels de travail en groupe (groupwares) et les grappes d'ordinateurs (computer clusters).

2.5.1 En intelligence artificielle distribuée

Habituellement, un groupe d'agents appelés encore *sources de connaissances* se sert d'un "blackboard" (tableau noir) pour communiquer et partager des connaissances, des faits, des règles, des commandes, bref, toute sorte d'informations permettant de résoudre un problème donné dans une démarche collaborative. Les connaissances requises pour résoudre le problème sont subdivisées entre ces sources de connaissances. Chaque agent est indépendant mais ils peuvent s'exécuter tous en parallèle et être géographiquement distribués. Ils interagissent et communiquent UNIQUEMENT via le blackboard qui reflète l'état d'avancement de la tâche. À chaque étape, le contrôleur du blackboard détermine le sous-problème qui sera traité à la prochaine itération. Les sources de connaissances apportent des modifications successives au contenu du blackboard, ce qui conduit progressivement à la résolution du problème.

Il existe aussi des modèles de blackboards hiérarchisés où les agents sont groupés par champ de compétence. Ainsi, chaque groupe résout les problèmes relevant de sa spécialité et l'ensemble est synthétisé progressivement dans un blackboard central comme solution d'un problème complexe. Cette démarche est appelée *distributed problem-solving*. Elle repose sur la planification des tâches et la complémentarité pour atteindre un but. DYNACLIPS et CAIBL sont deux exemples de systèmes basés sur le blackboard.

DYNACLIPS

C'est un schéma d'échange dynamique de connaissances entre agents intelligents via un ensemble de blackboards (Cingeloglu et al., 1994). Chaque agent est un shell CLIPS exécutant un processus indépendant sous SunOS. Les échanges ne modifient pas les agents mais peuvent leur apporter de nouvelles connaissances et accroître leur capacité à régler des problèmes. Les éléments caractéristiques de DYNACLIPS sont :

- l'utilisation de IPC (interprocess communication de SunOs) comme mécanisme d'échange de messages ;
- l'implantation du blackboard sous-forme de mémoire partagée ;
- l'utilisation d'une file d'attente (FIFO) pour recevoir les requêtes des agents participants à la résolution du problème. Ceci permet de synchroniser les mises à jour du blackboard ;
- le contrôle de la concurrence par l'utilisation de sémaphores ;
- chaque processus (agent ou contrôleur) dispose d'une adresse d'entrée / sortie (un numéro de port) comme interface de communication avec les autres applications (monde extérieur).

CAIBL (Cooperative Architecture of Independent BLackboards)

CAIBL est une architecture coopérative qui fédère des systèmes experts distribués (ou sources de connaissances) s'exécutant sur des plate-formes hétérogènes et participant à la résolution d'un problème complexe (Brandon, 1990). L'objectif visé est de fournir un schéma flexible capable de gérer les problèmes posés par la distribution et l'exécution concurrente. CAIBL utilise des tableaux noirs orientés objet (*object-oriented blackboards*) dans lesquels les informations sont encodées sous formes d'objets. D'un point de vue structurel, les systèmes experts sont regroupés par spécialité et chaque spécialité est représentée par un objet qui lie lesdits systèmes experts. L'auteur estime que ceci facilite le contrôle des tâches puisque chacune d'elles correspond à un objet unique (i.e la spécialité y afférente). Aussi, chaque système expert implante une politique de communication qui définit : quand envoyer des messages, quel type

d'informations transmettre, à qui envoyer, quand recevoir un message, de qui accepter de recevoir.

Habituellement, le modèle du blackboard s'organise autour de trois composantes:

- **les sources de connaissances** qui se répartissent le savoir requis pour résoudre le problème. Elles peuvent être distribuées ;
- **le blackboard** qui contient à tout instant l'état actuel du problème soumis à la résolution ;
- **un mécanisme de contrôle** qui gère les changements apportés au contenu du blackboard et décide quel sous-problème aborder à l'itération suivante.

De manière analogue, dans un groupe de laboratoire virtuel, deux ou trois personnes mènent une expérience en échangeant des informations et en collaborant sans être nécessairement au même endroit. Aussi, on peut avoir plusieurs groupes qui discutent et s'entraident. Ceci résulte en une organisation à deux niveaux : collaboration à l'intérieur de chaque groupe et coopération entre les groupes. D'où une structure non uniforme et irrégulière, contrairement aux blackboards hiérarchiques dans lesquels chaque niveau comprend les trois éléments de base (sources de connaissances, blackboard et mécanisme de contrôle). Aussi, dans l'espace collaboratif de travaux pratiques, il y a une différence fondamentale entre la nature des interactions dans les groupes et celles entre les groupes. Néanmoins, l'idée même du blackboard comme espace de partage d'informations, reflétant à tout instant l'état d'avancement du problème traité nous semble intéressante pour notre propre modélisation. Il en est de même du mécanisme de contrôle des mises à jour. Par contre, la notion de partage du savoir requis entre plusieurs sources de connaissances et l'utilisation du blackboard comme canal UNIQUE de communication entre les agents est une vue propre à l'intelligence artificielle distribuée.

2.5.2 Dans les logiciels de travail en équipe

Les logiciels de travail en équipe (groupwares) reposent sur l'action concertée, le partage des données, la gestion de la concurrence et la télé-présence. Begole (1998)

dresse un large éventail des artifices utilisés pour traiter le problème de la télé-présence. Globalement, on distingue deux catégories de groupwares: ceux qui utilisent **le partage d'application** et ceux qui s'appuient sur **la réPLICATION**. Dans le partage d'application, une seule instance de l'application est exécutée et son résultat est intercepté puis diffusé sur les écrans des membres du groupe. La synchronisation repose sur un jeton tournant qui donne le droit d'écriture uniquement à celui qui l'acquiert. Le partage d'application garantit la **cohérence stricte** puisque tous les usagers voient toujours la même chose. Mais, il génère un trafic réseau important, d'où une inefficacité surtout dans un contexte de réseau étendu. Dans cette catégorie, on peut citer tous les produits basés sur X Window^(*). Microsoft NetMeeting est un autre exemple, mais il utilise le protocole graphique T.128.

Les groupwares basés sur **la réPLICATION** diffusent des évènements uniquement lors des mises à jour. La bande passante est épargnée, **la cohérence est relâchée** mais il faut la maintenir. Ceci montre qu'il est possible d'utiliser la réPLICATION dans l'espace collaboratif de façon à ne propager que des événements de mise à jour consécutifs, par exemple, à la modification du montage expérimental (i.e la tâche effectuée). D'ailleurs, la même démarche est adoptée par certains logiciels de jeux sur Internet (distributed multiplayer games).

2.5.3 Dans les grappes d'ordinateurs

Les grappes visent à remplacer la machine parallèle traditionnelle à la fois très chère, monolithique et complexe, souvent conçue pour réaliser une tâche spécifique (prévision météorologique, calculs scientifiques complexes ou autre) par un réseau de stations de travail avec un meilleur rapport qualité / prix. Ces stations constituent une communauté de calcul dans laquelle on partage des ressources et on les utilise de manière coordonnée. Parmi les thèmes étudiés dans ce cadre, on peut citer : l'usage optimal de la bande passante et la mémoire virtuelle distribuée. Le premier rejoint la réPLICATION et la

^(*) X window est un logiciel client/serveur basé sur un protocole graphique permettant de reproduire sur des clients distants l'écran du serveur. SharedX (HP), ShowMe (SUN), XTV utilisent X window.

cohérence relâchée dont on a parlées à la section précédente. Mais qu'est-ce qu'une mémoire virtuelle distribuée ? Comment fonctionne t-elle ? Quels sont les problèmes posés par sa mise en œuvre ?

Pour faire communiquer des processus distribués on a trois solutions: l'échange explicite de messages (exemple RPC), la lecture / écriture dans une mémoire partagée et le réseau à capacité d'adressage dans lesquels un nœud peut lire ou écrire directement la mémoire d'un autre nœud.

« une mémoire virtuelle partagée est un dispositif logiciel utilisant certains mécanismes matériels (comme la pagination et les interfaces de communication) pour donner à des applications l'illusion d'une mémoire commune, partagée, sur un système où la mémoire est physiquement distribuée...Pour le programmeur, une mémoire partagée distribuée est un espace linéaire d'adressage accessible par tous les processus formant une application parallèle. Les données y sont identifiées par des adresses » (Mentré, 2001)

Dans les faits, la mémoire virtuelle distribuée réplique les données dans la mémoire locale des nœuds participants. La cohérence est maintenue par un protocole de cohérence intégré au logiciel qui implémente la mémoire. La sémantique employée est telle que l'effet de toute écriture est directement visible par l'ensemble des usagers et toute lecture retourne la dernière valeur écrite. Cependant, le principal frein à la généralisation de cette technologie reste la granularité des pages, c'est à dire le volume élémentaire de données lues ou écrites à chaque accès. En effet, une page ne correspond pas à une variable unique manipulée par l'application, même si elle est traitée comme un bloc indivisible. Soient P1 et P2 deux processus, x et y deux variables situées sur la même page. Si P1 écrit sur x et qu'au même moment P2 écrit sur y, ceci sera interprété comme une mise à jour concurrente alors qu'il ne le devrait pas. C'est le problème du faux partage dû à une insuffisance de granularité. Cependant, des solutions comme *Multiview* de *Israël Institute of Technology* permettent d'y remédier en projetant la page contenant x et y sur deux vues contenant chacune une variable afin d'éviter le conflit (Ayal et al., 1999). De cette manière, chaque variable accédée est associée à une vue et

les règles de cohérence (éventuellement le verrouillage) sont appliquées à la vue. Alors, la taille d'une vue peut aller de la taille d'une page mémoire à celle de la plus petite unité d'information adressable par l'application qui utilise la mémoire virtuelle distribuée.

En conclusion, la mise en grappe d'ordinateurs nous montre que l'utilisation de la mémoire virtuelle distribuée est possible et qu'il existe des mécanismes permettant d'assurer un niveau de granularité satisfaisant pour éviter le faux partage. Ceci est d'autant plus intéressant que le blackboard a été souvent implanté sous-forme de mémoire partagée.

2.6 La standardisation des méta-données

Dans tous les débats sur l'ingénierie de la formation en ligne, les méta-données sont considérées comme un thème stratégique. Mais qu'est-ce que les méta-données ? Littéralement, ce sont des données sur les données. Plus concrètement, c'est un groupe d'attributs qui décrit un objet.

Exemple 1: dans une bibliothèque, un livre peut être décrit par une côte, un numéro (ISBN), un titre, un auteur, un éditeur, une année de publication et une table de matière. La côte est attribuée par la bibliothèque alors que l'ISBN (International Serial Book Number) est l'identifiant international utilisé par tout le monde. La normalisation peut dire que la côte est unique, de type alphanumérique et sa présence est obligatoire, le numéro (ISBN) est numérique, unique et obligatoire, le titre et l'auteur sont alphanumériques et obligatoires, le reste est alphanumérique et de présence facultative.

Exemple 2 : La notice qui accompagne tout produit pharmaceutique est une illustration des méta-données. C'est si précieux que la loi interdit pratiquement de commerçer ces produits sans leurs notices. Pourtant, quand on achète, ce n'est pas la notice qui nous intéresse mais le produit lui-même. Aussi, si les notices sont écrites selon un standard, il est aisément de comprendre qu'on n'aura aucune difficulté à retrouver tous les produits pharmaceutiques qui obéissent à un ensemble de critères. Il suffit d'exprimer cette requête dans les termes des méta-données incluses dans le standard. Ainsi, on aura tous

les produits qui nous intéressent et rien d'autres, contrairement aux moteurs de recherche actuels sur le Web. D'où une pertinence plus accrue.

Avec les métadonnées, on peut identifier, définir, classer et rechercher de manière normalisée et efficace chaque objet. L'objet peut être une ressource, un acteur ou une action. Cette démarche facilite la réutilisation, l'interopérabilité et l'échange de contenu. Ainsi, un cours peut être composé à la volée (*on the fly*), donc sur mesure, au moment où la requête est reçue par un serveur de cours. Pour ce faire, l'application sélectionne et combine des contenus d'origines diverses et variées (*scratching*) mais bien adaptés à la situation. On parle alors de «granules pédagogiques».

Derrière cette philosophie, il y a également une volonté de structuration d'un marché mondial du télé-enseignement, par l'émergence de fournisseurs de contenus (providers) et de clients (consumers). Le PRÉAU, équipe de veille technologique de la chambre de commerce et d'industrie de Paris (Site #7), soutient que « *la maîtrise des métadonnées est stratégique car le modèle gère les interfaces entre projets, contrôle les échanges et les flux et influe directement sur le modèle pédagogique... La compétition est internationale dans un marché oligopolistique* » (Préau, 2002).

D'autre part, en plus d'être un pas vers le Web sémantique, la normalisation procure de nombreux avantages. En amont, les "meilleures pratiques" (best practices) mises en place par des groupes de professionnels évolueront progressivement vers des outils méthodologiques. En aval, des critères d'évaluation émergeront pour permettre un contrôle de qualité, ce qui accroîtra la confiance des consommateurs et le développement du marché. Les principaux acteurs de la standardisation sont :

- Dublin Core Metadata Initiative (DCMI): fondé en 1995 à Dublin dans l'Ohio (USA), c'est un regroupement international et interdisciplinaire qui a proposé une ontologie basée sur RDF (Resource Description Format), puis RSS (RDF Site Summary) qui permet d'exprimer différents vocabulaires par un modèle et une syntaxe XML. Puis, XML facilite le transport et l'échange de documents (Site #8) ;
- IMS Global Learning Consortium, Inc.: consortium d'émanation américaine financé par des compagnies privées (IBM, Oracle, Sun, Cisco), des fondations

- (EDUCAUSE) et certains organismes publics depuis 1997. Il a déjà proposé une spécification sur les contenus (*content packaging*), les questions et tests interopérables (*QIT*), les informations sur l'apprenant (*learner information packaging*), l'échange des données (*enterprise data exchange*), des schémas d'utilisation avec XML (XML bindings) ainsi que des "meilleures pratiques" d'implémentation (best practice and implementation guide) (Site #9);
- ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe) est l'équivalent européen de IMS. Il a fait plusieurs propositions dans le cadre de la normalisation. ARIADNE est un projet de l'Union Européenne ;
 - IEEE Learning Technology Standard Committee (LTSC): coordonne les efforts de normalisation et a déjà défini le Learning Object Metadata (LOM) en s'appuyant sur les propositions faites par IMS, ARIADNE et Dublin Core ;
 - International Standards Organization (ISO) : prépare une norme sur l'apprentissage électronique (e-learning) avec comme date butoir 2003 (groupe de travail JCT1 / SC 36).

Il existe de nombreux autres acteurs comme le Department of Defense (ADL) (Site #10), l'industrie de l'aviation (AICC), le CanCore (organisme canadien), etc. Ce qui montre la taille des enjeux. La Figure 2.9 montre l'évolution historique de la standardisation. On y retrouve les acteurs majeurs du domaine ainsi que les inter-relations entre leurs propositions. Leurs travaux se complètent car les uns s'appuient sur les résultats des autres pour faire évoluer les débats. C'est ce que matérialisent les flèches qui apparaissent sur la Figure 2.9.

Réaliser une infrastructure informatique ouverte de télé-enseignement basée sur les métadonnées nécessite la représentation du contenu de l'enseignement (*learning content*) par des objets d'apprentissage. Pour être complet, on a également besoin d'un standard décrivant la cible de l'enseignement qu'est l'apprenant (*learner information package*). Ceci facilitera l'échange de données entre institutions et le suivi de l'apprenant dans son cheminement à travers le cursus de formation.

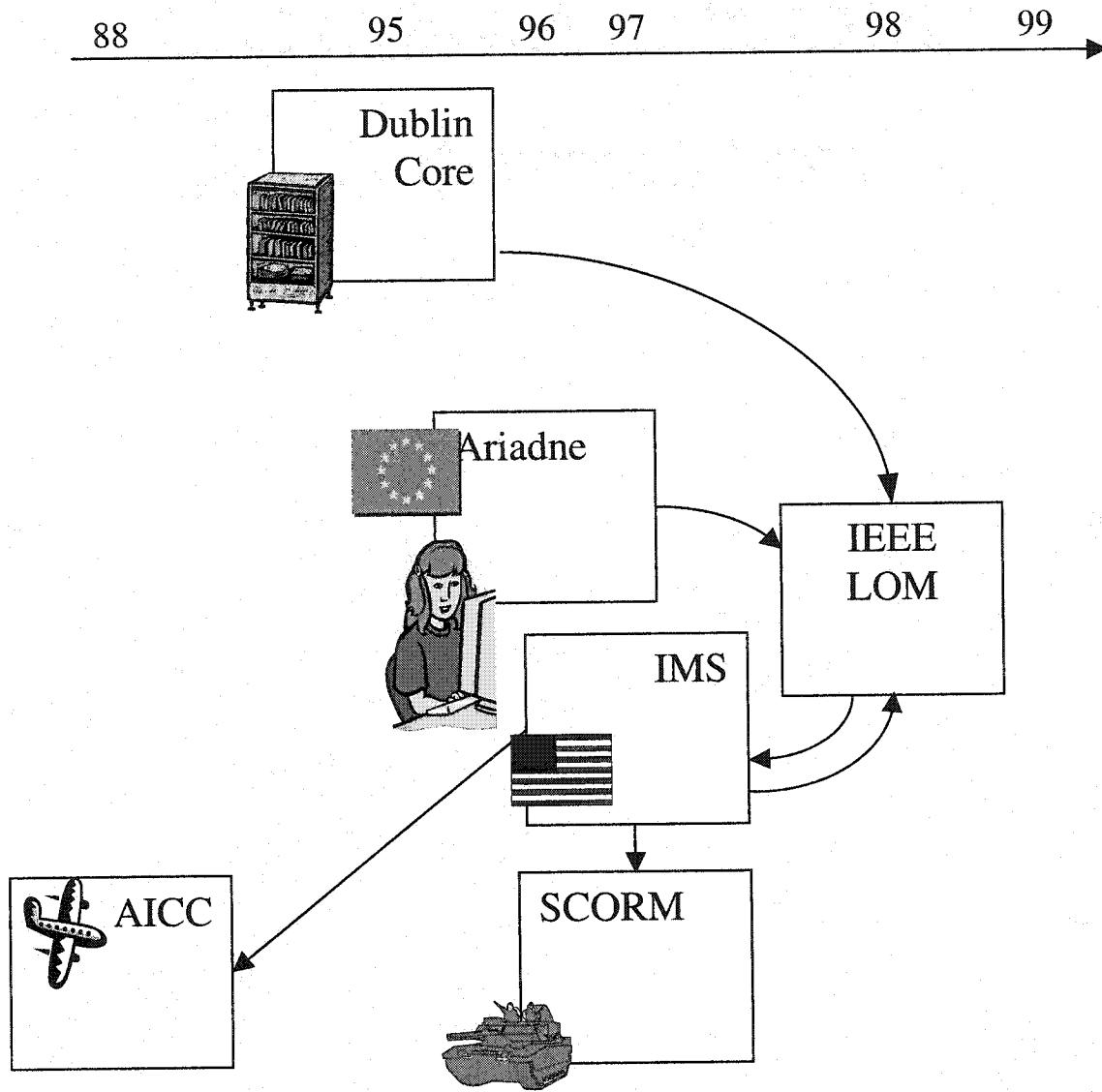


Figure 2.9 Évolution de la standardisation des méta-données (Chung et al., 2002)

2.6.1 Le Learning Object Metadata (LOM)

Comme indiqué au chapitre 1, les objets d'apprentissage sont des composants granulaires (objets) permettant d'encapsuler un contenu d'enseignement. En termes simples, un objet d'apprentissage n'est rien d'autre qu'un contenu d'enseignement décrit par un ensemble de méta-données conformément à un standard. De cette façon, le design pédagogique repose sur deux concepts : granularité et combinaison. L'un des bénéfices

attendus est celui de pouvoir générer dynamiquement des cours personnalisés et bien adaptés à des besoins spécifiques (Wiley, 2000). Quand on mesure toute la difficulté que les institutions conventionnelles de formation ont à répondre cas par cas à chaque apprenant, on comprend aisément la portée d'une telle approche.

Learning Object Metadata (LOM) est le standard de méta-données proposé par IEEE pour décrire les objets d'apprentissage. Selon le draft D6.4 du LOM (IEEE, 2002):

« The purpose of this standard is to facilitate search, evaluation, acquisition, and use of learning objects, for instance by learners or instructors or automated software processes. The purpose is also to facilitate the sharing and exchange of learning objects, by enabling the development of catalogs and inventories while taking into account the diversity of cultural and lingual contexts in which the learning objects and their metadata will be exploited ».

Ce standard (LOM) procure un modèle sémantique décrivant les propriétés des objets d'apprentissage indépendamment de la manière dont ils seront exploités dans un contexte particulier de télé-enseignement. Ceci favorise la mise en place de bases de données d'objets d'apprentissage et la réalisation de moteurs de recherche efficaces. Une description en LOM comprend neuf groupes thématiques de méta-données appelés catégories et ayant chacun une fonction. Chaque catégorie comprend plusieurs méta-données dont les détails sont présentés dans l'annexe A. Le Tableau 2.2 en donne un aperçu.

Les critiques adressées au LOM portent essentiellement sur deux aspects :

- 1- la difficulté d'exprimer les liens qui existent entre les valeurs de certaines méta-données. Exemple : dans la catégorie 5, le 5.9 "Typical learning time" dépend de 5.7 "Typical age range" mais l'arborescence LOM n'en rend pas compte (Suthers, 2001) ;
- 2- au delà des inter-relations, comment déterminer objectivement certaines valeurs ? Exemple : le 5.9 "Typical learning time" (temps d'apprentissage typique) d'un objet d'apprentissage.

Ces observations sont très pertinentes, mais comme le dit Edward Walker, président

de IMS « no standards, no coalitions, no boom» (Edward, 2000). Autrement dit : sans standards, pas de regroupements (de professionnels) et sans regroupements pas de développement. De ce fait, le standard continue d'être affiné et amélioré pour tenir compte des critiques.

Tableau 2.2 Catégories du LOM

Catégorie	Fonction	Exemple
1- General	informations générales sur l'objet	titre, mots clés
2- LifeCycle	évolution et état actuel de l'objet	Draft, final, version 1.2
3-Meta-metadata	informations sur les méta-données	LOMv1.0
4- Technical	données technique concernant l'objet	Vidéo/Mpeg, Unix OS
5- Educational	caractéristiques pédagogiques de l'objet	Easy, difficult
6- Rights	droits d'auteur et aspects légaux	Cost = « Yes »
7- Relation	liens avec d'autres objets d'apprentissage	Ispartof, IsbasedOn
8- Annotation	personnes ayant revu et corrigé l'objet	Agence de certification
9- Classification	extensions et besoins spécifiques	

2.6.2 Le Learner Information Package (LIP)

Proposé par IMS, LIP s'est appuyé sur les travaux antérieurs de PAPI Learner (Public And Private Information on Learner) de IEEE. Celui-ci visait la mise au point d'une description "portable" de l'item apprenant dans un système éducatif (portable learner record). Comme l'illustre le Tableau 2.3, LIP comprend onze catégories de métadonnées permettant :

- le recueil et la gestion de données relatives à l'histoire académique, professionnelle ainsi qu'aux objectifs poursuivis par l'apprenant ;
- la gestion d'une session d'apprentissage ;
- l'échange de données relatives aux apprenants entre systèmes de gestion ;
- la recherche et la découverte d'opportunités adéquates de télé-apprentissage sur

le Web.

Comme pour le LOM, chacune de ces catégories se compose de plusieurs métadonnées. A titre illustratif, l'annexe B présente de façon détaillée la catégorie "Goal".

Tableau 2.3 Catégories du LIP

Catégorie	Fonction	Exemple d'information
Identification	Informations d'identification de l'élève	Nom, adresse, courriel
qcl	Qualifications, license and certification	Diplômes reçus, certifications
Accessibility	Profils et préférences	Langues parlées, handicaps
Activity	formation professionnelle, civique, ...	Expérience de travail
Goal	Objectifs et buts visés	Option de spécialisation
Competency	Connaissances déjà acquises	Langage de programmation
Interest	Centres d'intérêt, passions et loisirs	Tennis, théâtre
Transcript	Performances académiques	Notes
Affiliation	Organismes dont l'apprenant est membre	IEEE étudiant
Securitykey	données de sécurité	Mots de passe, clés publiques
Relationship	classification et relations entre catégories	qcl / transcript (i.e des notes)

CHAPITRE III

ARCHITECTURE ET MODÈLE COLLABORATIF

Il est communément admis que l'innovation technologique est un puissant facteur de progrès. Après avoir marqué notre manière de communiquer et de faire des affaires, l'Internet influence notre façon d'apprendre. Dans l'apprentissage électronique (e-learning) en sciences et génie, les laboratoires virtuels sont un élément essentiel. Dans ce chapitre, nous proposons une architecture générale de plate-forme pour des laboratoires virtuels distribués. Après avoir présenté globalement cette architecture et décrit ses différentes composantes, nous en retenons deux pour une étude plus approfondie :

- *les cahiers de laboratoire* pour lesquels nous introduisons le concept de répertoire d'objets d'apprentissage utilisant les standards LOM (Learning Object Metadata) et LIP (Learner Information Package) ;
- *le gestionnaire de la collaboration* pour lequel nous proposons un modèle collaboratif multi-agent reposant sur la plate-forme. Ce modèle aborde le problème de la gestion d'une classe d'apprenants géographiquement répartis et scindés en groupes de travaux pratiques. Chaque groupe réalise une expérimentation (i.e un TP), différente ou non de celles des autres et ses membres collaborent entre eux. Aussi, ils peuvent coopérer avec les autres groupes.

