

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UN OUTIL D'ASSISTANCE POUR LE DESIGN PÉDAGOGIQUE :
CONCEPTION ET IMPLÉMENTATION DU SYSTÈME CIAO

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE
OPTION GÉNIE LOGICIEL

PAR

GOUDJO-AKO PATRICK FULGENCE

AVRIL 2006

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

*À ma mère Anidié Rosalie Solange,
Pour Servais, Marie-Josée, Michel et Ashley.*

Remerciements

Je tiens à remercier le *laboratoire GDAC* et le laboratoire *LICEF* de la *Télé-Université du Québec À Montréal* pour la confiance mise en moi, et tout le soutien qu'ils m'ont accordé en mettant à ma disposition des moyens matériels et financiers qui m'ont permis de mener ma recherche dans les meilleures conditions possibles.

Je souhaite remercier particulièrement *Pr. Roger NKAMBOU*, directeur du laboratoire GDAC pour m'avoir accueilli au sein de son équipe et pour avoir accepté d'encadrer ce mémoire. Merci pour la confiance qu'il m'a manifestée, et le soutien sans faille qu'il m'a accordé.

Je remercie la *Pr. Jacqueline BOURDEAU*, professeur à la TÉLUQ, de m'avoir accueilli au sein de son équipe. Encore merci pour l'écoute, la disponibilité et le matériel mis à ma disposition. Merci à tous les employés et chercheurs(es) du laboratoire *LICEF* qui par leur bienveillance ont rendu mon séjour au sein du groupe des plus agréables.

Je remercie Mlle. Valéry PSYCHÉ, membre du laboratoire GDAC et chercheure à *Télé-Université du Québec*, qui a été ma collègue tout au long de ce projet, pour son étroite collaboration. Merci pour sa disponibilité, ses remarques, conseils et encouragements.

Merci également à Michel Léonard, Karin Lundgren, Marcelo Maïna, Diane Ruelland et Isabelle Savard, chercheurs et assistants de recherche au laboratoire LICEF pour l'aide et les commentaires fournis durant le projet CIAO.

Je remercie vivement Mr. *Pierre CLAVER SECYUGU*, professeur au Cégep Ahuntsic, pour l'attention qu'il a porté et continue de porter à tout ce que j'ai entrepris.

Mes remerciements vont également à l'endroit de Messieurs *Denis* PARADIS et *Régis* PARENT, professeurs aux Hautes Études Commerciales de Montréal, pour toute l'attention portée à ce que j'ai réalisé et pour l'honneur qu'ils m'ont fait en lisant ce mémoire. Merci d'apporter vos cautions scientifiques à ce travail.

Un merci à tous ceux qui de près ou de loin ont contribué à la réussite de ce travail.

TABLE DES MATIÈRES

| | |
|---|------|
| LISTE DES FIGURES | viii |
| LISTE DES TABLEAUX | x |
| LISTE DES ABRÉVIATIONS | xi |
| RÉSUMÉ | xii |
| CHAPITRE I | 1 |
| INTRODUCTION | 1 |
| 1.1 Contexte | 1 |
| 1.2 Problématique | 3 |
| 1.3 Objectifs de la recherche | 5 |
| 1.4 Plan du mémoire | 6 |
| CHAPITRE II | 8 |
| ÉTAT DE L'ART | 8 |
| 2.1 Les systèmes d'apprentissage en ligne | 8 |
| 2.2 Les systèmes auteurs et leurs limites en termes d'assistance aux usagers | 12 |
| 2.2.1 Comment sont développées les différentes parties d'un STI par un SA ? | 18 |
| 2.2.2 Quelles méthodes de conception et d'acquisition de connaissances ont été utilisées ? | 20 |
| 2.2.3 Vers des systèmes auteurs généraux ou spécialisés ? | 23 |
| 2.2.4 Utilisation et évaluation des systèmes auteurs | 23 |
| 2.2.5 Les limites des SA | 25 |
| 2.3 L'ingénierie ontologique et son rôle dans l'apprentissage | 26 |
| CHAPITRE III | 29 |
| UNE APPROCHE D'INTÉGRATION DE L'ASSISTANCE DANS UN SYSTÈME AUTEUR BASÉE SUR LES ONTOLOGIES DE THÉORIES | 29 |
| 3.1 Les ontologies des théories | 29 |
| 3.1.1. Les