3.1 Architecture générale de la plate-forme

L'examen approfondi du modèle conceptuel actuel de la plate-forme nous a conduit à rechercher dans chacune de ses trois couches, des fonctions ou des propriétés susceptibles d'être regroupées dans des modules fonctionnels cohérents. Ces derniers pourraient faire, le moment venu, l'objet de recherches séparées destinées à lever les verrous scientifiques qu'ils constituent. Ceci a débouché sur l'architecture générale présentée à la Figure 3.1.

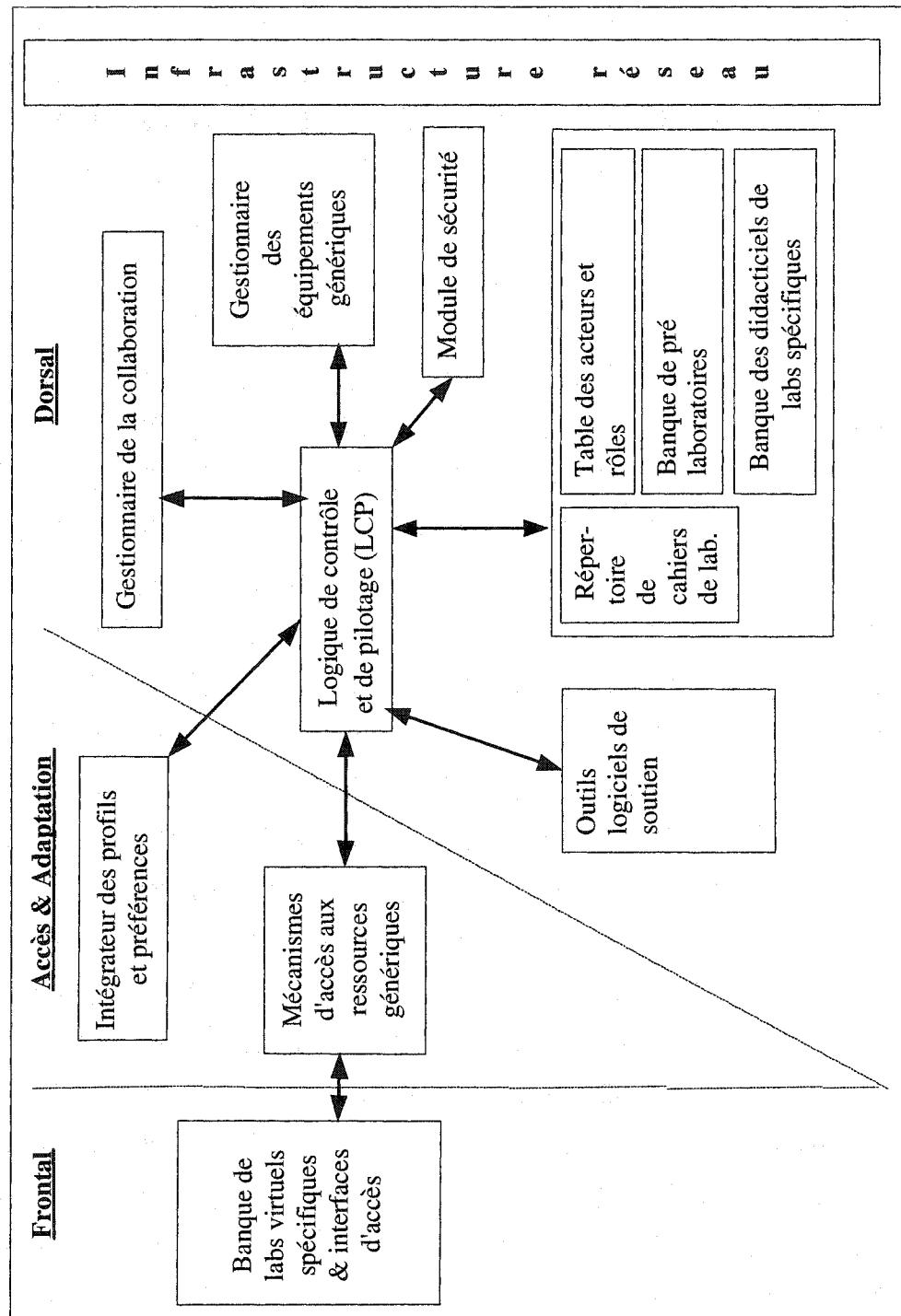


Figure 3.1 Architecture générale de la plate-forme

Pour y parvenir, nous avons dans un premier temps déterminé les composantes principales d'une plate-forme de laboratoires virtuels. Par la suite, nous avons décomposé chacune d'elles en modules plus simples en appliquant les principes suivants :

- chaque module de l'architecture proposée doit avoir un rôle précis ;
- éviter la cohabitation de fonctions très hétéroclites dans un même module ;
- réduire au maximum le degré de couplage entre modules afin qu'on puisse les développer séparément au besoin.

Enfin, nous avons décrit en détail chaque module et analysé les moyens pouvant permettre de l'implémenter.

3.1.1 Banque de laboratoires virtuels spécifiques et interface d'accès

C'est l'ensemble des applications de laboratoires virtuels spécifiques déjà fonctionnelles qui exploitent les ressources génériques, partagées, offertes par la plate-forme. Chacune d'elles dispose d'une interface d'accès (un fureteur par exemple) et répond à un besoin spécifique. Elles peuvent être le résultat de développements séparés, concertés ou non, et provenir de disciplines différentes. Leur point commun est le partage des ressources génériques et spécifiques de la plate-forme. L'objectif à terme est de constituer une banque de laboratoires virtuels spécifiques autour de la même plate-forme. Celle-ci mettra en œuvre un ensemble d'outils et de fonctionnalités génériques et spécifiques.

3.1.2 Mécanismes d'accès aux ressources génériques

Les mécanismes d'accès définissent les règles d'invocation que les développeurs de laboratoires virtuels spécifiques devront suivre pour exploiter les ressources offertes par la plate-forme. Ces mécanismes comprennent essentiellement des protocoles et sont tributaires de l'implémentation, ou au moins de la spécification, des équipements génériques auxquels ils donnent accès. De ce fait, il y a une relation de précédence

temporelle entre leurs réalisations respectives. Pour ce qui est de la démarche, on peut adopter différentes approches:

- négociation – réservation - accès: ce serait une démarche analogue à celle du protocole RSVP (Resource ReSerVation Protocol) qui permet de garantir la qualité de service (QoS) dans la pile de protocoles TCP/IP pour des applications telles que la vidéo sur demande ou plus généralement le multimédia temps réel ;
- envoi d'une requête afin qu'une tâche soit exécutée localement et les résultats rapatriés : c'est la migration de processus ;
- tentative de connexion sans précaution préalable – utilisation pendant un laps de temps - déconnexion: contrairement aux précédentes, cette stratégie est plus sensible à la garantie de la continuité du lien réseau sur les canaux de communication utilisés. Autrement dit, elle peut paraître aléatoire, surtout dans les contextes où on ne peut faire d'autres tentatives. Reprendre ou non l'essai dépendra aussi de la sémantique d'invocation retenue (au plus une fois, au moins une fois, peut-être). Cette sémantique a elle-même des implications sur le choix du protocole de transport. On optera pour le protocole TCP là où la fiabilité est la principale contrainte et pour UDP lorsque c'est la rapidité qui importe et que les informations échangées sont par exemple des petits paquets de commande. En effet, la latence induite par TCP serait pénalisante dans ce dernier cas.

En l'état actuel des connaissances, on a le choix entre deux modèles de programmation: l'invocation distante et la migration de processus. La migration peut prendre la forme d'un ou de plusieurs agents mobiles. Du point de vue des spécifications formelles, un tel protocole doit garantir les propriétés suivantes: la performance des accès, la confidentialité des traitements, l'intégrité des données et la sûreté de fonctionnement. Bref, il s'agit là d'un problème complexe qui nécessite une approche pluridisciplinaire basée sur les méthodes formelles (modélisation et validation), l'automatique (commande et contrôle de processus temps réel), la réseautique, l'intelligence artificielle distribuée voire d'autres disciplines. L'invocation distante peut s'appuyer sur des mécanismes tels que RPC (Remote Procedure Call), Java RMI

(Remote Method Invocation) ou CORBA/IDL (Interface Definition Language). Ces deux dernières techniques permettent de mettre en œuvre une infrastructure d'objets distribués. Cependant, CORBA offre plusieurs avantages tels que la possibilité d'une invocation dynamique (objet dont on ne connaît pas les détails de l'interface avant de l'invoquer) et l'indépendance par rapport à un langage de programmation comparativement à Java RMI. Une autre alternative est DCOM (Distributed Component Object Model), appelé antérieurement "Network OLE", qui est un produit de *Microsoft*. Ce dernier s'appuie sur les spécifications du modèle DCE (Distributed Computing Environment) édictées par l'OSF (Open Software Foundation). Cependant, on peut noter que DCOM a été conçu dès l'origine pour être utilisé avec les autres produits de *Microsoft*. Aussi, on peut combiner plusieurs de ces produits. Par exemple, si CORBA assure l'interopérabilité au niveau middleware, Java procure la transparence d'implémentation puisque le concept de machine virtuelle (JVM) permet de réaliser un programme sans se soucier de la machine cible sur laquelle ce dernier sera installé et exploité.

Les mécanismes d'accès aux équipements distants pour des fins de télémesure/télémanipulation dans le cadre des télélaboratoires et laboratoires virtuels, constituent un champ de recherche à part entière.

3.1.3 L'intégrateur des profils et préférences

Ce module relève des interfaces personne-machine adaptatives, avancées. Il joue un rôle de présentation et d'adaptation pour permettre la prise en compte du contexte particulier du déroulement d'une expérience: calibrage des équipements, taux de précision des mesures, système métrique utilisé, limites de l'interface d'accès utilisateur (par exemple, écran d'un PDA, fureteur pas à jour), etc. C'est ce que montre la Figure 3.2. Notons que, contrairement à ce que pourrait laisser penser cette figure, les laboratoires virtuels spécifiques et l'intégrateur ne communiquent pas directement. Ils le font par l'intermédiaire de la logique de contrôle et de pilotage de la plate-forme. Mais

cette représentation a été retenue par souci de clarté. En effet, mettre la logique de contrôle et de pilotage des deux côtés de la figure ne serait pas très expressif.

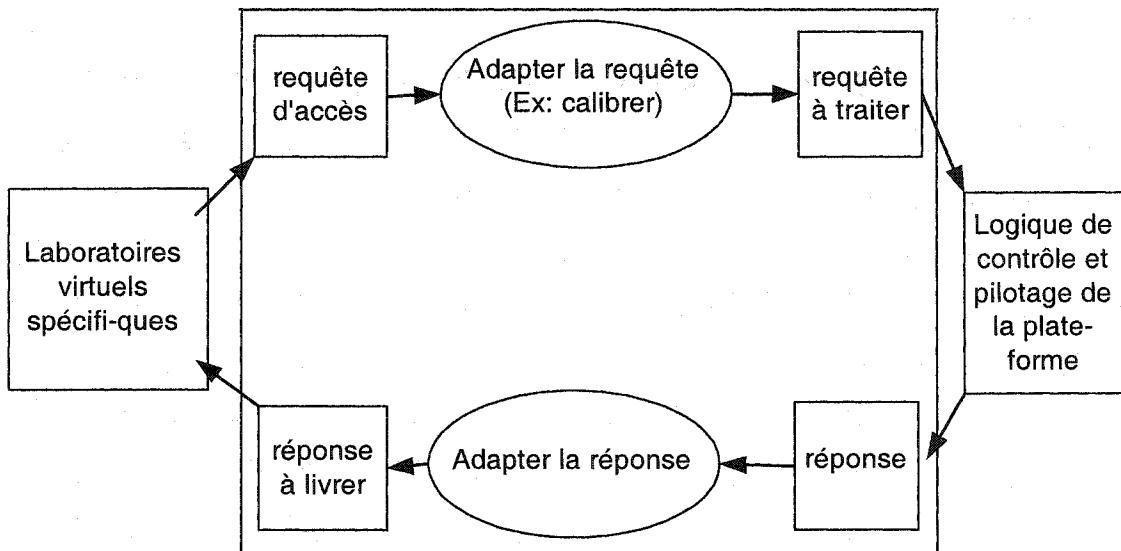


Figure 3.2 L'intégrateur des profils et préférences

Dans sa forme la plus évoluée, l'intégrateur des profils et préférences traite des problèmes liés au multilinguisme ou à l'inter-linguisme, ainsi qu'au type d'utilisateur (personnes âgées, handicapés, enfants). En effet, l'un des bénéfices attendus de la "virtualisation" est l'ouverture de l'enseignement et l'accès du plus grand nombre à la science et aux technologies. Actuellement, beaucoup de handicapés en sont exclus et l'une des barrières découle de leur difficulté d'accès aux équipements des laboratoires conventionnels (Chetz et al., 2002). Certains handicaps seront sans doute difficiles à vaincre mais des efforts peuvent être faits. À titre d'exemple, il existe des interfaces évoluées personne-machine où :

- pour des mal-voyants, un logiciel intercepte les caractères envoyés à l'écran, les convertit en braille et le résultat est affiché à l'intention de l'usager ;
- des handicapés de la main, qui ne peuvent pas se servir de l'ensemble clavier/souris, dictent les tâches dans un sous-langage approprié (non ambigu) et

un logiciel de reconnaissance de la parole génère un fichier de commandes qu'on soumet au système (Chetz et al., 2002).

Ces exemples montrent clairement qu'il faut aller au delà du traditionnel clavier/souris/écran et pousser les recherches dans ce que les anglo-saxons appellent "*assistive technology*". D'ailleurs, plusieurs équipes de recherche, comme celle de la Open University en Angleterre sont déjà sur le créneau (Site #3).

Pour prendre en compte des terminaux clients hétérogènes (PC, laptop, PDA, etc.), la paire XML/XSL offre une bonne solution. Tel que le montre la Figure 3.3, en réponse à une requête (étape 1 de la figure), la logique de contrôle et de pilotage de la plate-forme ne renvoie pas directement une page HTML ou WML (Wireless Markup Language) comme dans les applications Web classiques, mais génère un document XML (étape 2). En effet, HTML et WML enchevêtrent la description du contenu et celle de la présentation. Or, la première est générique et la deuxième dépend du terminal de destination. À l'opposé, XML/XSL dissocie les deux, ce qui est indispensable dans un contexte multi-support. Avec cette dichotomie, pour desservir correctement des terminaux de types différents, il suffit d'avoir un document XML unique (après l'étape 2 de la Figure 3.3) et un formulaire XSL par type de terminal. Le moment venu, la seule chose à faire est de lui associer le bon formulaire XSL, dépendamment du terminal destinataire, pour générer la page HTML ou WML adéquate (étape 4) que le serveur transmettra à l'appelant. Aussi, combiné à des API Java, XML peut faciliter l'interaction et l'échange d'informations avec des bases de données, contribuant ainsi à la gestion de contenu. Cette question relève de l'intergiciel car elle vise à faciliter la communication entre processus applicatifs en exécution. Les paramètres d'adaptation à prendre en compte ici sont de nature statique et liés à l'usager ou à la séance de travaux pratiques. Ils sont connus au moment du login (connexion à la plate-forme).

Cette capacité d'adaptation multiterminale constituera un critère de plus en plus important avec l'émergence progressive de l'apprentissage électronique mobile (m-learning). Aussi, elle tient de la flexibilité qui est indispensable à une diffusion plus large des contenus éducatifs.

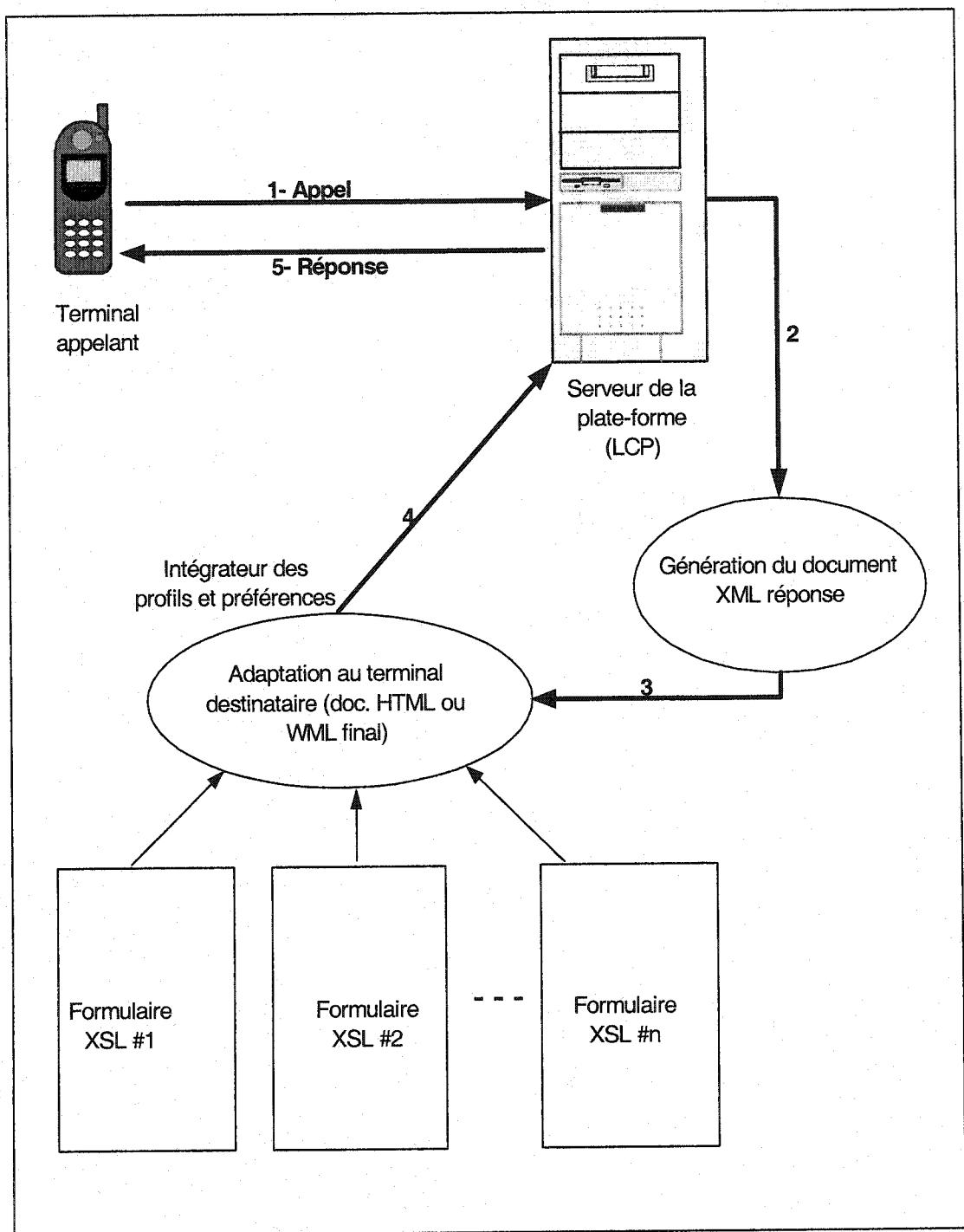


Figure 3.3 Adaptation multi-terminale avec XML/XSL

3.1.4 Logique de contrôle et pilotage de la plate-forme (LCP)

Pour dissocier les moyens qui concourent à la tenue d'une séance de laboratoire virtuel (i.e les ressources génériques), de la manière dont l'ensemble est agencé puis exploité, nous avons isolé la logique de contrôle et de pilotage de la plate-forme en tant que module à part. Ce choix nous paraît fondé dans la mesure où l'architecture relève d'un niveau d'abstraction élevé où les préoccupations sont plutôt d'ordre conceptuel, le critère d'appréciation étant la rationalité des regroupements et la cohérence d'ensemble. Néanmoins, dans une implémentation donnée, on peut décider de distribuer la logique de contrôle et de pilotage à travers l'ensemble des autres modules.

Fonctionnellement, la logique de contrôle et de pilotage:

- gère la relation avec l'apprenant en recevant ses requêtes d'accès aux ressources génériques et en lui livrant plus tard la réponse du système (le reste de la structure est transparent à l'usager) ;
- authentifie et contrôle l'accès aux ressources génériques de la plate-forme ;
- coordonne le partage des ressources en interagissant avec les autres modules (en effet, toutes les communications entre modules, autres que les laboratoires virtuels spécifiques, passent par elle, ce qui lui permet d'avoir un contrôle sur l'ensemble de la plate-forme) ;
- traite les erreurs qui peuvent être de deux types (Rahkila, 2001) : les erreurs de niveau système (exemple : rupture de lien réseau, indisponibilité d'équipement) et les erreurs liées à l'utilisation (exemple : violation de partage).

3.1.5 Outils logiciels de soutien

C'est un ensemble d'outils logiciels spécialisés mis à la disposition de l'usager pour lui permettre d'effectuer plus facilement et en peu de temps certaines tâches qu'on retrouve dans un grand nombre de laboratoires virtuels spécifiques: tracé des courbes, calculs statistiques répétitifs, manipulations d'images 3D, recherche d'informations sur le Web, etc. La gamme peut aller d'une simple calculatrice scientifique à un système expert élaboré, adapté à la résolution d'un problème relativement récurrent (exemple :

analyse de sang dans un laboratoire de médecine). On y retrouve surtout des outils transversaux utilisables dans plusieurs disciplines. Ce module est comparable à une boîte de « logiciels-outils » ou une bibliothèque de moyens opérationnels visant à soutenir l'usager. Ceci dit, aucun d'entre eux ne saurait avoir l'ambition d'être générique et prétendre tout couvrir pour tous les laboratoires virtuels imaginables. L'approche modulaire nous semble plus réaliste et féconde puisqu'elle évite le piège de la générnicité à tout prix, susceptible de conduire à une impasse.

Les fonctions qui nécessitent une capacité de raisonnement, de coopération, de planification et de partage de tâches peuvent correspondre à des agents intelligents, capables de s'adapter aux variations de l'environnement expérimental selon le profil de l'usager et le stade d'avancement du travail pratique. Cette capacité d'adaptation à la réalité est une propriété majeure. D'un point de vue fonctionnel, ces ressources sont placées sous la responsabilité de la logique de contrôle et de pilotage de la plate-forme qui en crée des instances et les affecte aux tâches selon les besoins du moment.

3.1.6 Le gestionnaire de la collaboration

Comme le souligne Himmelman (2002):

« collaborating is a relationship in which each organization wants to help its partners become the best that they can be at what they do ... The qualitative difference between collaborating and cooperating in this definition is the willingness of organizations (or individuals) to enhance each other's capacity for mutual benefit and common purpose ».

En d'autres termes, la collaboration est une relation reposant sur la volonté de chacun d'aider ses partenaires à devenir encore meilleurs dans la poursuite d'un objectif commun. Ici, l'objectif est de réaliser une expérience de laboratoire dans un scénario pédagogique. Le gestionnaire de la collaboration offre des moyens de communication et de coordination entre les différents acteurs impliqués dans la conduite des expériences. Il matérialise le caractère collaboratif de l'activité et peut s'appuyer sur un modèle de collaboration. Les outils de communication qu'on y trouve sont : le courriel, le chat, la

vidéo-conférence, toute l'infrastructure nécessaire à l'exécution des agents, les services de notification d'évènements auxquels les acteurs s'abonnent préalablement, etc.

Même si la quasi-totalité des applications réparties actuelles est asynchrone du fait du partage des temps CPU et des canaux de communication par plusieurs processus en exécution parallèle, l'architecture proposée n'exclut pas le mode synchrone. L'une des principales difficultés de la collaboration dans les laboratoires virtuels vient de la faiblesse de la bande passante des réseaux, ce qui se traduit fréquemment par un recours massif à l'interaction textuelle (courriel, chat). Or, celle-ci est elle-même fondamentalement fastidieuse et mieux adaptée au mode asynchrone. Aussi, tout n'est pas facilement explicable par du texte, sans compter qu'il s'agit de croire ce qu'on lit et non de voir les faits pour s'en convaincre. À terme, la solution serait d'aller vers la visioconférence synchrone et le multimédia temps réel, le texte n'étant qu'un moyen additionnel.

Le gestionnaire de la collaboration joue un rôle essentiel dans la mise en œuvre de l'espace de travail collaboratif qui est l'enceinte virtuelle où a lieu l'expérimentation en équipe. C'est aussi lui qui procure la télé-présence et distingue les laboratoires virtuels dont il est question ici de ceux qui s'appuient sur une simple applet Java pour reproduire un phénomène naturel. En résumé, c'est le ciment qui lie les membres du même groupe de TP et les groupes d'une même classe. Sans cela, la notion de collaboration disparaît, aucun acteur ne sera au courant (awareness) de ce que font les autres apprenants et les laboratoires virtuels ne relèveront plus du « groupware » (logiciel de travail en équipe). La collaboration obéit également à une dimension organisationnelle dans la mesure où on collabore pour faire un certain travail et ce travail est organisé d'une manière donnée. Les deux sont donc liés et interdépendants.

3.1.7 Le gestionnaire des équipements génériques

Le gestionnaire des équipement génériques joue un double rôle :

- implanter les outils et fonctions génériques (équipements de mesure et autres) ;

- collecter les résultats de leur utilisation et les transmettre à la logique de contrôle et de pilotage de la plate-forme.

Réussir la modélisation et la réalisation de cette composante sera l'une des tâches les plus ardues. Quelques unes des questions qui se posent sont :

- a) à quoi correspond exactement un équipement virtuel générique ?
 - La simulation du fonctionnement d'un équipement réel en y intégrant des techniques de réalité virtuelle ?
 - Une interface (instrument virtuel) présentant les résultats d'une télémesure/télémanipulation consécutive à une acquisition des données et accédé via un réseau de télécommunications ?
- b) quels sont les facteurs qui sont dus à la virtualité et qui affectent la qualité des résultats comparativement à la réalité ?
- c) que faire si une panne réseau survient alors qu'on est en pleine interaction avec un équipement distant ? Quel protocole d'arrêt mettre en œuvre ? Comment faire la reprise après panne ? Que faire des résultats partiels obtenus avant la panne ?
- d) comment ajouter un nouvel équipement à la plate-forme (aspect évolutivité) ?

Des deux approches sus-mentionnées, la simulation repose sur les modèles mathématiques et l'informatique graphique. La télémesure/télémanipulation s'appuie sur l'intelligence artificielle distribuée, l'automatique (capteurs / actionneurs, commande de processus, etc.) et la réseautique pour vaincre la distance et offrir une possibilité de téléactivité. Cette deuxième option est confortée par le fait que de plus en plus d'équipements prévoient des ports série ou USB (Universal Serial Bus) permettant de les relier à un ordinateur via une carte d'interface (GPIB par exemple) et de les commander à distance (Saliah, 1999). D'un point de vue pratique, si la réalité virtuelle a fait l'objet de nombreuses études et d'applications fort intéressantes dont les résultats ont été en partie exploités par la première génération de laboratoires virtuels (basés sur la simulation), peu de recherches ont été consacrées à la télémesure/télémanipulation dans

la perspective d'un usage dans l'apprentissage électronique. Le secteur spatial détient une grande expertise en la matière et y a recours depuis de longues années. Des compagnies privées comme *National Instruments* ou *Agilent Technologies* offrent des moyens permettant de connecter leurs équipements de mesure à un ordinateur et d'y accéder via un réseau. Dans un effort concerté, elles ont créé la fondation "Interchangeable Virtual Instruments Foundation" (IVI foundation ; Site #4) dont le but est d'arriver à un standard sur les pilotes des équipements pour favoriser l'interopérabilité et l'hétérogénéité matérielle. De même, le protocole SOAP (Simple Object Access Protocol) permet à des programmes s'exécutant sous des systèmes d'exploitation différents de communiquer en utilisant HTTP et XML (Vaughen-Nichols, 2002). Ainsi, il suffirait à un fureteur d'être "SOAP-compatible" pour être ouvert.

La télémesure/télémanipulation est d'autant plus indispensable que les laboratoires virtuels basés sur la simulation sont extrêmement limités puisqu'ils annihilent carrément toute idée d'interaction, même distante, avec des équipements réels. De ce fait, par leur capacité à effectuer des traitements déportés pour le compte d'un mandataire tout en optimisant l'usage de la bande passante, les agents mobiles peuvent constituer un élément intéressant. En effet, leur mobilité et leur autonomie permettraient d'étendre, avec une bonne tolérance aux fautes, des services de base (ex : une mesure) offerts localement. Cependant, comme toujours le problème majeur avec les agents mobiles reste celui de la sécurité (Moore, 1998).

3.1.8 Le répertoire de cahiers électroniques de laboratoire

Habituellement, un cahier de laboratoire joue deux rôles :

- fournir au départ l'énoncé du TP, le mode opératoire des manipulations et une série de questions;
- contenir les résultats en fin de séance.

Le répertoire de cahiers électroniques que nous proposons est une structure hiérarchique à plusieurs niveaux. Sur la plate-forme, on a un seul répertoire qui contient tous les cahiers des TP qui ont été faits. Il est subdivisé en disciplines. Chaque discipline

est composée de plusieurs domaines et chaque domaine offre différents TP. Un groupe a un cahier pour chaque TP qu'il fait. Le cahier est composé d'un ou de plusieurs paragraphes et chaque paragraphe correspond à un objet d'apprentissage (learning object) décrit par un ensemble de méta-données conformément au standard LOM (learning object metadata) présenté dans au chapitre 2. Ceci conduit à la structure arborescente de la Figure 3.4.

Le répertoire

Il est constitué de plusieurs cahiers de laboratoire. Chaque cahier comporte plusieurs paragraphes et chaque paragraphe est un objet d'apprentissage décrit par un ensemble de méta-données. Ainsi, le répertoire revient en fait à une base de données d'objets d'apprentissage provenant des différents cahiers. Le répertoire implémente l'interface qui permet de manipuler les cahiers (rechercher, créer, modifier, consulter et supprimer).

Le cahier électronique de laboratoire

Il implémente l'interface qui permet de créer, modifier, rechercher ou supprimer les paragraphes. On a un cahier de laboratoire par séance de TP et par groupe. Au moment de la mise en exploitation d'un nouveau laboratoire virtuel spécifique sur la plate-forme, la personne chargée de cette tâche crée simultanément un cahier initial contenant l'énoncé et les directives du TP. Celui-ci apparaît comme un ancêtre dont une copie est automatiquement affectée à chaque groupe lors de son inscription pour une séance dudit laboratoire. Ensuite, les membres du groupe peuvent s'en servir et y inscrire les informations qu'ils partagent. Tous les paragraphes du cahier sont stockés dans une base de données hébergée sur le serveur de la plate-forme. Chaque membre peut ajouter de nouveaux paragraphes au cahier de son groupe ou en récupérer à partir du serveur et modifier voire supprimer (s'il en est l'auteur et si personne d'autre n'est entrain de l'utiliser). Le système de gestion de la base de données (SGBD) gère les accès concurrents et veille à l'intégrité des données. Un apprenant ne peut accéder à un cahier que si celui-ci est la propriété d'un groupe dont il est membre.

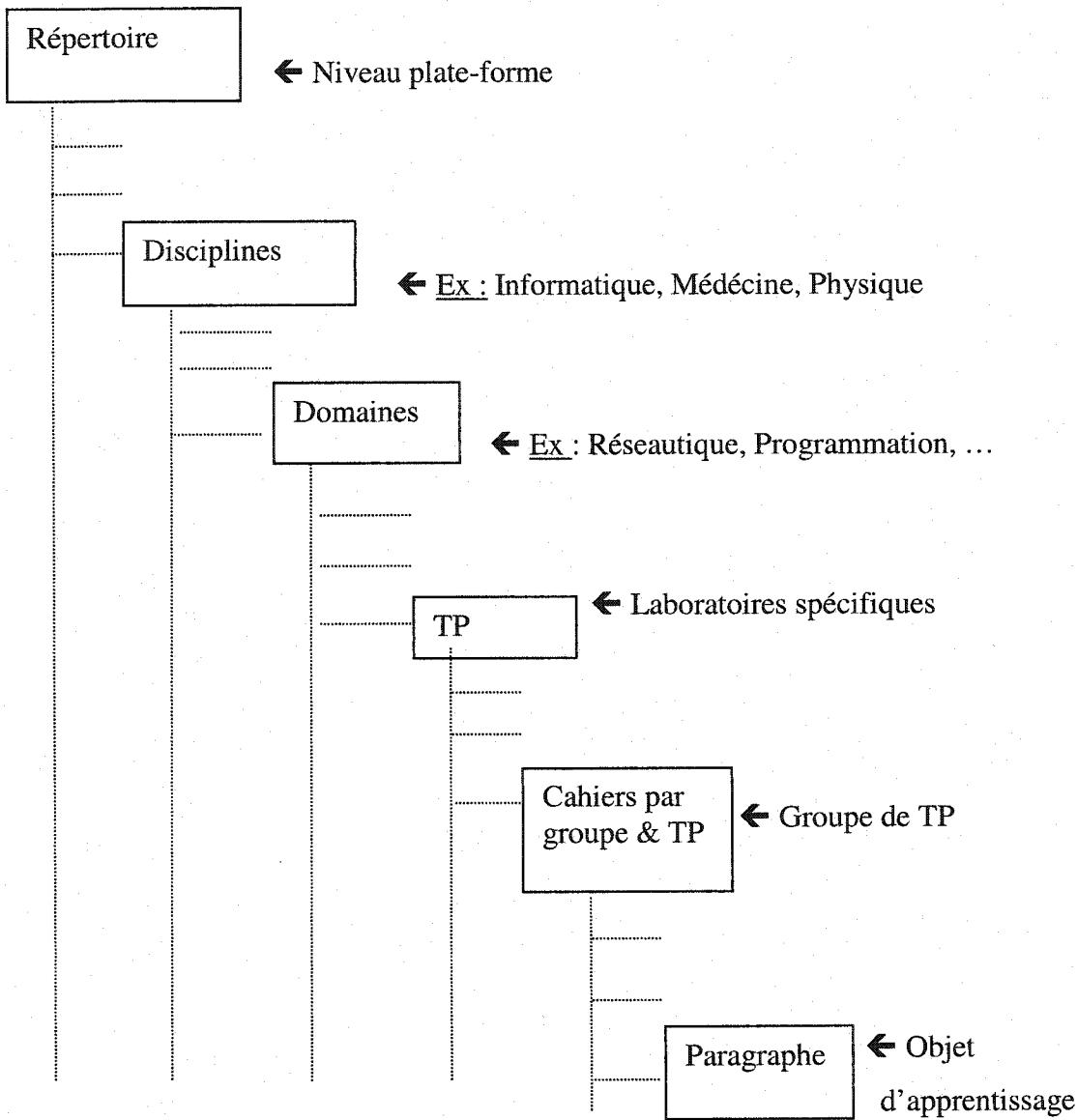


Figure 3.4 Structure arborescente du répertoire de cahiers de laboratoire

Le paragraphe

Il est créé et supprimé par son auteur (i.e le propriétaire). Mais toute personne membre du même groupe peut le lire ou le modifier. Il porte un numéro d'identification unique. D'autre part, chaque paragraphe peut être vu comme un objet (au sens de la programmation Objet) que les membres du groupe "écoutent" puisqu'il leur notifie les modifications qu'il subit. De ce fait, si les apprenants A1 et A2 travaillent sur le

paragraphe P et que A1 effectue une modification, alors P envoie une notification à A2. Ce dernier est alors libre de rafraîchir ou non la copie qu'il est entrain d'utiliser. Un paragraphe apparaît une seule fois dans un cahier (i.e numéro d'identification unique) mais il peut contenir un lien vers un autre paragraphe (par exemple une version plus récente du même paragraphe). Ceci permettra la traçabilité de la séance (backward chaining). Un apprenant crée un paragraphe lorsqu'il veut conserver une information et y accéder en partage avec ses coéquipiers.

Les acteurs et leurs rôles

Les acteurs sont les apprenants, les professeurs, les chargés de laboratoire et l'administrateur de la plate-forme. Le rôle d'un acteur peut éventuellement changer d'un laboratoire virtuel à un autre. Par exemple Joblo est chargé de laboratoire pour le laboratoire numéro 1 mais il est apprenant pour le laboratoire numéro 2. Chaque acteur a un numéro d'identification personnel (NIP) unique dans la plate-forme. Celui-ci sera rattaché à une liste d'autres NIP que l'acteur possède dans d'autres institutions. Par exemple, si un étudiant de l'École Polytechnique suit un cours hors-faculté à l'Université de Montréal, on doit pouvoir relier ses NIP dans les deux institutions, ne serait-ce que pour des besoins de traçabilité. C'est l'acteur qui donne les informations lors de son inscription en remplissant une fiche de renseignements. De manière générale, on peut subdiviser les acteurs en deux catégories : l'encadreur (administrateur de la plate-forme, professeur ou chargé de laboratoire) et les apprenants. L'encadreur dispose d'un privilège d'accès élevé au répertoire. L'apprenant est décrit par un ensemble de métadonnées conformément au standard LIP (Learner Information Package) présenté au chapitre 2.

La sécurité des accès au répertoire

La transmission d'informations sensibles (liées à l'identité, aux notes, etc) via le réseau doit être chiffrée et hachée afin d'en préserver la confidentialité. Une bonne solution reviendrait à utiliser un système asymétrique (clé publique, clé privée). Avant

d'envoyer un message à un acteur, les personnes autorisées utilisent la clé publique de l'acteur pour l'encrypter. À la réception, l'acteur se sert de sa clé privée (qu'il est en principe le seul à connaître) pour décrypter. De plus, avec ce système, on peut garantir la non répudiation. Pour cela, il suffirait que chacun signe ses messages avec sa clé privée et les autres peuvent vérifier cette signature avec la clé publique correspondante. Ainsi donc, l'acteur ne peut se rétracter par la suite s'il posait un acte revêtu de sa signature. Aussi, la catégorie numéro 11 du standard LIP (Securitykey) a été prévue pour cela.

3.1.9 Autres composantes de l'architecture

Les autres composantes de l'architecture sont : la table des acteurs et rôles, la banque de pré-laboratoires, la banque des didacticiels de laboratoires spécifiques, le module de sécurité.

La table des acteurs et leurs rôles

Elle regroupe tous les usagers de la plate-forme, qu'ils soient apprenant ou encadreur (tel que décrit plus). On identifie chacun avec ses priviléges d'accès, ses préférences, ses mots de passe (encryptés), ses clés publiques et privées. Ceci sert surtout à deux choses :

- l'administration de la plate-forme ;
- la prise en compte des particularités de l'usager (langue, système métrique, etc) par l'intégrateur des profils et préférences.

La banque des pré-laboratoires

Elle contient les préalables nécessaires pour bien réussir le laboratoire et en tirer le maximum d'acquis. Cette banque peut contenir des simulations ou procurer un ensemble de liens dont la consultation permettrait de bien comprendre les notions fondamentales essentielles au laboratoire: points précis des notes de cours, entrées d'encyclopédies électroniques spécialisées, des "handbooks", des glossaires de termes techniques, des liens avec d'autres cours dans le cursus, etc. Avec les simulations, elle permet aussi de se mettre en situation avant même que le TP ne commence réellement. C'est le

concepteur du laboratoire virtuel spécifique qui le réalise. Une démarche systématique en la matière et des règles précises doivent être définies et appliquées pour leur rédaction, conformément aux principes du génie logiciel.

La banque des didacticiels de laboratoires spécifiques

C'est le recueil des documentations spécifiques de chaque laboratoire virtuel spécifique. Ceci comprend le manuel utilisateur et le manuel technique. L'objectif visé est double :

- faciliter l'apprentissage et l'utilisation efficiente du laboratoire ;
- s'en servir comme guide de dépannage en cas de problème technique avec l'application concernée.

Comme pour les pré-laboratoires, la rédaction de ces documents doit être normalisée.

Le module de sécurité

Il assure les fonctions de sécurisation de la plate-forme en plus de celles d'accès au répertoire de cahiers électroniques décrit plus haut. Pour réaliser sa fonction, ce module collabore avec la logique de contrôle et de pilotage de la plate-forme qui gère l'accès aux ressources de la plate-forme et traite les erreurs du système tel qu'indiqué plus haut en 3.1.4. Comme on est dans un environnement d'apprentissage, il peut être judicieux de définir différentes politiques de sécurité et de les corrélérer avec des scénarios pédagogiques. Par exemple, la limitation de certaines opérations à des classes de sessions bien définies.

3.2 Modèle collaboratif multi-agent

Le travail collaboratif synchrone est fondamentalement réparti. En effet, les personnes impliquées interagissent par l'intermédiaire de machines reliées à un réseau de télécommunications. Pour nous, ceci revient à scinder une classe en groupes de travaux pratiques, effectuant chacun une tâche identique ou non à celles des autres et à leur offrir un espace collaboratif basé sur la plate-forme pour effectuer des expériences

virtuelles. Ainsi, chaque groupe de TP constitue une équipe virtuelle dont les membres collaborent pour atteindre l'objectif commun qui est de réussir le laboratoire spécifique à réaliser. La nature de la relation (collaboration ou coopération) dépend du degré de couplage entre les actions des personnes concernées. Lorsqu'il est fort et qu'il y a unicité du but, c'est la collaboration; dans le cas contraire, on parle de coopération (Begole, 1998).

3.2.1 Contexte d'utilisation, hypothèses et idées de base

La subdivision d'une classe en groupes de travaux pratiques pour faire des expériences virtuelles présente les caractéristiques suivantes :

- les apprenants ne sont pas nécessairement au même endroit, même s'ils sont dans le même groupe et travaillent ensemble ;
- dans un groupe chacun contribue à la réalisation d'une même tâche; il n'y a pas de divergence d'objectif ou d'indépendance de stratégie entre les membres; l'action est alors concertée ;
- d'un point de vue relationnel, les interactions à l'intérieur d'un groupe diffèrent de celles qu'il y a entre les groupes.

Dans un groupe : Chacun doit savoir (et idéalement voir) à tout moment ce que les autres font. C'est le thème de la télé-présence. De même, on doit pouvoir communiquer et recevoir des réponses aux questions qu'on pose comme dans un laboratoire réel, sans attendre une synchronisation fastidieuse qui prendrait beaucoup de temps. Ceci renvoie à la visioconférence synchrone. Dans notre démarche, nous faisons l'hypothèse que des moyens de communication efficients existent déjà et que la visioconférence synchrone n'est pas vraiment un problème. Cependant, même si on se voit mutuellement, il faut que chacun puisse intervenir de façon à agir sur la tâche menée en commun et dont l'état d'avancement est reflété par l'image du montage expérimental.

Entre les groupes : il s'agit essentiellement de s'entraider, que l'on fasse ou non le

même TP. Ils peuvent communiquer par échange explicite de messages car ces messages sont relativement rares, comparativement à ce qui se passe à l'intérieur d'un groupe. Les délais et la latence sont moins gênants, l'asynchronisme n'est pas tellement un problème et les pertes de paquets ne sont pas si critiques même si le nombre de groupes est potentiellement grand et que l'espace virtuel est à la dimension de l'Internet. Enfin, les groupes partagent des ressources, celles-ci étant offertes par la plate-forme de laboratoires virtuels, dont la logique de contrôle et de pilotage veille à leur bonne utilisation. Pour avancer dans le TP, les usagers modifient le montage expérimental. A titre d'illustration, supposons que ce dernier est un instrument virtuel partagé LabVIEW comme celui de la Figure 3.5.

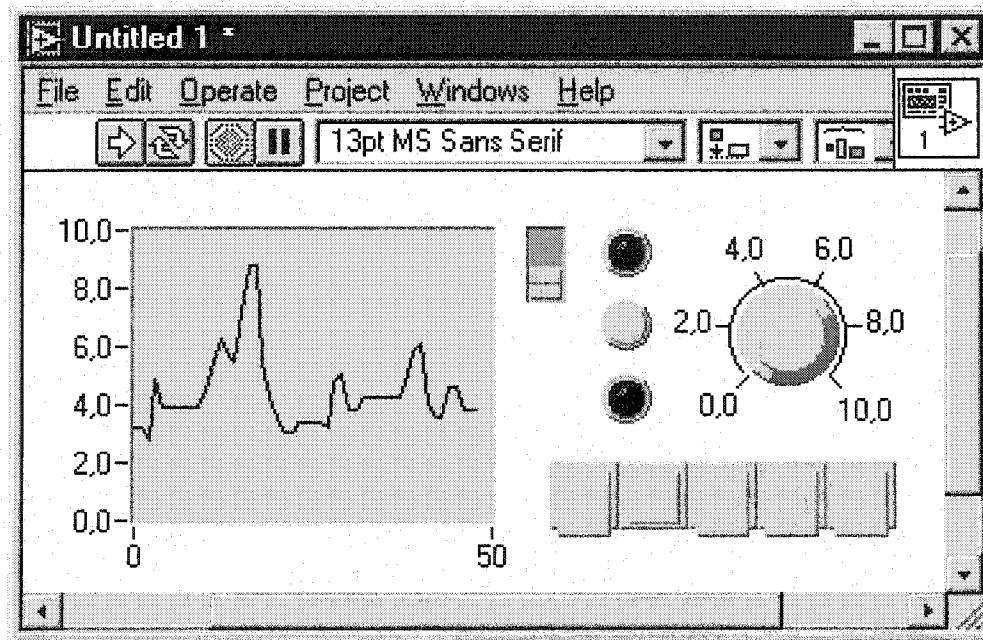


Figure 3.5 face avant d'un instrument virtuel LabVIEW (site #13)

Habituellement, pour piloter l'expérience, l'apprenant manipule les boutons de la face avant de l'instrument à l'aide du clavier et de la souris. Chacune de ces actions provoque l'exécution d'un code, ce qu'un agent peut également faire. Cependant, dans

un contexte de travail en équipe, l'apprenant n'est pas seul et toute modification du montage expérimental requiert :

- *un mécanisme d'ordonnancement, de coordination et d'exécution des mises à jour demandées par les différents membres du groupe ;*
- *une vue fidèle du montage expérimental.* Celle-ci peut être garantie par divers mécanismes comme la visioconférence, la mémoire virtuelle partagée ou AppletVIEW. Ce dernier permet de voir dans un fureteur un instrument virtuel s'exécutant à distance ;
- *des moyens de communication dans chaque groupe et entre les groupes.*

Par ailleurs, nous nous situons dans un scénario de télélaboratoire avec présence simultanée. Nous écartons la simulation puisqu'en réalité, il n'y a pas de collaboration. L'apprenant y est complètement autonome. Précisons aussi que nos groupes sont statiques et connus d'avance puisque les apprenants s'inscrivent, le responsable du laboratoire virtuel (professeur ou chargé de laboratoire) forme les groupes et une plage horaire est réservée pour travailler ensemble, tout en n'étant pas au même endroit.

Dans notre modélisation, nous avons:

- **un agent par équipe virtuelle** qui matérialise la notion abstraite de groupe. C'est lui qui ordonne, coordonne et exécute les actions sur requêtes des membres du groupe. De ce fait, ce sont les membres du groupe qui ont l'initiative de l'action mais c'est l'agent qui en assure la gestion et l'exécution au nom du groupe. L'ordonnancement peut découler d'une stratégie particulière comme un partage de tâches entre les membres ou relever tout simplement du « premier arrivé, premier servi ». Dans ce cas, chaque action sera exécutée suivant son ordre d'arrivée dans le *collecteur local* comme le montre la Figure 3.6. L'exécution d'une action revient pour l'agent à lancer le code qui s'exécute habituellement lorsque l'usager actionne le bouton de la face avant de l'instrument virtuel. Sur le plan structurel, cet agent fait partie du gestionnaire de collaboration. Il prend en charge les interactions qui ont lieu à l'intérieur du groupe. Vis à vis du reste de la plate-forme, il est le représentant du groupe et

agit en son nom. Ainsi, lorsque le groupe a besoin d'une ressource de la plate-forme, l'apprenant qui en prend l'initiative adresse une requête au gestionnaire de groupe. Celui-ci contacte le gestionnaire des ressources partagées et en reçoit plus tard la réponse. Enfin, cet agent maintient une table des usagers actifs contenant uniquement les membres du groupe qui ont démarré le TP. Nous l'appelons le *gestionnaire de groupe (ou team manager)* ;

- **un agent de courtage** (broker agent) qui relaye les messages échangés entre les groupes. Pour cela, il gère un service de répertoire qui est en fait une base de données permettant de localiser chaque groupe. C'est une démarche assez classique en systèmes distribués ;
- **un agent gestionnaire des ressources partagées** qui affecte celles-ci à la demande.

Dans l'architecture générale proposée en 3.1, le gestionnaire de groupe (team manager) et l'agent de courtage font partie du gestionnaire de la collaboration. Le gestionnaire des ressources partagées est une partie intégrante de la logique de contrôle et de pilotage de la plate-forme. Ce dernier dresse parallèlement les statistiques d'utilisation qui peuvent servir à mettre en œuvre des fonctions telles que la facturation lorsque l'accès à certaines ressources s'avère payant. En effet, les équipements de laboratoire coûtent parfois très chers et imposent des investissements importants aux institutions qui en sont propriétaires. Par conséquent, un mode d'organisation complètement décentralisé où chaque agent serait maître de son destin nous semble inapproprié et loin de la réalité. Une certaine centralisation nous paraît inévitable, ne serait-ce que pour des raisons liées au contrôle d'accès et à l'utilisation des ressources. C'est ce que montre le modèle collaboratif illustré à la Figure 3.6.

3.2.2 Description des éléments du modèle collaboratif

Le modèle de collaboration intègre deux niveaux : le niveau local et le niveau central.

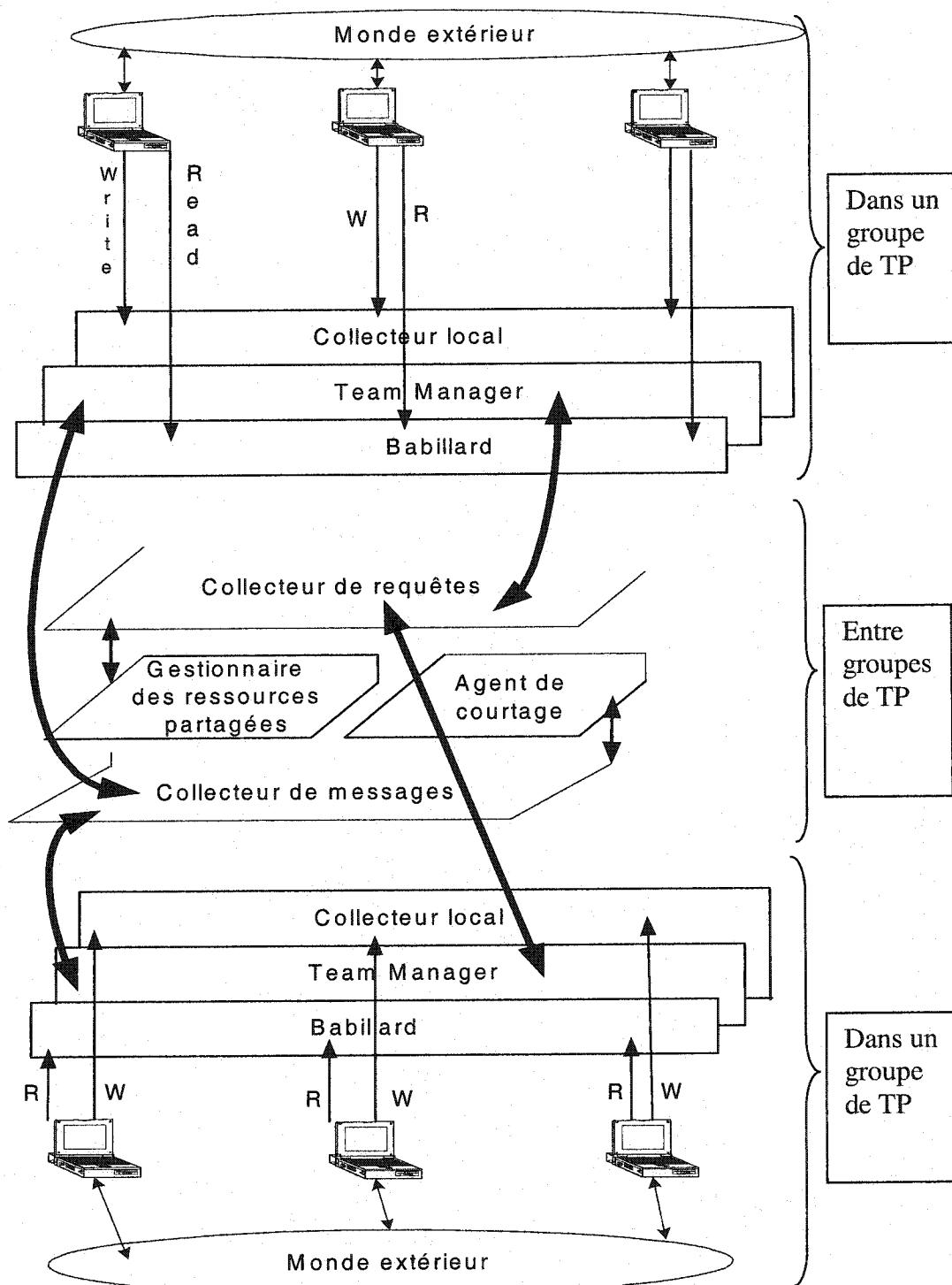


Figure 3.6 Modèle collaboratif multiagent

Le niveau local matérialise l'activité au sein d'un groupe de travaux pratiques dont tous les membres exécutent la même application de laboratoire virtuel spécifique. C'est un espace de collaboration qui comprend: le collecteur local, le babillard et le gestionnaire de groupe (team manager). Le babillard contient l'état actuel du montage expérimental, le collecteur reçoit les requêtes de mise à jour exprimées par les membres du groupe et le gestionnaire ordonne et exécute ces mises à jour. Ainsi donc, les usagers écrivent dans le collecteur local et lisent dans le babillard.

Le niveau central comprend les ressources partagées et supporte les interactions entre les groupes. C'est un espace de coopération composé du collecteur des messages, du collecteur des requêtes d'accès aux ressources, du gestionnaire des ressources génériques partagées et du courtier d'agents.

Le collecteur local: recueille les requêtes des membres du groupe. Celles-ci portent sur la mise à jour du montage expérimental, l'accès aux ressources de la plate-forme ainsi que les messages envoyés à d'autres groupes. Le collecteur local les stocke et le gestionnaire de groupe va les ordonner et les exécuter.

En pratique, on peut réaliser le collecteur local comme :

- Une file d'attente dans laquelle le gestionnaire de groupe lit ce que les membres du groupe écrivent ;
- un groupe de files d'attente (à raison d'une par membre de groupe) que le gestionnaire lit successivement dans un mouvement de tourniquet ;
- un service d'événements (CORBA, Java RMI ou autre) ayant plusieurs fournisseurs (les membres du groupe) et un seul client qui est le gestionnaire de groupe. Chaque fournisseur (publisher) publie des informations et le client (consumer) les lit.

Le babillard: c'est l'espace qui contient le montage expérimental vu par tous les membres du groupe. Chacun d'eux y accède en lecture. Cependant, il ne peut être modifié que par le gestionnaire du groupe. À tout instant, le babillard reflète l'état

d'avancement du TP. Pour la mise en œuvre, il existe fondamentalement trois méthodes: les mécanismes propres à un logiciel comme LabVIEW (VI TCP/IP, VI Client/VI Sever, DataSocket ou AppletVIEW) tel que décrit au chapitre 2, le partage d'application et la duplication. Quant à la vidéoconférence, elle ne fait qu'exploiter l'une de ces approches.

Dans le partage d'application, le résultat d'exécution de l'application est intercepté puis affiché sur différents écrans correspondants chacun à l'un des membres du groupe. Mais, c'est une seule instance de l'application qui est exécutée. Quelques fois, on appelle également cette méthode le partage d'écran (screen sharing). La duplication du babillard dans la mémoire de chacune des machines du groupe peut prendre différentes formes :

- la mémoire virtuelle distribuée (MVD) qui est un logiciel donnant l'impression d'avoir une mémoire commune partagée alors que celle-ci est physiquement répartie sur tous les nœuds du groupe. La cohérence globale est assurée par un protocole de cohérence incorporé dans le logiciel qui implémente la MVD (Mentré, 2001) ;
- l'échange explicite de messages de mise à jour (ex : avec RPC) ;
- les réseaux à capacité d'adressage dans lesquels un nœud peut lire et écrire directement dans la mémoire d'un autre nœud (Cecchet, 2001).

En ce qui concerne l'efficacité, la duplication contrôlée est généralement moins gourmande en bande passante, même si la cohérence globale est à maintenir. À l'inverse, le partage d'application assure une cohérence parfaite mais exige un réseau rapide. Or les laboratoires virtuels sont destinés à être exploités sur Internet. Pour allier ces exigences contradictoires (cohérence forte versus économie de la bande passante), on peut garantir une forme relâchée de WYSIWYS (what you see is what i see) et opter pour la réPLICATION partielle (Begole, 1998)

Le gestionnaire de groupe: c'est la partie du gestionnaire de la collaboration qui gère les interactions à l'intérieur d'un groupe. Fonctionnellement, c'est autour de lui que le groupe est organisé. À ce titre, il :

- lit le collecteur local et met à jour le babillard ;
- maintient la coordination entre les actions des membres du groupe ;
- assure la communication avec le reste de la plate-forme aux yeux duquel il matérialise le groupe et en est le représentant.

Le gestionnaire de groupe est un agent proactif qui lit le contenu du collecteur local, l'analyse et agit en conséquence. En effet, si l'information lue dans le collecteur local est une requête de mise à jour, il modifie le contenu du babillard dont il est le seul écrivain. Si c'est un message destiné à un autre groupe, il l'envoie à l'agent de courtage qui en localisera le destinataire et effectuera la remise. Si par contre, il s'agit d'une requête d'accès à une ressource de la plate-forme, il l'envoie au gestionnaire des ressources. A toute fin pratique, les messages échangés entre agents ont la structure syntaxique (*type*, *origine*, *destination*, *données*) où *type* indique la nature du message, *origine* l'expéditeur, *destination* le destinataire et *données* l'information à traiter. Ceci est détaillé à la Figure 3.7. On peut créer le gestionnaire sur la première machine du groupe qui démarre le laboratoire. Une fois lancé, il initialise le collecteur et le babillard locaux.

La communication entre les niveaux local et central pose plusieurs problèmes d'intergiciel (middleware) inter-agent (Tadié, 1998): langage de communication (ontologie accessible et compréhensible par tous), protocole de communication (mécanisme de prise de parole prévenant la famine), structure des messages (syntaxe d'un message valide), etc. Pour ce qui est de l'architecture de communication, c'est à dire le schéma d'échange de messages entre agents, nous avons choisi l'étoile puisque toutes les communications entre groupes transitent par l'agent de courtage. Du point de vue des fonctionnalités, le niveau central joue un double rôle : gérer les ressources génériques partagées et relayer les communications inter-groupes.

Le collecteur des messages : son rôle est analogue à celui du collecteur local, mais au lieu des requêtes émises par les apprenants, il reçoit les messages échangés entre les groupes. Il peut être implanté sous la forme d'un service d'événements ayant comme

client (consumer) le courtier d'agents et comme fournisseurs (publisher) les gestionnaires des différents groupes.

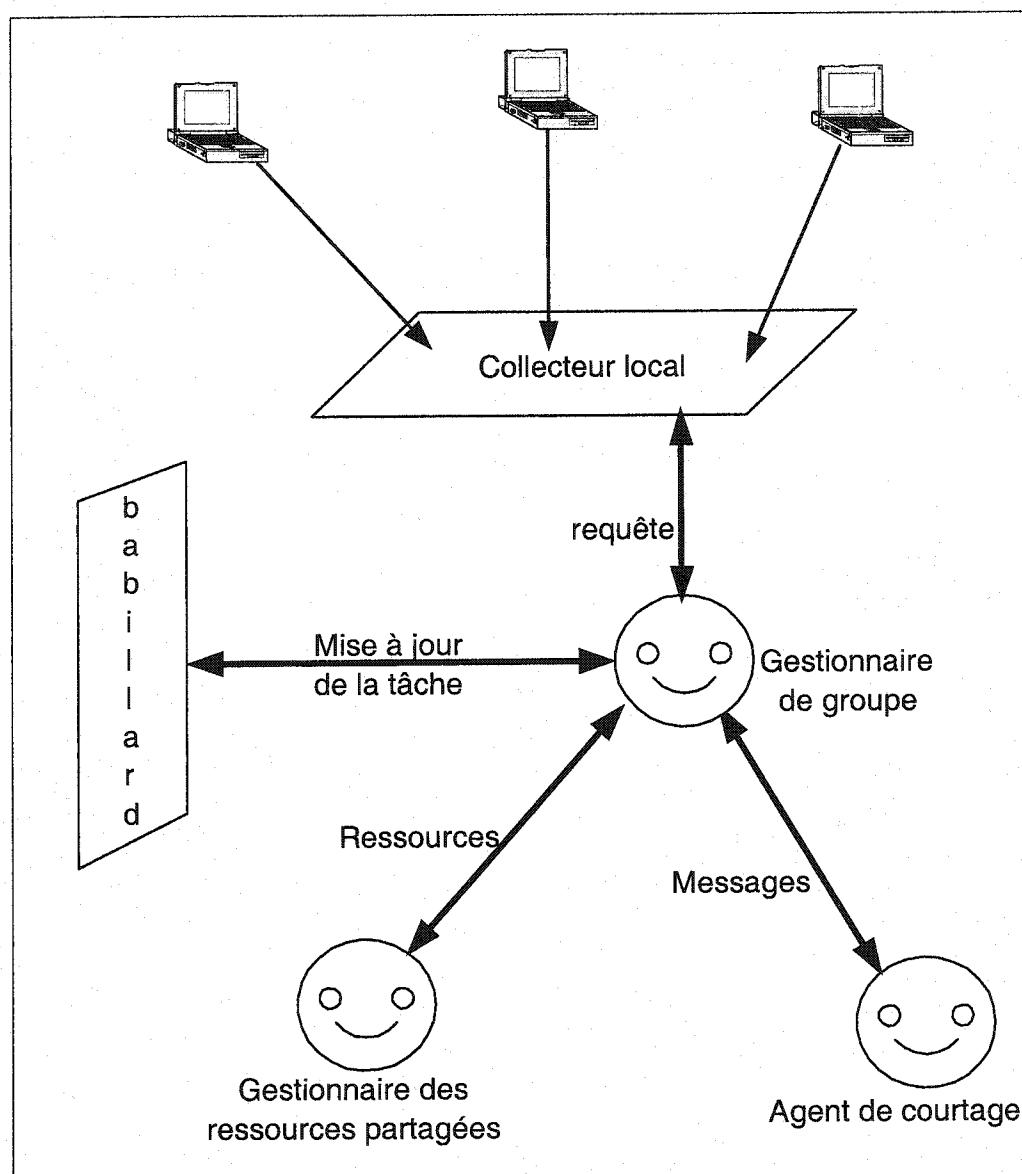


Figure 3.7 Interactions entre les différents agents

Le collecteur des requêtes : sa fonction diffère légèrement de celle du collecteur des messages car il reçoit les requêtes d'accès aux ressources de la plate-forme. Celles-ci sont exprimées par les différents gestionnaires de groupe. Ce collecteur peut être réalisé

Tableau 3.1 Le niveau local du modèle collaboratif

Élément	Fonction	Solutions d'implémentation
Collecteur local	Reçoit les requêtes des membres du groupe. Il est lu par le gestionnaire de groupe	File(s) d'attente – Service d'évènements (publisher / consumer)
Babillard	Procure une vue fidèle du montage expérimental. Il est mis à jour par le gestionnaire de groupe et lu par chaque membre du groupe	AppletVIEW, DataSocket, Vidéoconférence, mémoire virtuelle distribuée
Gestionnaire de groupe	Ordonnance, coordonne et exécute les requêtes des membres du groupe	Agent proactif

Tableau 3.2 Le niveau central du modèle collaboratif

Élément	Fonction	Solutions d'implémentation
Collecteur des messages	Reçoit les messages échangés entre groupes	Service d'évènements (publisher / consumer)
Collecteur des requêtes	Reçoit les requêtes d'accès aux ressources de la plate-forme	Service d'évènements (publisher / consumer)
Gestionnaire des ressources partagées	Partie de la logique de contrôle et de pilotage de la plate-forme qui donne accès aux ressources	Agent intégré dans la logique de contrôle et de pilotage
Agent de courtage (broker)	Relaye les messages échangés entre groupes	Agent s'appuyant sur LDAP (Lightweight Directory Protocol)

également comme un service d'évènements ayant comme fournisseurs les gestionnaires de groupe et comme lecteur unique le gestionnaire des ressources partagées.

Enfin, les différents constituants du modèle collaboratif sont résumés par les Tableaux 3.1 et 3.2

3.2.3 Initialisation et arrêt

Le niveau central (collecteur de messages, collecteur de requêtes d'accès aux ressources, gestionnaire des ressources partagées) est hébergé sur le serveur de la plate-forme. Ses éléments constitutifs fonctionnent sans arrêt. Les agents qui en font partie correspondent à des processus qui s'exécutent dans une boucle infinie. Dans chaque groupe, le gestionnaire de groupe est lancé sur la première machine qui démarre l'application de laboratoire virtuelle spécifique. Puis, ce dernier s'inscrit immédiatement au niveau de la logique de contrôle et de pilotage de la plate-forme. L'agent de courtage qui en fait partie met à jour son service de répertoire et note que le groupe en question est désormais actif. Ensuite, le gestionnaire de groupe initialise le collecteur et le babillard locaux puis note dans sa table locale d'usagers actifs son premier apprenant (en fait, celui sur le poste duquel il s'exécute). Par la suite, dès qu'un autre membre du groupe rejoint l'application, il est ajouté à la table.

Lorsqu'un usager quitte le groupe, il en informe lui-même son gestionnaire de groupe. Sinon, il est facile de vérifier périodiquement que tous les membres du groupe sont effectivement en ligne. Ceci ne génère pas un gros trafic réseau puisque, dans la réalité, ne serait-ce que pour des raisons pédagogiques, les groupes sont de petite taille (rarement plus de 3 ou 4 personnes). Si un groupe quitte l'application, son gestionnaire en informe l'agent de courtage et le service de répertoire est automatiquement mis à jour. Par contre, si la première machine du groupe à démarrer le TP n'est pas la dernière à s'arrêter, l'agent gestionnaire du groupe qui s'y trouve doit migrer sur l'une des machines restantes. Mais cette situation est rare car en pratique, un apprenant peut certes être en retard pour le début du TP, mais une fois qu'il a rejoint ses coéquipiers, il reste généralement jusqu'à la fin. On peut donc dire que tous les membres d'un groupe

quittent l'application en même temps et il n'y pas nécessité d'envisager une migration de l'agent.

En appliquant ces règles, il on peut programmer des évènements tels que :

- avertir si tous les groupes commencent leurs TP ;
- avertir si tout le monde a fini;
- prévenir si la majorité des groupes démarrent ;
- alerter si tel usager se déconnecte pendant plus de x minute.

Ceci peut aider à appliquer des politiques spécifiques pour mieux gérer les ressources partagées mais également télé-surveiller la classe et contrer entre autre la "e-école buissonnière".

CHAPITRE IV

IMPLÉMENTATION, MISE EN ŒUVRE ET RÉSULTATS

Après avoir proposé une architecture générale de plate-forme supportant des laboratoires virtuels répartis, nous en avons amplement étudié deux composantes: le répertoire de cahiers électroniques et le gestionnaire de la collaboration. Dans ce chapitre, nous nous attelons à l'implémentation et à la mise en œuvre du premier sous la forme d'une base de données d'objets d'apprentissage. Parallèlement, les travaux de standardisation effectués sur les méta-données ont décrit de manière précise ce qu'est un apprenant (learner information package) ou un contenu d'enseignement (learning object), mais il y a eu peu d'implémentations concrètes. Autrement dit, à ce jour on sait ce qu'il faut faire parce qu'on a bien défini le "QUOI" mais peu d'études ont été consacrées au "COMMENT" (Bassiliades, 2002). Dans les sections suivantes, nous expliquons pourquoi nous avons opté pour l'implémentation du répertoire avant de choisir nos outils de travail (analyse et langage de programmation). Puis, nous présentons les détails de l'analyse et de la conception. Nous terminons, par la mise en œuvre et l'analyse de nos résultats d'expérimentation.

4.1 Justification du choix du répertoire de cahiers électroniques

Lorsqu'on fait des expériences de laboratoire dans un environnement réparti, l'un des défis majeurs consiste à savoir comment collecter, communiquer, partager et gérer des données hétérogènes (résultats intermédiaires, analyses, images, observations, etc.) manipulées concomitamment par des participants géographiquement dispersés mais œuvrant en équipes virtuelles. Le cahier électronique permet de collecter les informations enregistrées par chacun pour les mettre à la disposition de l'ensemble des membres de l'équipe de travail. Selon la définition de la *Collaborative Electronic Notebook Systems Association* (CENSA) (Site #11):

« *An electronic notebook is a system to create, store, retrieve and share fully electronic records in ways that meet all legal, regulatory, technical and scientific*

requirements ... By "system" we mean the right combination of policies, procedures, technologies, and regulations».

Cet organisme (i.e CENSA) distingue deux types de cahiers :

- **le cahier personnel** (Personal Electronic Notebook System ou PENS) destiné normalement au travailleur indépendant (avocat, médecin, gens d'affaires). Il remplace le bloc-note traditionnel et peut intégrer des fonctionnalités évoluées telles que la recherche d'informations sur le Web ;
- **le cahier collaboratif** (Collaborative Electronic Notebook System ou CENS) axé sur la connectivité, la communication et le travail d'équipe. Pour cela, il s'appuie sur le cahier personnel afin de soutenir et de favoriser la collaboration et le partage d'informations au sein d'un groupe de travail. Dans les laboratoires virtuels, c'est cette catégorie de cahiers qui nous intéresse. En en collectionnant un certain nombre, on constitue un répertoire.

On peut donc dire que le cahier électronique de laboratoire vise la production et la publication de données durant les séances de laboratoires virtuels. Rich Lysakowski note que dans des compagnies comme *DuPont* ou *Air Products*, les travailleurs consacrent 15 à 25% du temps passé dans les laboratoires à remplir manuellement des documents en papier alors même que ceux-ci ne peuvent être partagés instantanément (Lysakowski, 1997). Il note aussi que ce sont souvent des personnes qualifiées et l'informatisation permettra de rationaliser davantage leurs tâches. Aussi, la notion de répertoire (repository) revêt une importance particulière dans les environnements d'apprentissage en ligne puisque c'est le lieu informatique qui héberge les contenus utilisés pour l'enseignement. D'où l'importance de ce thème.

4.2 Choix d'outils pour l'implémentation

En matière d'analyse, les méthodes structurées et fonctionnelles se sont historiquement imposées les premières. Inspirées par la structure interne des ordinateurs, elles consacrent la séparation des données et des traitements tout comme données et code le sont dans la mémoire physique des ordinateurs. Aussi, cette dichotomie reflète

l'habitude que les informaticiens ont prise de raisonner en terme de fonctionnalités du système (Muller et al., 2000). Dans les années 80, la prolifération des méthodes orientées objet qui intègrent la structure et le comportement fait naître un besoin d'unification. D'où la naissance de Unified Modeling Language (UML). Celui-ci est un langage de modélisation standard qui définit des diagrammes structuraux et comportementaux pour représenter à la fois la statique et la dynamique du système étudié. Son formalisme graphique est simple et expressif. Son approche globale (tous les aspects du système), son universalité et le fait qu'il soit un standard, nous amène à le choisir comme outils d'analyse.

Pour ce qui est du langage de programmation, Java présente plusieurs caractéristiques intéressantes:

- il est bien adapté à la programmation d'applications réparties, destinées à une exploitation sur le Web dont il est devenu le langage par excellence ;
- la transition presque naturelle d'une analyse UML à une programmation en Java;
- la possibilité d'avoir des objets qui notifient les événements alors que d'autres les écoutent. Ceci peut être utilisé lors de la mise à jour des paragraphes d'un cahier électronique. Les membres du groupe de TP sont des objets qui écoutent ces événements de mise à jour. Cela est d'autant plus important que les apprenants sont géographiquement répartis ;
- la disponibilité de moyens de chiffrement des données sensibles (identités des personnes, appréciations relatives à l'évaluation d'un travail) avant de les envoyer sur le réseau. À ce sujet, notons que la sécurité est un facteur important dans les environnements d'apprentissage en ligne, notamment dans les télé-laboratoires ;
- la portabilité des programmes Java (compile once, run anywhere). En effet, la compilation génère un code (le bytecode) destiné à une machine virtuelle (i.e qui n'a pas forcément une existence réelle) mais qui sera transformé à l'exécution en un code compréhensible par le microprocesseur de la machine cible ;

- la possibilité de paralléliser les traitements (multithreading) pour minimiser les temps de réponse.

Pour toutes ces raisons, nous adoptons Java comme langage de programmation.

4.3 Détails de l'analyse et de la conception

Dans cette section, nous présentons les différents diagrammes issus de l'analyse et de la conception du répertoire de cahiers électroniques à base d'objets d'apprentissage, tel qu'il a été décrit au chapitre 3. Rappelons que nous utilisons les standards de métadonnées LOM pour les contenus et LIP pour les apprenants. Ainsi, nous décrirons : les diagrammes des cas d'utilisation, les paquetages, les diagrammes des classes et le diagramme d'activités.

4.3.1 Diagrammes des cas d'utilisation

Un cas d'utilisation est un ensemble d'actions exécutées par le système en réponse à l'action d'un usager appelé encore acteur. L'acteur est une entité externe qui agit sur le système. Chaque cas d'utilisation représente une manière spécifique d'interagir avec le système. Mis ensemble, les cas d'utilisation centrent l'expression des exigences du système sur ses utilisateurs. Pour construire le diagramme correspondant, classons les usagers par catégories et voyons pour chacune de ces catégories les traitements effectués et les informations échangées.

Catégories d'usagers

Dans notre système, nous nous intéressons à deux types d'usagers: l'apprenant et l'encadreur.

- **L'apprenant:** il s'authentifie au système et peut soit consulter les cahiers auxquels il a le droit d'accéder, soit les mettre à jour en y ajoutant, en y modifiant ou en y supprimant des paragraphes. Pour s'authentifier, il doit fournir un nom d'utilisateur et un mot de passe valides. Pour modifier ou créer un nouveau paragraphe dans un cahier, il doit entrer les données puis saisir les métadonnées

y afférentes. Dès qu'un cahier est mis à jour, une notification est envoyée à toutes les personnes qui sont entrain de l'utiliser à cet instant précis. L'usager ne peut accéder qu'aux cahiers des groupes dont il est membre. Dans un cahier, il ne peut supprimer que des paragraphes dont il est l'auteur. Les problèmes liés aux accès concurrents et à la cohérence des données sont gérés directement par le système de gestion de base de données (SGBD) que nous utilisons pour stocker nos objets d'apprentissage. Pour nos tests, nous nous servons de *MicroSoft Access*.

- **L'encadreur**: c'est le professeur, le chargé de laboratoire ou toute autre personne qui joue le rôle d'administrateur de la plate-forme. Celui-ci crée, supprime ou modifie des laboratoires virtuels spécifiques, des cahiers, des groupes et des séances de travaux pratiques (TP) après s'être authentifié. C'est également lui qui inscrit les groupes dans les différentes séances. Pour cela, l'encadreur utilise le serveur de la plate-forme. La création d'un nouveau laboratoire virtuel spécifique s'accompagne immédiatement de celle du cahier originel qui contient l'énoncé du TP, les questions auxquelles les apprenants doivent répondre ainsi que le mode opératoire des manipulations à faire. De manière analogue, l'inscription d'un groupe dans une séance s'accompagne automatiquement d'une copie du cahier originel qui est affectée au groupe en question. Dès lors, tout membre du groupe qui démarre l'application peut s'en servir.

En faisant abstraction des autres acteurs de la plate-forme (par exemple l'expert de contenus), nous en déduisons les deux cas d'utilisation des Figures 4.1 et 4.2.

4.3.2 Diagrammes des classes

Selon le formalisme UML, les diagrammes des classes décrivent la statique du système à travers les classes d'objets. Chacun de ces diagrammes modélise un aspect particulier. Concrètement parlant, nous subdivisons le système en quatre sous-systèmes correspondants à quatre paquetages et nous faisons un zoom sur chacun d'eux. Puis, nous reconstruisons l'ensemble en les intégrant. Cette démarche nous conduit à trois

diagrammes de classes: le premier montre une vue d'ensemble du système, le second décrit l'apprenant et le troisième porte sur le contenu du cahier électronique vu comme une collection d'objets d'apprentissage stockés dans un répertoire (repository).

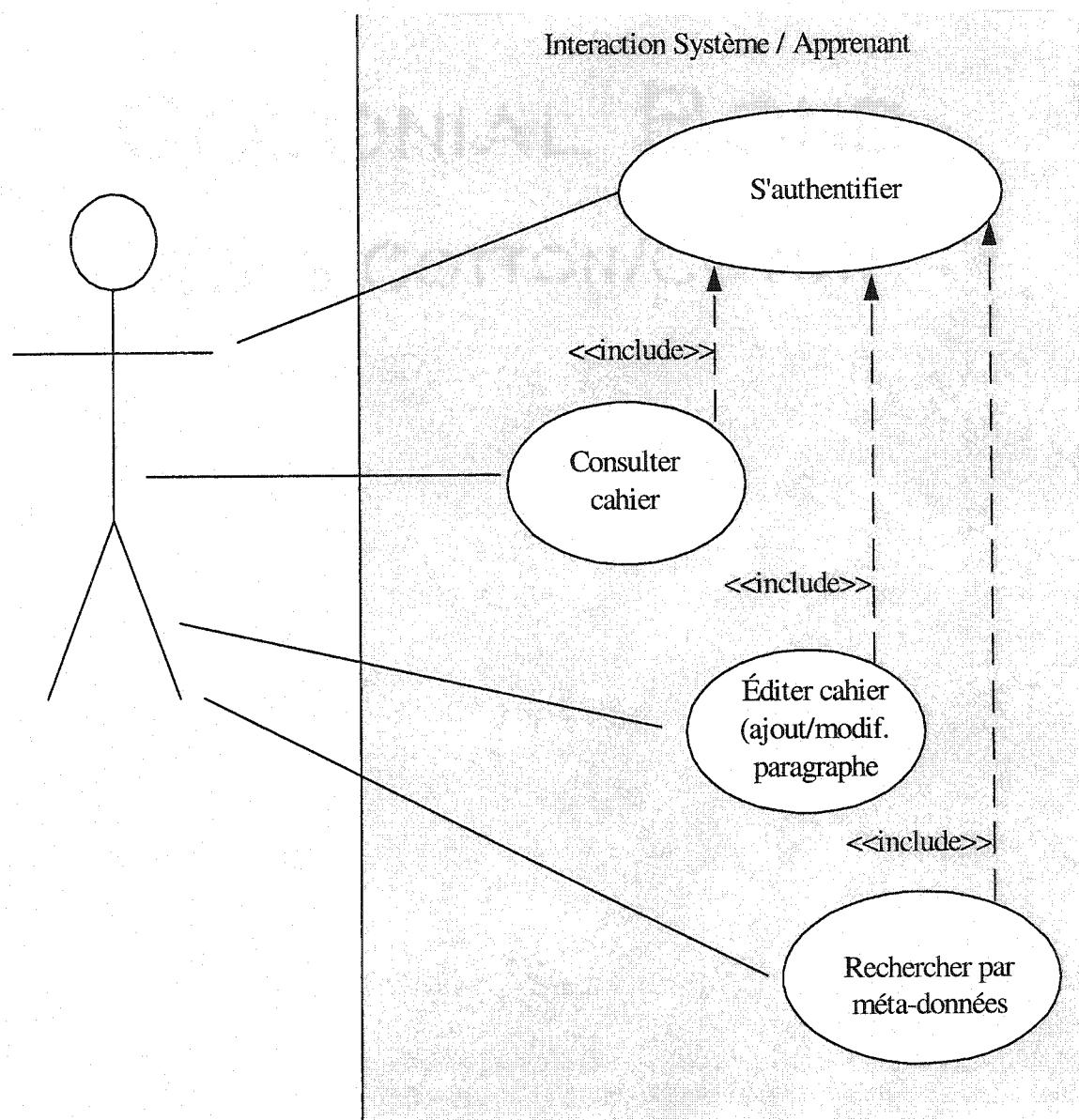


Figure 4.1 Interactions de l'apprenant avec le système

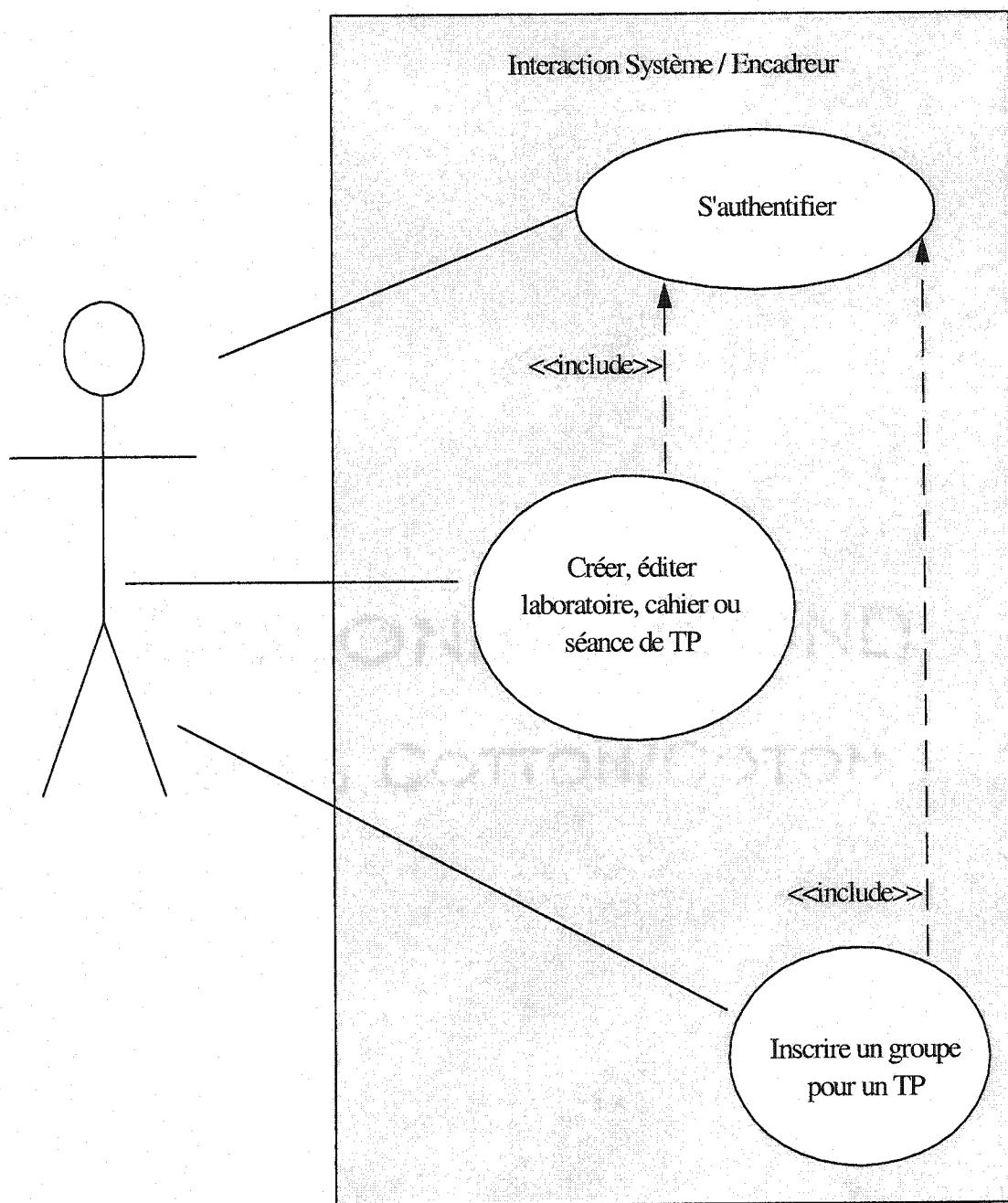


Figure 4.2 Interactions de l'encadreur avec le système

Pour décrire l'apprenant, nous utilisons le "learner information package" proposé par IMS. Il est largement inspiré du « Public And Private Information for Learners » (PAPI Learner) de IEEE qui lui est antérieur. Nous avons utilisé la version décrite dans le

document « IMS Learner Information Package : Information Model Specification – Final Specification Version 1.0» de Mars 2001, téléchargeable gratuitement sur le site de IMS (Site #9). Par souci de simplicité et de clarté, nous n'en avons gardé que trois des onze catégories, à savoir "identification", "affiliation" et "SecurityKey" qui sont les seules à intéresser notre propos.

Pour le contenu, il existe actuellement plusieurs propositions de standards de métadonnées (LOM, SCORM, Dublin Core, ARIADNE). Le passage de l'un à l'autre peut s'effectuer grâce aux tables de conversion qui ont été définies par les organismes promoteurs. Chaque développeur d'application est libre de choisir le schéma qui lui convient le mieux. Chez les militaires, le choix va généralement à SCORM (Sharable Content Object Reference Model) qui est issu d'un projet du DoD (Department of Defense des États Unis). En génie, on connaît mieux LOM (learning object metadata) qui est un standard proposé par IEEE. Nous en utilisons la version définie dans le document «Draft Standard for Learning Object Metadata » P1484.12.1 de Mars 2002. Il est librement accessible sur le site de IEEE (Site #12).

Paquetages du système

En procédant de manière itérative, nous faisons une première décomposition du système en quatre sous-systèmes tel que le montre la Figure 4.3. Puis, après les avoir examiné en détail, nous les réintégrons afin de reconstituer l'ensemble. Ces sous-systèmes sont :

REPOSITORY : contient les éléments qui constituent la charpente du répertoire, c'est-à-dire les disciplines et les domaines dans lesquels sont classés les différents TP.

LABORATORY : ce paquetage regroupe les laboratoires virtuels et leurs séances. Pour participer aux séances de travail dans un laboratoire virtuel, les apprenants sont organisés en groupes de TP. Pour chaque séance, chaque groupe utilise un cahier différent.

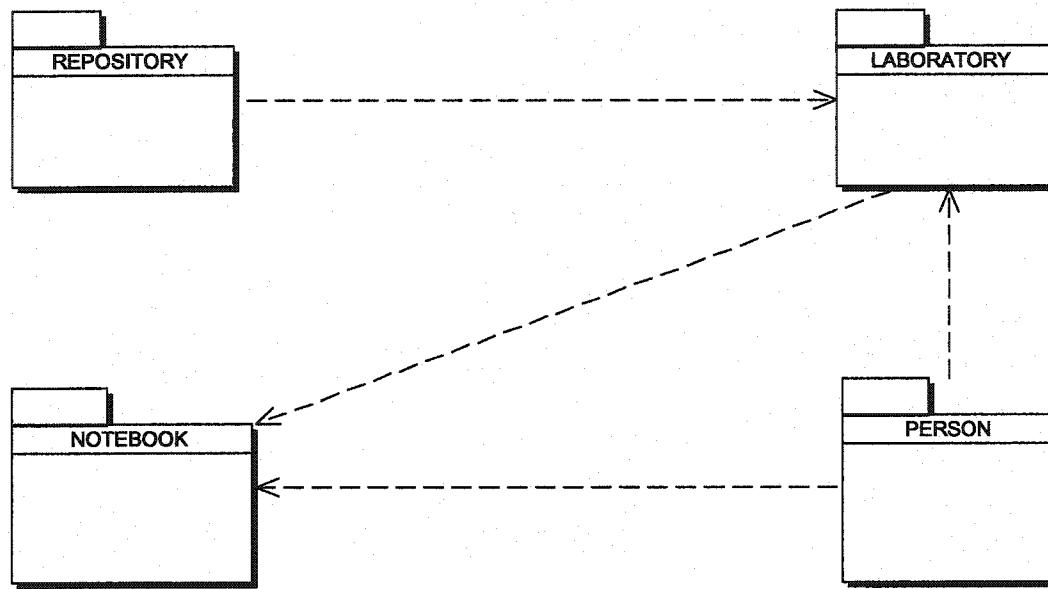


Figure 4.3 Diagramme des paquetages du système

NOTEBOOK : contient les cahiers électroniques avec leurs paragraphes et les métadonnées qui les décrivent.

PERSON : regroupe toutes les personnes qui interviennent dans le système. Nous préférons le terme PERSON à celui de ACTEUR utilisé jusque là car, dans le vocabulaire UML, ce dernier a une signification précise et peut désigner autre chose qu'une personne physique.

Mais chacun de ses sous-systèmes est lui-même complexe, ce qui requiert donc d'autres décompositions. Cette démarche progressive nous conduit au diagramme de classe de la Figure 4.4.

La classe *Personne* est une super-classe qui se spécialise en *Encadreur* et en apprenant. Chaque apprenant équivaut à l'objet composite *LIP* qui correspond à son tour à toutes les métadonnées descriptives présentées à la Figure 4.5.

La classe *paragraphe* est la composition d'un *Contenu* (texte, image ou son) et d'un objet composite *LOM* qui représente les métadonnées décrivant ce paragraphe.

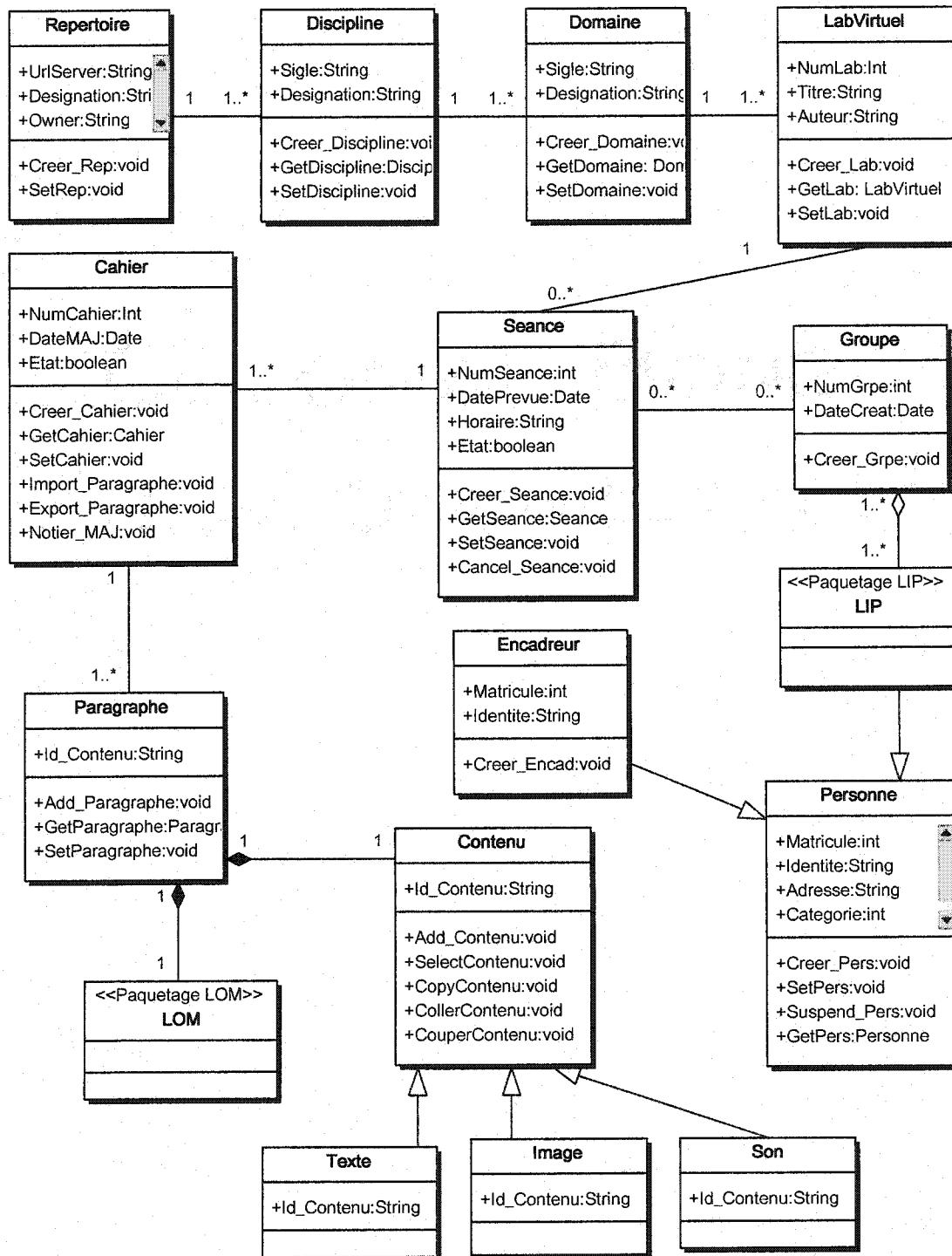


Figure 4.4 Diagramme des classes du répertoire

De cette manière, le paragraphe est un objet d'apprentissage. La Figure 4.6 montre le diagramme des classes du LOM.

La classe *Séance* correspond aux séances de laboratoire virtuel. Elle résulte d'une relation entre *LabVirtuel*, *Groupe* et *Cahier* (relation ternaire). De ce fait, un groupe peut répéter plusieurs fois le même TP mais avec des cahiers différents et donc dans des séances différentes.

Dans les diagrammes de classes du LIP (Figure 4.5) et du LOM (Figure 4.6), nous utilisons l'attribut "Champ", sans l'avoir défini explicitement. Celui-ci exprime l'extensibilité prônée par ces deux standards, car il permet d'ajouter de nouveaux attributs et de nouveaux objets. Aussi, IMS soutient qu'il peut également servir pour des besoins spécifiques d'implémentation qui seraient autrement impossibles à réaliser (IMS, 2001). Néanmoins, nous l'omettrons à l'implémentation car, en pratique, il est difficile de traiter une donnée non complètement définie mais prévue pour une utilisation future (for future use).

Pour les cardinalités (ou multiplicités) dans chacun des deux diagrammes, nous nous sommes appuyés sur les standards. Ainsi, pour le LIP, toutes les classes représentées à la Figure 4.5 possèdent soit directement elles-mêmes des attributs dont la présence est obligatoire soit elles ont des classes filles pour lesquelles certains attributs sont obligatoires. D'où des cardinalités minimales de "un" partout. À l'inverse, le LOM précise que toutes les catégories sont facultatives, ce qui implique des cardinalités minimales de zéro du côté des classes filles dans chacune des associations.

4.3.3 Diagrammes d'activités

Dans le formalisme UML, les diagrammes d'activités matérialisent les activités et les transitions, ce qui montre l'ensemble des traitements effectués dans le système étudié. Par souci de concision, ici nous n'aborderons que le diagramme d'activités relatif au premier cas d'utilisation (Figure 4.1), puisque l'essentiel des traitements est effectué par l'apprenant. Celui-ci est illustré à la Figure 4.7.

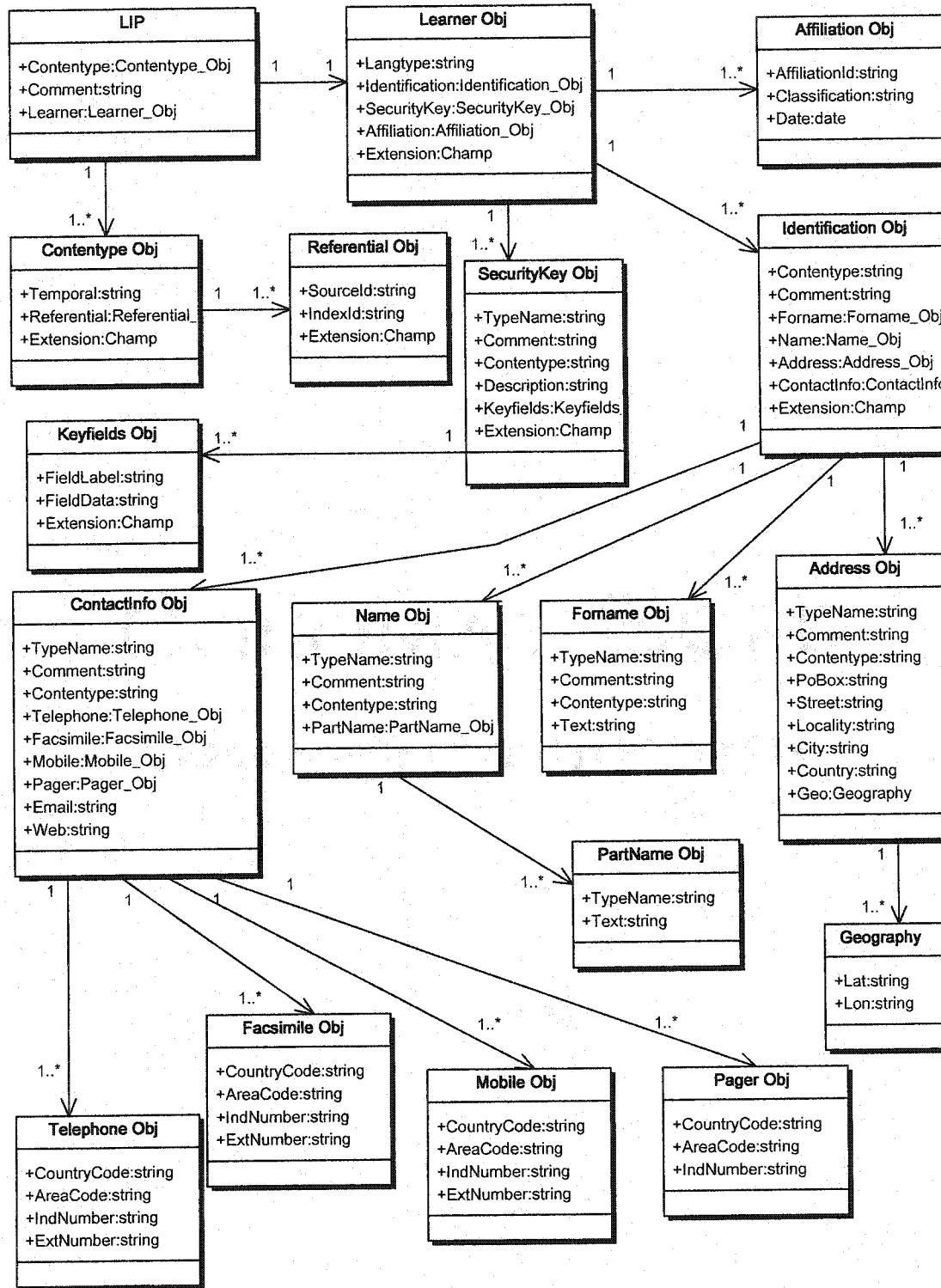


Figure 4.5 Diagramme des classes LIP

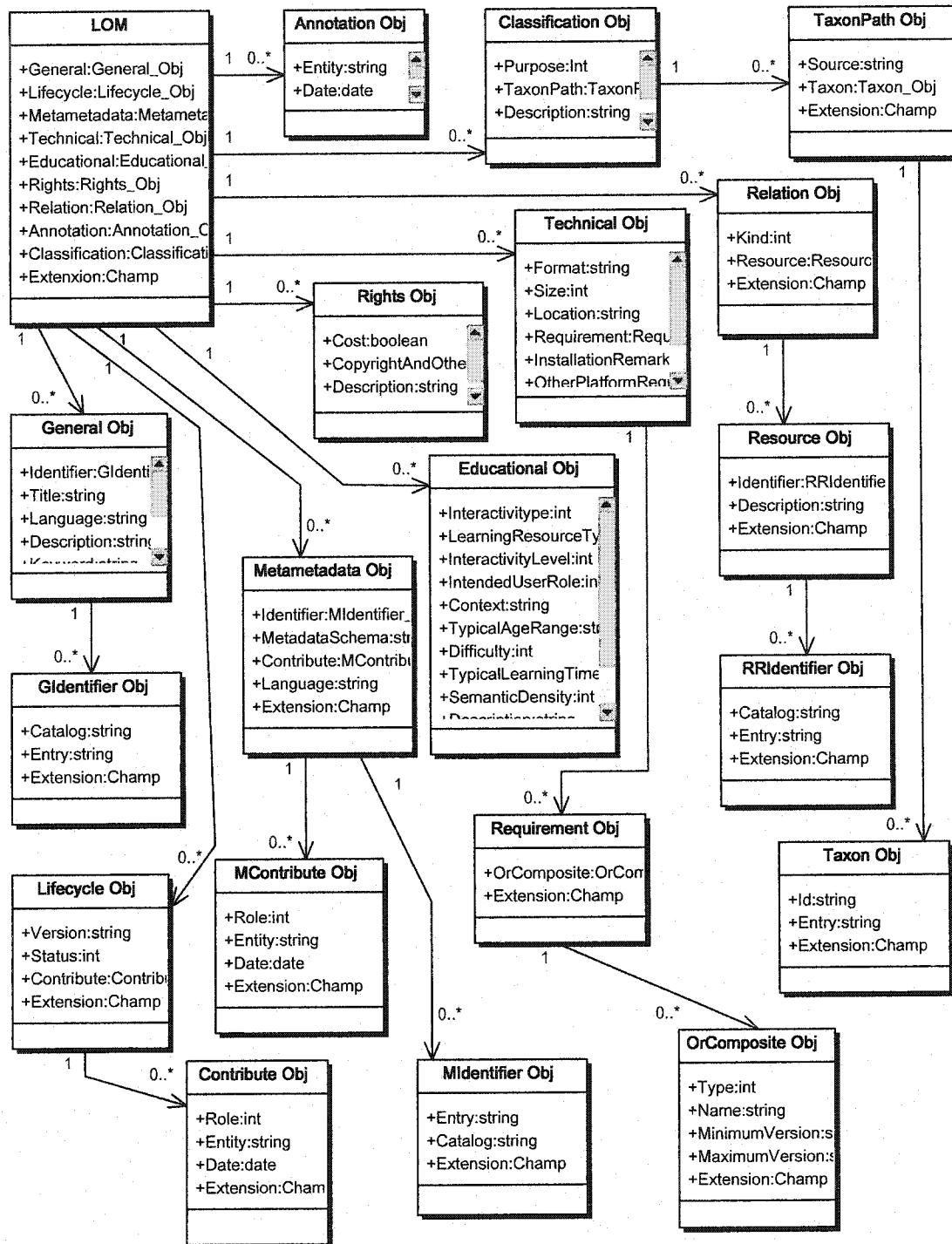


Figure 4.6 Diagramme des classes du LOM

L'encadreur n'intervient que de temps à autre pour créer de nouveaux laboratoires virtuels spécifiques, constituer des groupes de TP, enregistrer des apprenants, ouvrir des séances de laboratoire et d'y inscrire les groupes participants. Ces tâches s'effectuent du côté serveur. Elles permettent d'administrer le système et de contrôler l'utilisation des ressources. À ce titre, elles sont bien sûr importantes mais ne constituent pas un grand volume de traitement.

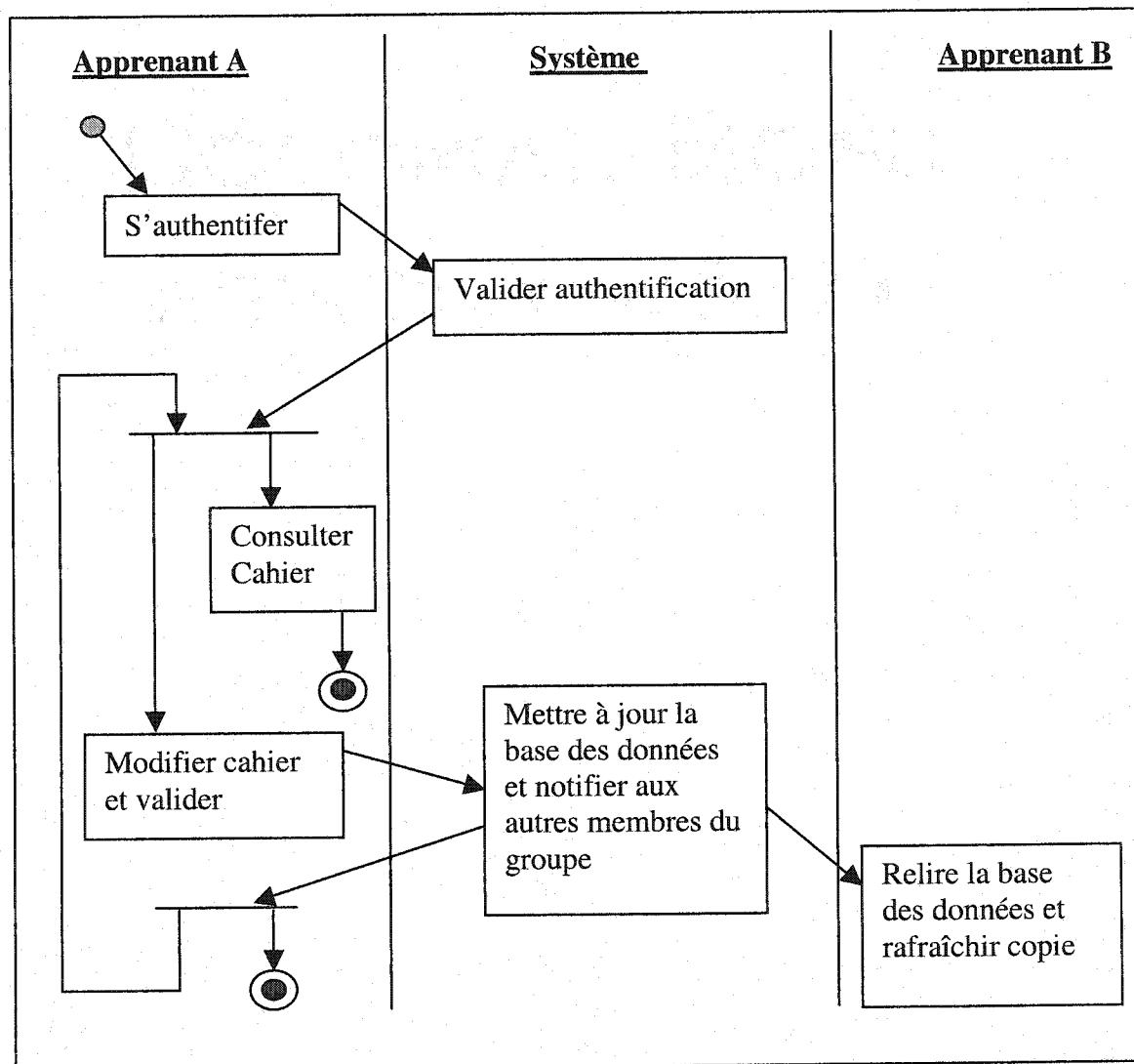


Figure 4.7 Diagramme d'activités pour les interactions Apprenant / Système

4.4 Mise en œuvre

Dans cette section, nous présentons la démarche que nous avons suivie pour mettre en œuvre le répertoire de cahiers électroniques à base d'objets d'apprentissage. Aussi, nous donnons quelques illustrations d'interfaces utilisées pour saisir, modifier ou consulter des données.

4.4.1 Connexion au répertoire

L'architecture est client/serveur comme dans la plupart des applications Internet actuelles. Dans un premier temps, l'apprenant télécharge l'éditeur de paragraphes à partir du serveur de la plate-forme de laboratoires virtuels et l'installe sur son poste. Puis, il tente de se connecter au serveur. Alors, il doit s'identifier en indiquant son nom d'utilisateur, son mot de passe, sa langue de travail et le schéma de métadonnées utilisé (LOM, DCMI, SCORM), comme le montre la Figure 4.8.

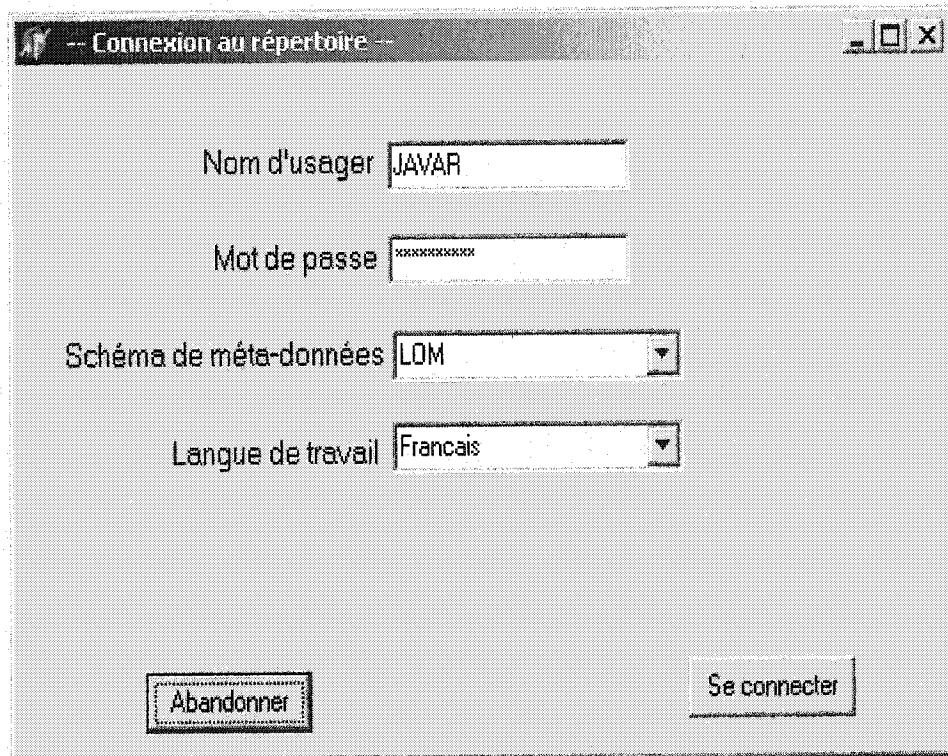


Figure 4.8 Connexion au répertoire

L'apprenant indique sa langue de travail car, dans une session donnée, cette information peut être utilisée pour faire des recherches plus efficaces basées sur l'exploitation des méta-données. Ainsi, si un document existe en plusieurs langues, on lui retournera, celle qui correspond à la sienne. Actuellement, cette fonctionnalité n'est pas implémentée mais c'est une extension possible.

4.4.2 L'éditeur de paragraphe

Si la procédure d'authentification réussit, une requête SQL est générée afin de retrouver l'ensemble des cahiers auxquels cet apprenant peut accéder. Pour cela, nous utilisons un servlet. La réponse est retournée dans une page JSP (Java Server Page).

Lorsque l'apprenant clique sur l'un des cahiers, l'éditeur de paragraphes est lancé localement sur son poste et une deuxième requête SQL est envoyée au serveur pour retrouver l'ensemble des paragraphes constituant ce cahier. Le résultat est affiché dans l'éditeur de paragraphes qui apparaît à la Figure 4.9. L'éditeur de paragraphes comprend quatre parties: le menu (2^{ème} ligne de l'écran), la structure arborescente du cahier (en haut à droite), l'espace de notification d'évènements (en haut à gauche) et l'éditeur proprement dit (partie inférieure). Le paragraphe sélectionné par l'apprenant s'affiche dans la partie basse. Le menu Fichier permet d'enregistrer les données dans la base de données, d'importer des paragraphes ou de les exporter vers d'autres logiciels (traitement de texte ou tableur par exemple).

Pour chaque modification de paragraphe ou pour toute insertion de nouveau paragraphe, dès que l'apprenant valide, la base de données est mise à jour et une notification est envoyée aux autres membres du groupe de TP qui sont connectés au même cahier afin de les en informer. Alors, ils sont libres soit de rafraîchir les copies qu'ils détiennent, soit d'ignorer cette notification. Le menu Métadonnées permet de créer, de modifier ou de consulter les méta-données associées au paragraphe ouvert dans l'éditeur. Enfin, le menu Impression permet d'imprimer, le paragraphe ouvert, le cahier dont il relève, les méta-données du paragraphe ou celles du cahier. En cas d'ajout de paragraphe, l'interface de saisie des méta-données (Figure 4.10) s'affiche

automatiquement. Par contre, si l'apprenant modifie un paragraphe, il lui revient de mettre à jour explicitement les méta-données. Cela se fait dans le menu Méta-données / Modifier.

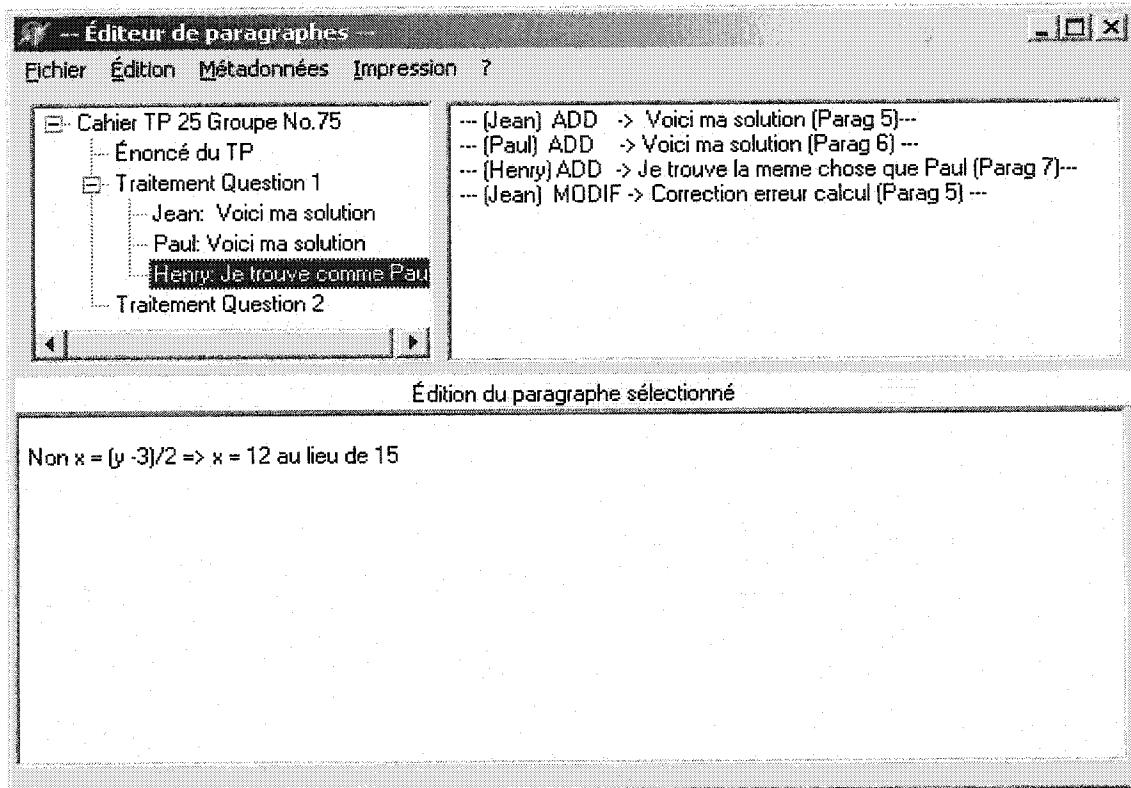


Figure 4.9 L'éditeur des paragraphes

4.4.3 Méta-données

Dans la Figure 4.10, les bandes horizontales (cycle de vie, Métaméta-données, Technique) délimitent les différentes catégories du LOM. Les boutons de la partie inférieure (Éducatives, Droits, Relations, Annotations, Classification) permettent d'accéder aux autres catégories. Nous ne les présentons pas ici par souci de concision. Sachant que chaque catégorie peut avoir plusieurs instances, celles-ci sont stockées dans une petite fenêtre à droite. Les boutons Add, Remove et Delete servent à gérer cette liste. Chaque méta-donnée (ou champ de saisie) se rapporte à sa catégorie et non directement au paragraphe. C'est pour cette raison que le même nom peut apparaître

dans plusieurs catégories avec des significations différentes à chaque fois. Exemple : *Contributeur* apparaît dans *Cycle de vie* et dans *méta méta-données*. Dans le premier cas, il décrit ceux qui ont contribué à la réalisation de l'objet d'apprentissage (le paragraphe) et dans le deuxième, il s'applique aux méta-données elles-mêmes et non directement au paragraphe.

The screenshot displays a window titled "Méta-données décrivant un paragraphe (Learning Object)". The window is divided into four main sections:

- 1 - Généralités:** Contains fields for Identifiant (TP25Gr75-11), Langue (fr-CA), Catalogue (LVEST), Description (fr, racine équation à 2 inconnues), Mots clés (racine, substitution), Thème (Virtual Lab), and buttons for Add >>, << Remove, and Delete.
- 2 - Cycle de vie:** Contains fields for Version (1), Entité (Jean, Paul, Henry), Status (Draft), Date (10 Nov 2002), Contributeur (Jean VAROT), and Rôles (Auteur). It also shows a "Cycle de vie" section with Auteur: Jean VAROT. Buttons for Add >>, << Remove, and Delete are present.
- 3 - Méta méta-données:** Contains fields for Identifiant (TP75Gr75-21), Entité (Jean, Didier), Catalogue (LVEST), Date (08 Nov 2002), Standard (LOMv1.0), Contributeur (Didier NOLA), Langue (fr-CA), and Rôle (Validateur). It also shows a "Métaméta-données" section with 7525-30 Jean [Auteur] and 7525-31 Didier [Validateur]. Buttons for Add >>, << Remove, and Delete are present.
- 4 - Technique:** Contains fields for Format (texte), Taille (200 Octets), Localisation (www.Larin.polyml.ca), Réquis (-), Condition_OR (-), Type (Éditeur de texte), Nom (OS: windows), Version Min (win 9x ou +), Version Max (Windows NT), Install remarks (unzip before install), Plate-forme (fr, carte son), Durée (-), and Technique (windows, linux Redhat, linux Mandrake). Buttons for Add >>, << Remove, and Delete are present.

At the bottom of the window are buttons for Éducatives, Droits, Relations, Annotations, Classification, and Fermer (Close).

Figure 4.10 Méta-données associées à un paragraphe

Côté serveur, l'encadreur crée les apprenants, les laboratoires virtuels, les groupes et les séances. De même, il inscrit les groupes dans les séances de TP. L'écran principal du serveur est présentée à la Figure 4.11.

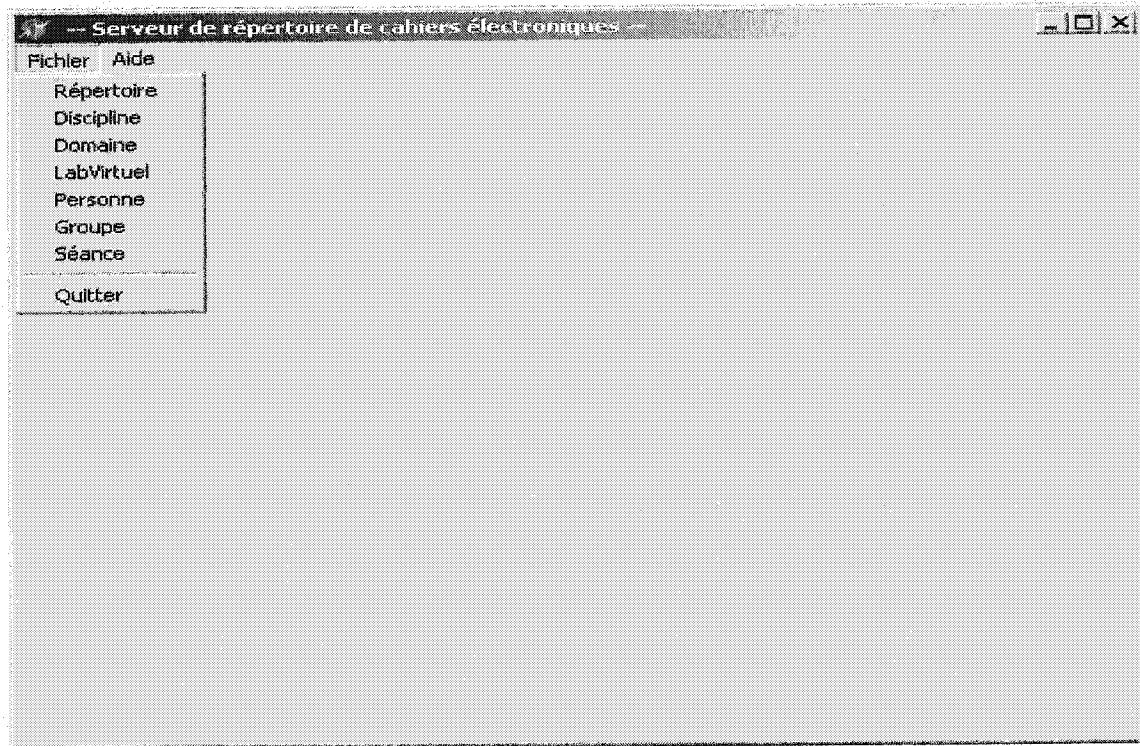


Figure 4.11 Écran principal du serveur

Pour créer un acteur, on se sert de l'interface présentée à la Figure 4.12. La partie supérieure (au dessus de la barre affiliation) sert à indiquer, en plus de l'adresse et de l'identité principales (valeurs utilisées par défaut), le matricule, le nom d'utilisateur, le mot de passe, le type d'acteur (apprenant, professeur, chargé de laboratoire ou administrateur), l'état (actif ou non) ainsi que le dernier paragraphe qu'il a créé.

La partie "Affiliation" correspond à la catégorie de même nom dans le standard LIP. Elle permet d'entrer toutes les organisations auxquelles l'acteur adhère ainsi que son statut et sa date d'entrée dans chacune d'elles. Comme les méta-données constituent également un pas vers le Web sémantique, ce sont là de bons critères de recherche. Un acteur pouvant s'affilier à diverses organisations, le bouton *Add* permet d'en dresser la liste (à droite). *Remove* permet de retirer un élément de la liste et de le modifier, alors que *delete* sert à la suppression. Le bouton "*Fermer*" permet de quitter cette fiche.

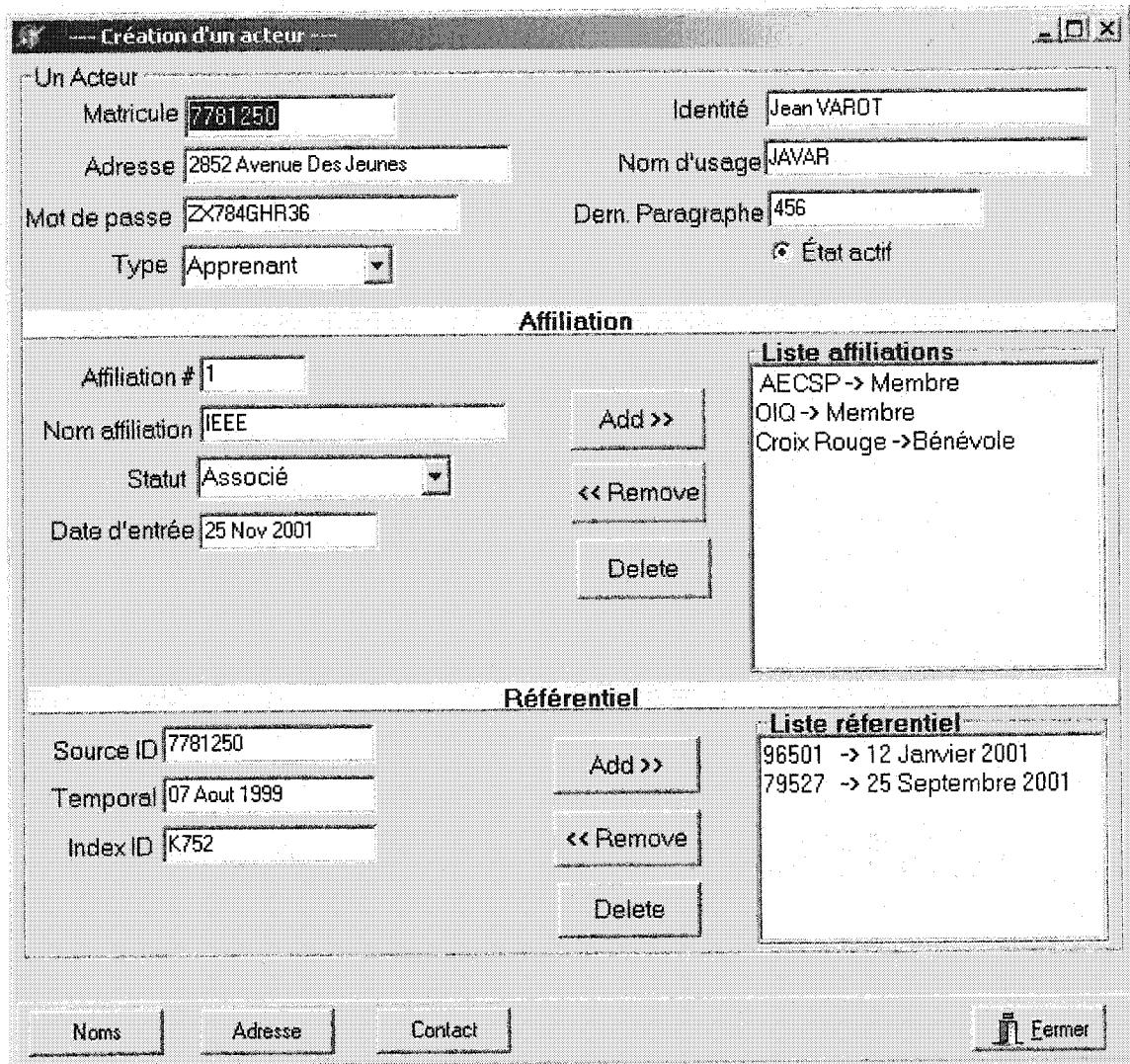


Figure 4.12 Crédit d'un apprenant

La partie "Référentiel" concerne l'identification de notre répertoire. Dans la logique du LIP, chaque fournisseur d'objets d'apprentissage (et donc son répertoire) doit être discerné. C'est la même chose qu'avec les opérateurs télécoms (les carriers) et les banques qui ont chacun un identifiant permettant de les reconnaître. L'idée qui est derrière cette démarche est celle de l'identifiant universel de l'usager. Cet identifiant peut bien être la concaténation de plusieurs clés élémentaires. Ainsi, on parlera de l'acteur numéro X dans le répertoire numéro Y, ce qui est unique. D'ailleurs, IMS

travaille présentement sur l'identifiant unique (appelé GUID ou *Global User Identifier*). Ceci favorise l'interopérabilité et l'échange de données entre répertoires. La métadonnée "Source ID" est la clé de l'acteur dans notre répertoire (i.e son numéro matricule) alors que "Index ID" est l'identifiant du répertoire.

Le saisie des métadonnées est obligatoire pour les acteurs de type apprenant. Pour les autres, on peut s'en dispenser. Les boutons *Nom*, *Adresse* et *Contact* offrent la possibilité de saisir plusieurs noms ou surnoms, adresses (domicile, travail, permanente, etc.), contacts (téléphones fixes, fax, cellulaire, pager) le cas échéant. La Figure 4.13 montre l'exemple des contacts d'un acteur.

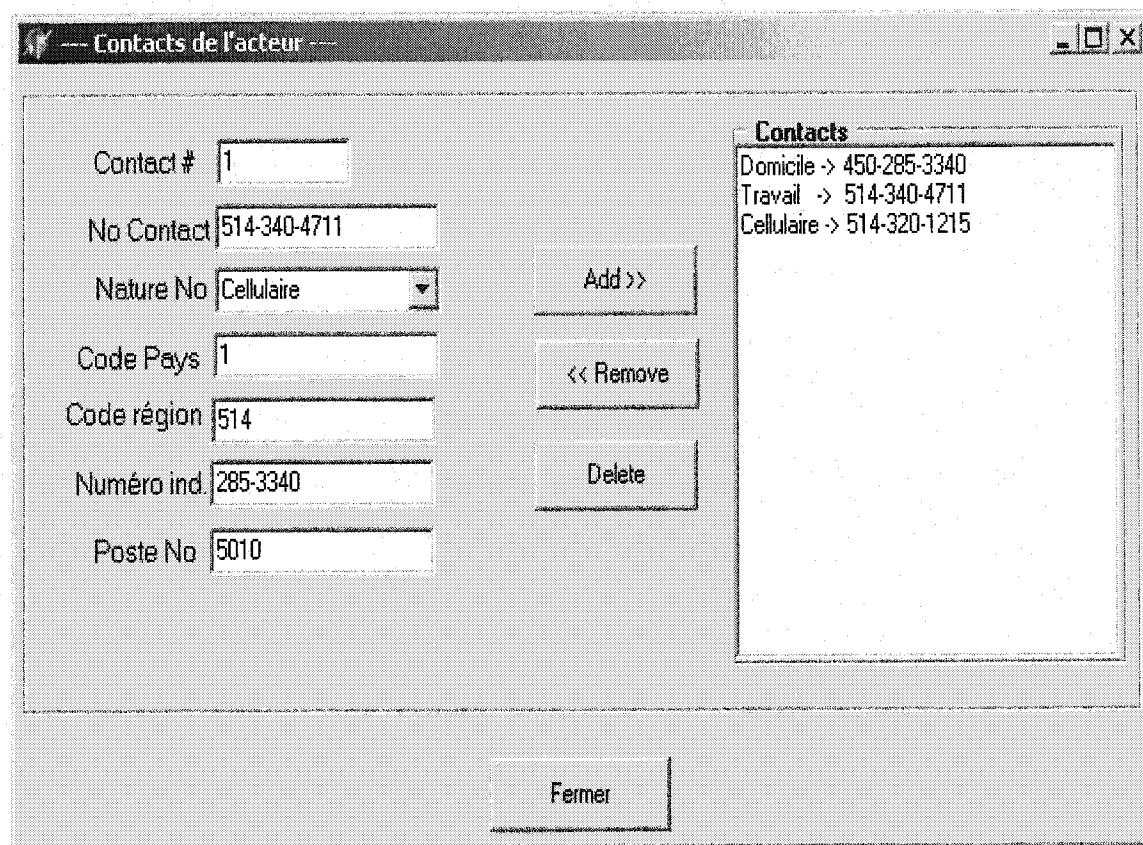


Figure 4.13 Méta-données sur les contacts d'un acteur

4.5 Expérimentation et analyse des résultats

Bien souvent, lorsqu'il s'agit d'expérimenter un système informatique, on est

confronté à la question du choix des métriques. Dans cette section, nous adoptons un critère de mesure et expliquons nos hypothèses et nos scénarios de test. Enfin, nous terminons par les mesures proprement dites et l'analyse des résultats obtenus.

4.5.1 Critère expérimental

Le répertoire de cahiers électroniques est utilisé par une équipe d'apprenants qui ne résident pas au même endroit pour partager des informations pendant la séance de laboratoire virtuel. Cette manière de faire sera d'autant plus efficace que les modifications effectuées par les uns seront rapidement portées à la connaissance des autres membres du groupe. De ce fait, le temps de réponse est le facteur qui caractérise le mieux l'interactivité. Alors, nous le retenons comme critère expérimental.

4.5.2 Hypothèses et scénarios de test

Nous considérons que les apprenants ont statistiquement les mêmes chances d'accéder au répertoire par des liaisons réseaux lentes ou rapides, indépendamment de leur distribution géographique. En réalité, ceci n'est pas forcément vrai, puisque certaines régions du monde sont mieux pourvues que d'autres en conditions et moyens d'accès. Mais sur la grande masse d'usagers de l'Internet, cette hypothèse est raisonnable. Autrement, nous n'avons aucun moyen d'en estimer les probabilités. Pour les mêmes raisons, si les temps de réponse sont satisfaisants pour un groupe, il y a de fortes chances qu'ils le soient également pour les autres. De ce fait, on peut ramener l'étude du temps de réponse à un seul groupe, à condition de prendre en compte tous les types de traitements. On suppose également que le serveur de la plate-forme et la base de données peuvent supporter un grand nombre de groupes. Au besoin, on peut éventuellement les répliquer afin d'en accroître les capacités. En ce qui concerne la taille, nous considérons un groupe de trois personnes. Ceci est raisonnable car le nombre de membres dans un groupe est petit (3 à 4 personnes), ne serait-ce que pour des raisons purement pédagogiques.

Types de traitements

Classiquement, il y existe quatre types d'opération lorsqu'on interagit avec une base de données: la création d'enregistrements, la suppression la consultation et la modification. Dans le cas qui nous concerne, la création se fait en exclusion mutuelle, c'est à dire après un verrouillage, alors que la suppression obéit non seulement à la même règle mais ne peut être faite que par l'auteur du paragraphe. Ce sont donc deux opérations qui ne relèvent pas du partage et elles ne peuvent être combinées ni entre elles ni avec les autres. Pour ces raisons, nous ne les étudierons pas. Nous nous intéressons aux consultations et aux modifications dont nous considérons les différentes combinaisons.

Notations

Dans ce qui suit, nous adoptons les notations suivantes :

- $C = \text{une consultation}$ $M = \text{une modification}$ $2C + 1M = 2 \text{ consultations et une modification faites simultanément (chacune par un apprenant différent)}$;
- $[1C + 1M] + 1C = \text{une consultation et une modification faites simultanément sur le même paragraphe pendant qu'une autre consultation est faite sur un autre paragraphe}$. En résumé les crochets [] délimitent un ensemble d'opérations qui portent sur le même paragraphe. Ainsi, $[1C + 1M] + 1C$ est différent de $[2C] + 1M$.

En considérant ces notations, nous avons le graphe de scénarios de la Figure 4.14. Il fait ressortir qu'à tout instant, les membres d'un groupe de trois personnes sont tous entraîn, soit de consulter un objet d'apprentissage (le cas $3C$), soit de modifier (le cas $3M$), soit deux consultent alors que le troisième modifie ou inversement (le cas $2C + 1M$ ou bien $1C + 2M$). Il n'y a aucune autre possibilité que celles-là et le graphe de la Figure 4.14 couvre tous les cas. De même, chacune de ces alternatives peut impliquer trois paragraphes (3 accès indépendants les uns des autres), deux paragraphes (2 des 3 accès sont concurrents) ou un paragraphe (3 accès simultanés au même objet). Ce qui nous donne deux cas extrêmes (maximum et minimum de concurrence d'accès au répertoire

de cahiers électroniques) et un cas intermédiaire.

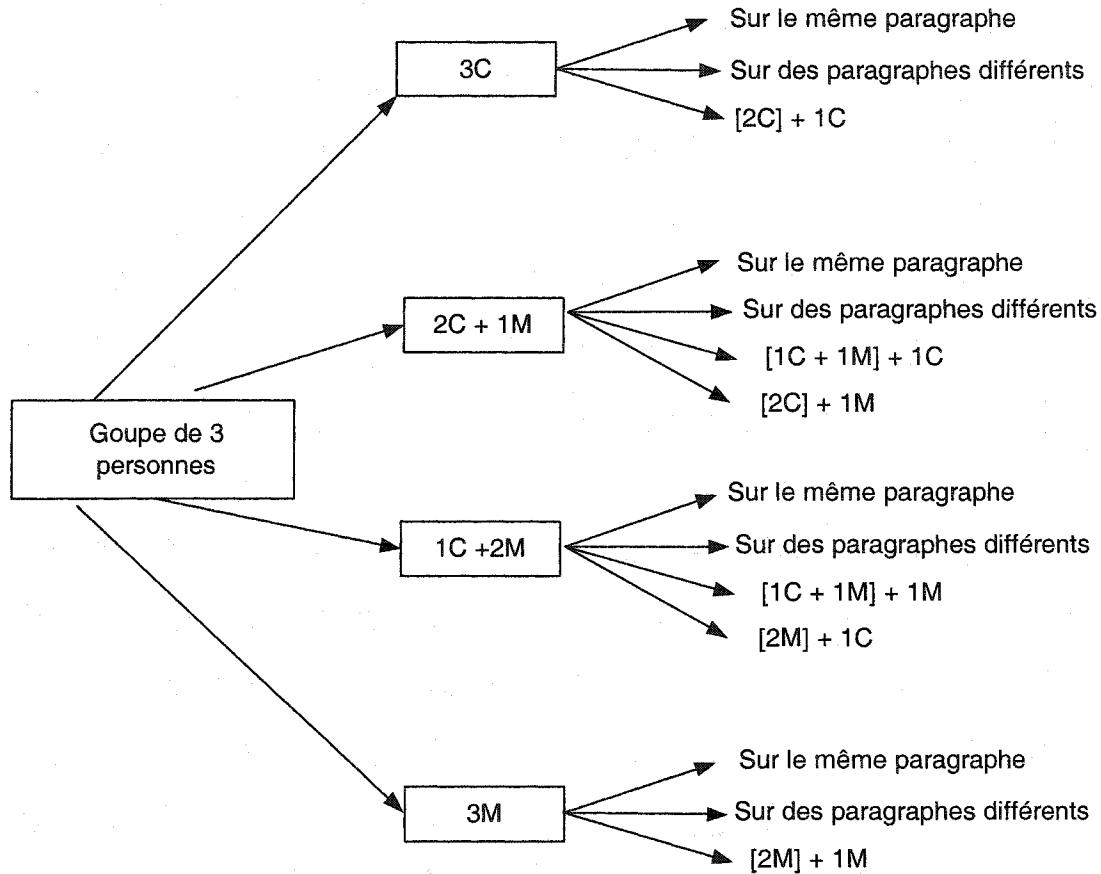


Figure 4.14 Graphe de scénarios

4.5.3 Résultats et analyses

Nous avons effectué trois séries de tests: lorsque tous les membres du groupe travaillent sur le même objet d'apprentissage ou paragraphe du cahier (exemple : passage matérialisant les étapes d'un calcul), lorsqu'ils sont sur des objets différents et lorsqu'il y a accès simultané aux objets. Ceci correspond bien aux cas les plus expressifs au regard du critère expérimental retenu qui est le temps d'accès. Pour effectuer les mesures, nous démarrons trois clients et le temps d'exécution retenu est la moyenne des temps de réponse. Pour cela, nous utilisons la classe Java **GregorianCalendar** et sa méthode **getTime()**. Avant d'adresser sa requête au serveur, chaque client initialise une

variable StartTime. Ensuite, il relit le temps et le met dans EndTime, dès qu'il reçoit la réponse du serveur. La différence EndTime – StatTime représente le temps de réponse.

Test 1 : Tous les membres du groupe sont sur des paragraphes différents

Ici, les apprenants accèdent de manière non concurrente aux objets d'apprentissage. Les dix séries de mesure effectuées nous ont donné les résultats du Tableau 4.1.

Tableau 4.1 Résultats du test 1

Scénario	Temps (ms)
3C	462
3M	1150
2C + 1M	644
1C + 2M	925

Ceci correspond au graphique de la Figure 4.15.

Analyse des résultats du test 1 : le temps de réponse moyen avec trois consultations distinctes sur des paragraphes différents est d'environ 462 millisecondes, ce qui est assez rapide. Mais dès qu'on change l'une des consultations en modification (2C + 1M), le temps augmente d'environ 40%. Des quatre scénarios étudiés, 3M est le plus défavorable avec un temps 2,5 fois supérieur à 3C.

Ce premier test montre que les modifications sont plus coûteuses en terme de temps de réponses, comparativement aux consultations. En effet, une consultation implique simplement une lecture de l'enregistrement demandé.

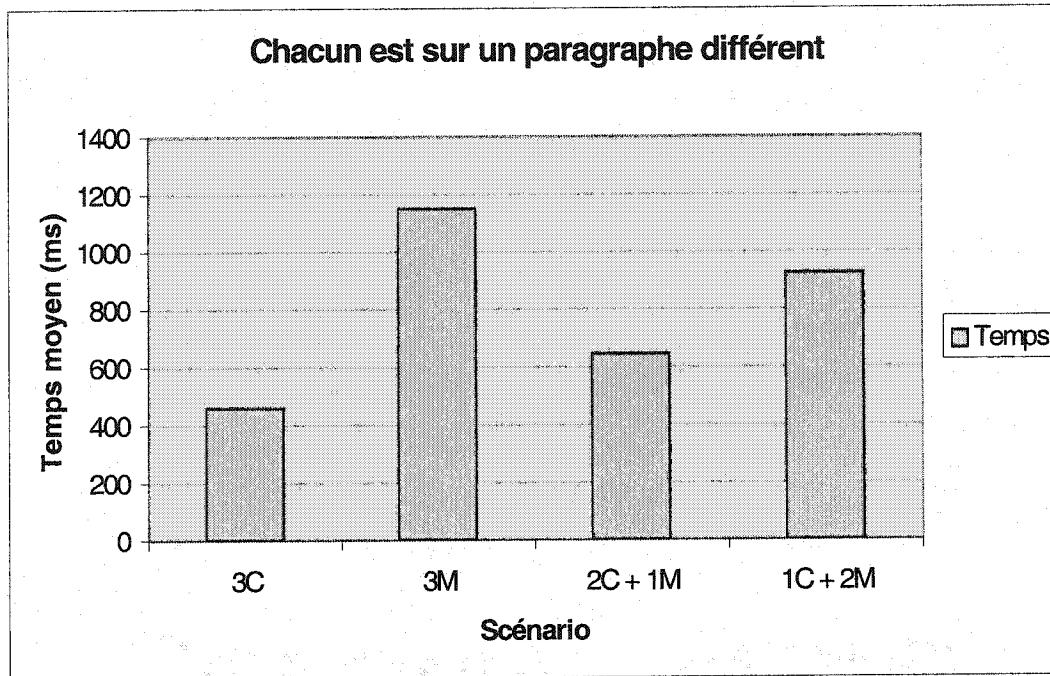


Figure 4.15 Temps de réponse du test numéro 1

Test 2 : Tous les membres du groupe sont sur le même paragraphe

Là, la concurrence des accès est la plus vive. Nous avons effectué dix séries de trois mesures dont nous prenons à chaque fois la valeur moyenne. Les résultats obtenus sont au Tableau 4.2.

Tableau 4.2 Résultats du test 2

Scénario	Temps (ms)
3C	477
3M	1908
2C + 1 M	1193
1C + 2M	1622

Ce qui donne le graphique de la Figure 4.16.

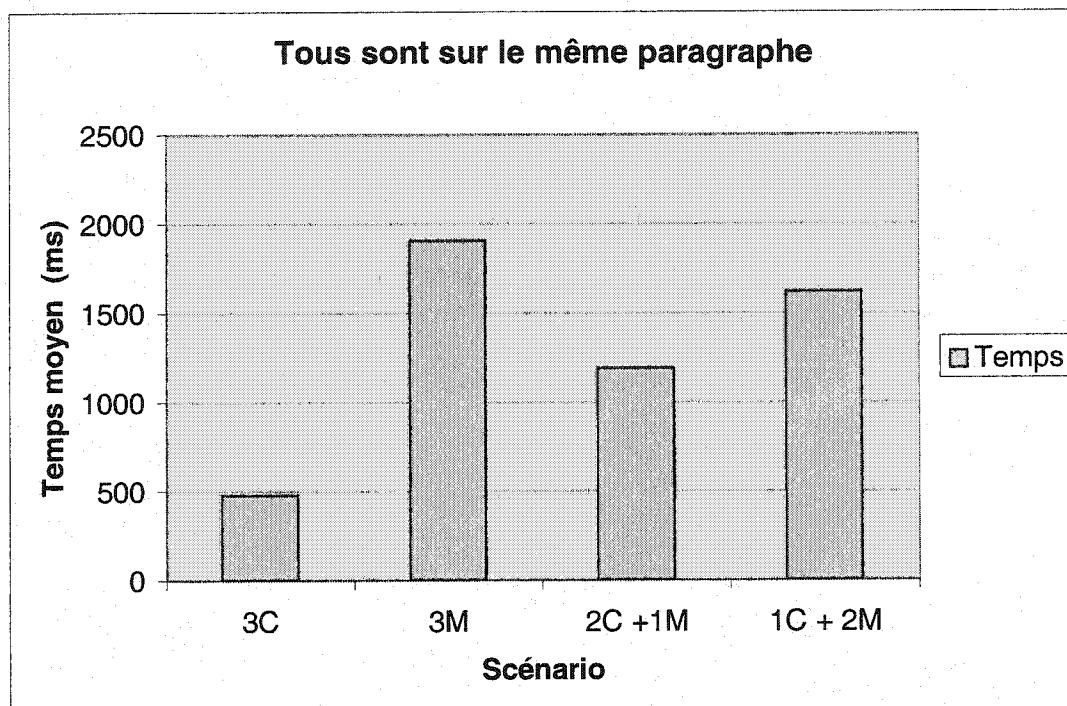


Figure 4.16 Temps de réponse du test numéro 2

Analyse des résultats du test 2 : Pour trois consultations simultanées du même paragraphe (3C), le temps de réponse est pratiquement identique à celui qu'il était dans le test 1. En effet, ces consultations reviennent à des lectures et donc, qu'on les fasse sur le même paragraphe ou sur des paragraphes différents, cela n'a pas beaucoup d'influence. Le scénario 3M est très coûteux car le premier apprenant qui envoie sa requête verrouille l'enregistrement et le suivant ne sera servi que si celui qui le précède finit. Le troisième apprenant est également bloqué par le second et ne sera servi que si celui-ci termine sa mise à jour. Le temps de réponse moyen qui était de 1150 millisecondes dans le test numéro 1 grimpe à 1908 millisecondes, soit une augmentation de 66% due essentiellement à la concurrence des accès.

Test 3 : Deux des trois membres du groupe sont sur le même paragraphe

Ici, on a des accès concurrents au répertoire contrairement au test numéro 1. Mais ils sont relativement moins nombreux que dans le test numéro 2 où tous accédaient simultanément au même paragraphe. En effectuant les mêmes mesures, nous obtenons les résultats du Tableau 4.3.

Tableau 4.3 Résultats du test 3

Scénario	Temps (ms)
3C	469
[2M] + 1M	1688
[1C + 1M] + 1C	844
[2C] + 1M	657
[1C + 1M] + 1M	985
1C + [2M]	1595

Ce qui correspond au graphe de la Figure 4.17.

Analyse des résultats du test 3 : [2C] + 1C est à peu près identique à 3C des tests 1 et 2.

Ce qui est logique puisque les consultations n'induisent que des lectures et celles-ci ne verrouillent pas les enregistrements. Le temps de réponse du scénario [2M] + 1M est plus proche du temps des trois modifications simultanées faites sur le même paragraphe (3M du test 2) que de celui de test 1 où les modifications ne présentent aucune corrélation (3M du test 1). Dans ce test numéro 3, on voit bien que [1C+1M] + 1C est plus lente que [2C]+1M car dans ce dernier scénario, la modification qui est l'opération la plus exigeante a lieu séparément. De façon similaire, [1C + 1M] + 1M est plus rapide que 1C + [2M] même si ces deux scénarios ont le même nombre de consultations et de modifications.

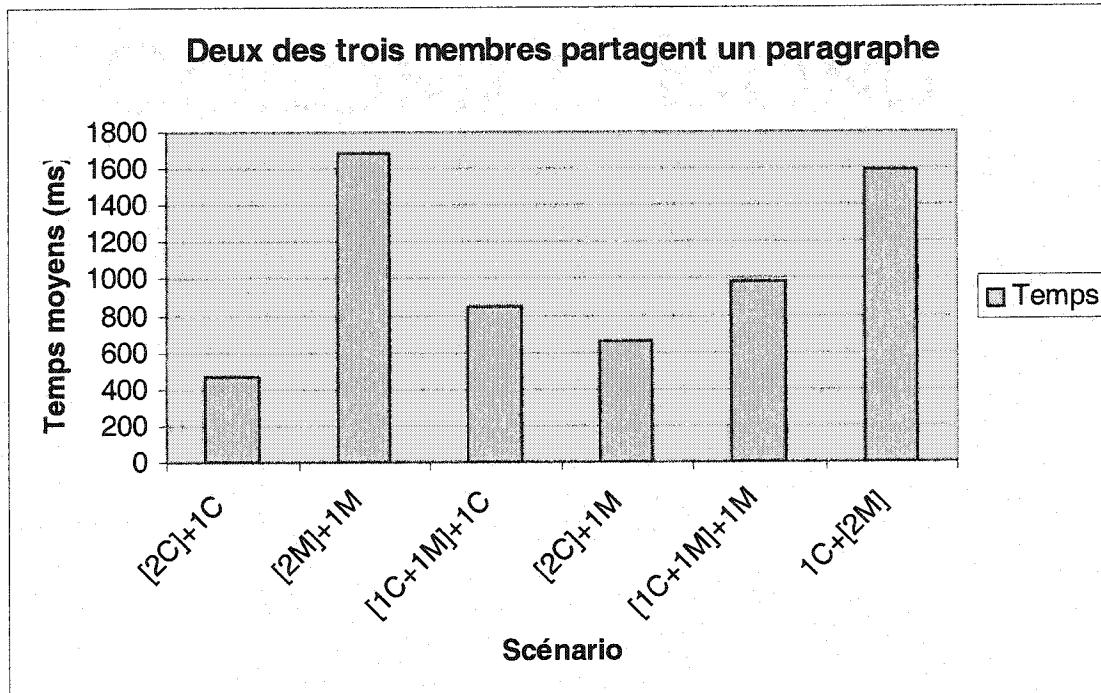


Figure 4.17 Temps de réponse du test numéro 3

Synthèse des résultats

Les mesures effectuées dans les trois tests sont regroupées dans le Tableau 4.4.

Tableau 4.4 Rapprochement des résultats des trois tests

Scénario	Test 1 (ms)	Test 2 (ms)	Test 3 (ms)
3C	462	477	469
3M	1150	1908	1688
2C + 1M	644	1193	-
1C + 2M	925	1622	-
[1C + 1M] + 1C	-	-	844
[2C] + 1M	-	-	657
[1C + 1M] + 1M	-	-	985
1C + [2M]	-	-	1595

Une entrée avec un tiret indique, que le scénario correspondant n'existe pas dans le test considéré. Ces mesures nous donnent le graphe de la Figure 4.18.

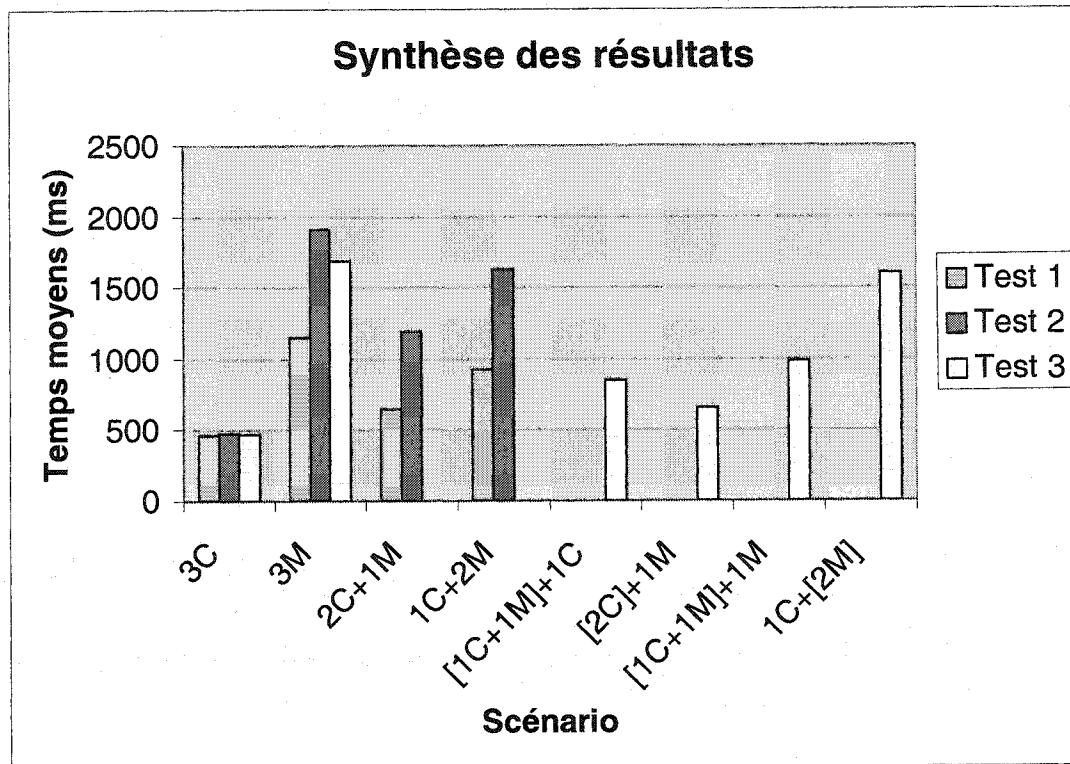


Figure 4.18 Synthèse des résultats

Sur la Figure 4.18, on voit bien que les temps de réponse en consultation varient très peu entre les trois tests, contrairement aux modifications. Aussi, les opérations les plus coûteuses sont les modifications simultanées du même paragraphe. Or, la probabilité de les avoir est d'autant plus faible que les paragraphes seront plus granulaires, ce qui correspond parfaitement à la philosophie des objets d'apprentissage.

CHAPITRE V

CONCLUSION

Dans ce mémoire, nous nous sommes intéressés aux questions relatives à une plate-forme de laboratoires virtuels répartis offrant des équipements partagés à des groupes d'apprenants qui effectuent des travaux pratiques. En guise de conclusion, nous faisons d'abord une synthèse de nos travaux puis nous explicitons leurs limitations pour terminer par des indications de recherches futures.

5.1 Synthèse des travaux

Notre première tâche a été d'élaborer une architecture de la plate-forme de laboratoires virtuels en partant du modèle conceptuel à trois couches introduit au chapitre 1. Pour y arriver, nous avons déterminé les composantes principales d'une telle plate-forme, étant donné que le requis primordial est d'assurer le partage de ressources tout en supportant le travail collaboratif. Ensuite, nous avons disséqué chacune de ces composantes en éléments plus simples en appliquant certains principes tels que la pertinence fonctionnelle de chaque module et l'obtention d'un degré de couplage aussi faible que possible entre eux. Ceci nous a conduit à une architecture dont nous avons décrit les douze constituants, abordant en même temps les voies et moyens qui permettraient de les mettre en œuvre. Enfin, nous avons étudié plus profondément deux de ces composantes: le répertoire de cahiers électroniques de laboratoire et le gestionnaire de la collaboration.

Pour le répertoire de cahiers électroniques, nous avons introduit le concept de base de données d'objets d'apprentissage et proposé une structure arborescente appelée répertoire (repository) qui intègre les laboratoires virtuels dans un cursus de formation, en tenant compte des disciplines et des domaines d'enseignement. Pour le réaliser, nous avons utilisé les standards de métadonnées LOM (Learning Object Metadata) et LIP (Learner Information Package) qui s'inscrivent dans la logique de l'émergence d'un marché mondial du télé-apprentissage (e-learning) avec des fournisseurs de contenus

(providers) et des consommateurs (consumers). Cette approche centre le design pédagogique sur deux concepts: *granularité* et *combinaison*. Granularité parce que les objets d'apprentissage qui encapsulent le contenu de l'enseignement doivent être les plus élémentaires possibles; et combinaison parce que chaque cours sera une composition judicieuse de tels objets.

Les principaux acteurs de la standardisation des métadonnées, socle sur lequel reposent les objets d'apprentissage ont été également passés en revue. Dans l'implémentation du répertoire, telle que nous l'avons faite, chaque groupe de travaux pratiques utilise un cahier pour chaque séance de TP qu'il fait. Celui-ci sert à échanger des informations (texte, son ou image) durant l'expérimentation. Aussi, les apprenants y cataloguent leurs résultats intermédiaires et les partagent avec leurs coéquipiers. L'étude de performance que nous avons faites, en prenant comme critère le temps de réponse, montre que les opérations de consultation sont assez rapides. Par contre, les mises à jour simultanées du même paragraphe (ou du même objet d'apprentissage) par plusieurs apprenants membres du même groupe sont les opérations les plus coûteuses. Mais, la probabilité d'avoir cette situation est elle-même d'autant plus faible que les paragraphes sont petits, ce qui épouse bien la philosophie des objets d'apprentissage. En outre, le recours aux objets d'apprentissage favorise la réutilisation, l'interopérabilité et facilite l'échange de contenus.

Pour le gestionnaire de la collaboration, nous avons proposé un modèle collaboratif multiagent en prenant en compte des équipes virtuelles. Chaque groupe de TP s'appuie alors sur un espace de collaboration. Entre les groupes, il s'agit de coopérer et de s'entraider, qu'on fasse ou non le même TP, comme cela se fait dans un laboratoire conventionnel réel. Pour nous, la différence entre ces deux types de relation (collaboration et coopération) découle du degré de couplage qui existe entre les actions des personnes concernées par l'activité. Lorsqu'il y a unicité d'objectif, action concertée et stratégie d'ensemble, il s'agit de collaboration. En d'autres termes, un groupe est une communauté d'apprentissage dont le fonctionnement découle des interactions entre ses membres.

Afin de matérialiser la notion abstraite qu'est un groupe pour un système logiciel, nous avons fait appel à un agent appelé *gestionnaire de groupe* (team manager). Celui-ci reçoit, ordonne, coordonne et exécute les requêtes de mise à jour exprimées par les apprenants membres du groupe. Les apprenants conservent l'initiative de l'action mais c'est l'agent qui les exécute. Pour avancer dans le TP, les apprenants modifient le montage expérimental.

En guise d'illustration, nous avons pris l'exemple d'un instrument virtuel LabVIEW dont les boutons de la face avant sont actionnés par un apprenant à l'aide du clavier et de la souris. Concrètement parlant, chacune de ces actions se traduit par l'exécution d'un code dont le résultat est affiché par l'instrument virtuel. L'agent peut parfaitement accomplir la même chose. Dans le modèle que nous avons proposé, l'instrument virtuel est perçu fidèlement par tous les membres du groupe grâce à des techniques telles que la mémoire virtuelle distribuée ou des mécanismes spécifiques comme AppletVIEW. Ce dernier permet de suivre dans un fureteur un instrument virtuel qui s'exécute à distance. Chaque *gestionnaire de groupe* gère ses membres dans une table locale d'usagers actifs. Lui seul interagit avec le reste de la plate-forme en tant que représentant du groupe. Les communications entre groupes s'appuient sur un *agent de courtage* (broker agent) qui tient un *service de répertoire* facilitant la localisation dans le système réparti constitué par l'ensemble des groupes. À toutes fins pratiques, nous avons indiqué la structure syntaxique des messages échangés entre les différents agents.

5.2 Limitations des travaux

En dépit des résultats satisfaisants obtenus, la performance des accès au répertoire de cahiers électroniques implanté comme une base de données d'objets d'apprentissage dépend également des hypothèses que nous avons faites. Par exemple, nous avons procédé comme si tous les usagers avaient des liaisons Internet aussi performantes les unes que les autres. De même, nous avons ramené l'étude à un seul groupe en considérant que si le temps de réponse était satisfaisant pour un groupe, ils l'est pour tout le monde et que le serveur est assez puissant (ou à défaut répliqué) pour

accommoder un grand nombre de groupes. Maintenant, il serait intéressant de lever certaines de ces restrictions et d'examiner leurs impacts. Par exemple, on peut considérer d'une part que les groupes ont des liaisons d'accès non identiques et d'autre part introduire cette même disparité à l'intérieur des groupes. Aussi, on peut augmenter progressivement le nombre de groupes et observer le comportement du répertoire sous haute charge. Mais, pour effectuer un tel test, il faut une base données plus élaborée, capable de supporter un grand nombre de traitements transactionnels dont les requêtes émanent d'apprenants connectés via un réseau WAN comme l'Internet. Ceci n'est pas le cas de *Microsoft access* que nous avons utilisé dans notre prototype.

Quant au modèle collaboratif, le niveau central (interactions entre groupes) pose des problèmes complexes tels que le partage efficient des ressources de la plate-forme entre tous les groupes et nous ne les avons pas tous abordés. En effet, on peut se demander jusqu'à quel point on peut partager ces ressources, combien d'agents gestionnaires de ressources partagées il faut mettre en œuvre, comment coordonner leurs interventions. Ce sont autant de questions qui restent ouvertes pour des recherches futures.

5.3 Recherches futures

Il serait opportun de poursuivre la réflexion sur le modèle collaboratif, l'implémenter et le tester. L'aspect partage de ressources dépendra beaucoup de la disponibilité d'équipements de laboratoires accessibles et manipulables à distance. Ces deux questions sont intimement liées car pour partager (le comment), il faut savoir concrètement ce qu'on partage (le quoi). À ce titre, LabVIEW permet de construire des instruments virtuels et de leur adjoindre au besoin un module d'acquisition de données pour des fins de télémesure/télémanipulation. De ce fait, l'émergence d'organismes comme *IVI Foundation* (Interchangeable Virtual Instrument Foundation), dont le but est de parvenir à des standards sur les pilotes d'équipements, peut conforter l'interopérabilité.

La sécurisation de l'accès au répertoire (par un système à clés asymétriques par exemple) et la flexibilité dans le choix du standard de méta-données utilisé (LOM,

SCORM, DCMI, ARIADNE) par l'implémentation de tables de transcodage entre standards, représentent des directions de recherches futures. Enfin, la mise au point d'un moteur de recherche Web destiné à découvrir des opportunités de télé-apprentissage par l'exploitation des notions de méta-donnée et d'objet d'apprentissage serait une démarche judicieuse et complémentaire à nos travaux.

BIBLIOGRAPHIE

- Ayal I., Assaf S. (February 1999), MultiView and Millipage – Fine-Grain Sharing in Page-Based, *Proceedings of The 3rd Symposium on Operating Systems Design and Implementation (OSDI '99)*, pp. 215--228, New Orleans. US patent pending. Application No. 60/093,232.
- Bassiliades N., Kokoras F., Vlahavas I., Sampson D. (2002), "An Intelligent Educational Metadata", to appear as a chapter in the book *Intelligent Systems, Techniques and Applications*, C.T Leondes (Ed.), CRC Press.
- Bastien J., Christian S. (Août 1999), "Outils pédagogique d'enseignement et d'apprentissage des mathématiques", *Colloque INITIATIVES'99*, Université de Moncton, Nouveau Brunswick (Canada).
- Begole J. M. A. (Dec. 1998), *Flexible collaboration transparency: Supporting Worker Independence in Replicated Application-Sharing Systems*, PhD dissertation, Virginia Tech., Dept. of Computer Science, Blaksburg, VA, (USA).
- Begole J. M. A, Randall B. S., Stubble A. C., Shaffer A. C. (Dec .2001), "Resource Sharing for replicated Synchronous Groupware", *IEEE/ACM transactions on networking*, vol. 9, no.6, pp. 833-843.
- Brandon L. B. (1990), A Generic Framework for Distributed Cooperating Blackboard Systems, *Proceedings of ACM Conference on Computer Science*, February 20-22, Washington, pp. 358-365.

Cecchet E. (Juil.2001), *Apport des réseaux à capacité d'adressage pour des grappes à mémoire partagée logicielle – conception et application*, Thèse de doctorat, Institut National Polytechnique de Grenoble (France).

Chetz C., Eileen S., Martin C. (April 2002), "Using remote laboratories to extend access to science and engineering", *Computers & Education Journal*, vol.38, issues 1-3, pp. 65-76.

Christian S., Bastien J., Duguay M., Warzée J. (2000), Laboratoires à distance et laboratoires virtuels, *VII^{èmes} Journées Internationales de Technologie (JIT)*, Université Saint -Joseph, École Supérieure d'Ingénieurs de Beyrouth (ESIB), Beyrouth, Liban.

Chung N., Cormenier M.N (Avril 2002), "Ressources éducatives numériques : normes et standards de métadonnées", in *Les fiches techniques*, édité par Educnet, ministère de l'Education nationale, France.

Cingeloglu Y., Khajenoori S., Linton D. (Sept 1994), "DYNACLIPS a dynamic knowledge exchange tool for intelligent agents", *Proceedings of The Third CLIPS Conference at NASA's Johnson Space Center*, pp. 357-366.

Edward W. (Dec.2000), "The new agenda for e-learning specifications", *Coalition for Networked Information , Task force meeting*, 7-8 December, San Antonio, Texas.

Flanagan D. (1999), *JAVA in a nutshell: a desktop quick reference*, 3rd Edition, Sebastopol, CA, O'Reilly.

Gulotta M. (Nov. 1995), "Teaching Computer interfacing with Virtual instruments in Object-Oriented Language", *Biophysical Journal*, vol. 69, no.5, pp. 2168-2173.

Himmelman A. (Jan. 2002), *Collaboration for a change, Definitions, Decision-making models, Roles and collaboration Process Guide*. The HIMMELMAN Consulting Group, Minneapolis, MN, USA.

IEEE P1484.12/D6.4 (Mach 2002), Draft Standard for Learning Object Metadata.

IMS Global Learning Consortium, Inc. (March 2001), *IMS Learner Information package – Information Model Specification, Final Specification*, Version 1.0.

Kassouf M. (Juin 1999), *Conception d'une plate-forme de télécommunication pour desservir des laboratoires virtuels distribués*, Mémoire de maîtrise, département de génie électrique et génie informatique, École Polytechnique de Montréal, 111 pages.

Le Préau (2002), Chambre de commerce et d'industrie de Paris : *Quel modèle qualité pour la e-formation ?*, 154 pages.

Lysakowski R. (May 1997), "Comparing Paper and Electronic Laboratory Notebooks", *Scientific computing & automation magazine*.

Marty H. (2000), *Servlets and JavaServer Pages*, CampuPress, Paris, 565 pages.

Mentré D. (Févr. 2001), *Une méthode de construction des mémoires partagées intégrant spécification, vérification et réalisation*, Thèse de doctorat, Université de Rennes I, France.

Moore, J. T. (May 1998), *Mobile Code Security Technics – Technical report MS-CIS-98-28*, University of Pennsylvania, USA.

Muller P.A, Gaertner N. (2000), *Modélisation objet avec UML*, 2^{ième} édition, Eyrolles, Paris, 540 pages.

Pierre S., Kassouf M. (2001), "Towards a Telecommunication Platform for Supporting Distributed Virtual Laboratories", *International Journal of Educational Telecommunications*, vol. 7, no. 2, pp. 157-194.

Rahkila M. (August 2001), "A double agent architecture for E-learning applications", *International conference on engineering education*, Oslo, Norway, pp.25-29.

Saliah-Hassane H. (Août 1999), "Mesures électroniques et télémanipulations sur les réseaux informatiques pour la formation, l'apprentissage et le télétravail collaboratif", colloque initiatives'99, Université de Moncton, Canada.

Saliah-Hassane H., Villardier L., Kedowidé C., B. Assaogba, Wong, T. (Oct. 2000), "Resource management strategies for remote virtual laboratory experimentation", *30th ASEE/IEEE Frontiers in Education Conference*, Kansas city, MO, pp. 8-12.

Suthers D. D. (August 2001), Evaluating the Learning Object Metadata for K-12 Education Resources, *Proceedings of The IEEE International Conference on Advanced Learning Technologies (ICALT 2001)*, Madison, Wisconsin, pp. 371-374.

Tadié G.S. (Juin 1998), *Un système multi-agent pour l'enseignement et la simulation des tâches coopératives*, Thèse de PhD, Université de Montréal (Canada).

Travias J. (2000), Internet Applications with LabVIEW, Prentice Hall, Upper Saddle River, NJ, 600 pages.

Vaughen-Nichols J. S. (Feb.2002), "Web Services: beyond the hype", *IEEE Computer*, vol. 35, no. 2, pp 18-21.

Vé F. (Avr.2001), *Méthodologie de conception des laboratoires virtuels et cadre d'expérimentation*, Mémoire de maîtrise (M.Ing.), département de génie électrique et de génie informatique de l'École Polytechnique de Montréal.

Wiley, D. A.(2000), "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy", *The Instructional Use of Learning Objects*, D. A. Wiley (Ed.), Bloomington, IN.

SITES INTERNET

1. Université d'OREGON (USA): <http://ww.jersey.uoregon.edu/vlab/piston>
2. Grand dictionnaire terminologique de la langue française :
<http://www.grand-dictionnaire.com>
3. The Open University (U.K), <http://ww.open.ac.uk>
4. The IVI foundation <http://ww.ivifoundation.org>
5. National University of Taiwan, <http://ww.phy.ntnu.edu.tw/java/wave>
6. Grouplab, University of Calgary, <http://www.cpsc.ucalgary.ca/Redirect/bmv/vlab>
7. Le Préau, "*centre de veille sur les technologies de l'information et de la communication pour l'éducation*" de la chambre de commerce et d'industrie de Paris :
<http://www.preau.ccip.fr>

8. Dublin Core Initiative, <http://dublincore.org>

9. IMS Global learning <http://www.imsproject.org>

10. Advanced Distributed Learning (DoD) , <http://www.adlnet.org>

11. Collaborative Electronic Notebook Systems Association (CENSA),
<http://www.CENSA.org>

12. Learning Technology Standards Committee of the IEEE Computer Society,
<http://www.lsc.ieeee.org>

13. <http://www.tecatlant.fr/labview/lvp1.html>

ANNEXE A

No	Name	Size	Type	Example	order
1	General	1 item	-	-	-
1.1	identifier (Lom)	10	-	-	-
1.1.1	catalog	1	string	"URI"	-
1.1.2	entry	1	string	"http://ieee.org/doc/123"	-
1.2	title	1	"	"The life of Jules Cesar"	-
1.3	language	max 10	"	"fr-CA"	no
1.4	description	"	"	("en", "in this video clip ...")	no
1.5	keyword	"	"	("en", "Jules Cesar")	no
1.6	coverage	"	"	("en", "16 th century France")	no
1.7	structure	1	énum	"atomic", "collection"	-
1.8	aggregation level	1	énum	1, 2, 3 ou 4	-
2	life cycle	1 item	-	-	-
2.1	version	1	string	("1.2 Alpha")	-
2.2	status	1	énum	"draft", "final"	-
2.3	contribute	max 30	liste	("Jean", "Jak", "Alan")	yes
2.3.1	role	1	énum	Author, editor	-
2.3.2	entity	max 40	string	author = Jak, validator=Alan	yes
2.3.3	date	1	dateT	"2001-08-25"	-
3	Meta-metadata	1	-	-	-
3.1	identifier	max 10	string	-	-
3.1.1	catalog	1	"	"ARIADNE", URI"	-
3.1.2	entry	1	"	"Kulw532", http://i.eee.org/12	-
3.2	contribute	max 10	"	"Poly", "Téluq", "UdeM"	yes
3.2.1	role	1	énum	creator, validator	-
3.2.2	entity	max 10	liste	("Pierre", "Paul", "David")	yes
3.2.3	date	1	dateT	"2001-08-28"	-
3.3	metadata schema	max 10	string	"LOMv1.0"	no
3.4	language	1	"	"en-GB", "fr-CA"	-
4	Technical	1	-	-	-
4.1	format	max 40	string	"video/mpeg", "text/html"	no
4.2	size	nb octet	"	"32"	-
4.3	location	max 10	"	"http://host/id"	yes
4.4	requirements	max 40	-	-	no
4.4.1	OrComposite	max 40	-	-	no
4.4.1.1	type	1	énum	MacOs, Windows NT	-
4.4.1.2	name	1	énum	Unix, Netscape, pc-dos	-
4.4.1.3	minimum version	1	string	"4.2", "1.0"	-
4.4.1.4	maximum version	1	"	"6.2", "4.6"	-
4.5	installation remarks	1	"	("en", "unzip before intall")	-

No	Name	Size	Type	Example	order
4.6	other platform reqt	1	string	("en", "sound card")	-
4.7	duration	1	string	"PT1H30M", "PT1M45"	-
5	Educational	< 100	-	-	-
5.1	interactivity type	1	énum	Active, expositive	-
5.2	L. resource type	max 10	"	"exercise", "simulation"	yes
5.3	interactivity level	1	"	"low", "very hight"	-
5.4	Semantic density	1	"	"very low", "medium"	-
5.5	intended end user	max 10	"	"teacher", "learner"	-
5.6	context	max 10	énum	("LOMv1.0", "high school")	no
5.7	typical age range	max 5	string	"7-9", "18-"	no
5.8	difficulty	1	énum	Easy, very easy, difficult	-
5.9	typical learning tim	1	string	"PT1H30M"	-
5.10	description	max 10	"	("en", "teacher's guidelines")	-
5.11	language	max 10	"	"en", "fr", "fr-CA"	no
6	Rights	1	-	-	-
6.1	cost	1	bool	Yes, No	-
6.2	copyright and other	1	"	"	-
6.3	description	1	string	("en", "use restricted to kids")	-
7	Relation	< 100	"	-	no
7.1	kind	1	énum	Ispartof, isbasedon	-
7.2	resource	1	-	-	-
7.2.1	identifier	max 10	-	-	-
7.2.1.1	catalog	1	String	"ISBN", "ARIADNE"	-
7.2.1.2	entry	1	"	"2-7342.03", "http:ieee.org/"	-
7.2.2	description	max 10	"	("en", "QuickTime movie")	-
8	Annotation	max 30	-	-	no
8.1	entity	1	String	"Larim- Poly Montreal"	-
8.2	Date	1	DateT	"2001-08-25"	-
8.3	description	1	String	("en", "video used with K-12")	-
9	Classification	max 40	-	-	no
9.1	purpose	1	énum	Idea, educational objective	-
9.2	taxon path	max 15	-	-	no
9.2.1	source	1	string	("en", "ACM"), ("en", "IMF")	-
9.2.2	taxon	max 15	string	{"Poly", "Gegi", "Larim" }	yes
9.2.2.1	Id	1	"	"320", "3.2.1"	-
9.2.2.2	entry	1	"	("en", "medical sciences")	-
9.3	description	1	"	("en", "medical stethoscope")	-
9.4	keyword	max 40	"	("en", "diagnostic instrument")	yes

Table extraite de Content Metadata (LOM Draf I3E 1484.12, Mars 2002)

ANNEXE B

No	Name	Explanation	Required	Multiple
4.1	typename	the type of goal	O	n
4.2	comment	comment of the LIP information		
4.3	contentype	language used for the comment		
4.4	date	dates appropriate to the goal	O	
4.5	priority	priority of the goal	O	
4.5.1	langtype	default language used for the goal		
4.6	status	recorded status for the goal	O	n
4.7	description	description of the goal itself	O	
4.8	goal	reference to subgoal	O	n
4.9	extension	extension facility	O	

O = optional n = multiple occurrences allowed

Méta-données de la catégorie "Goal" du standard LIP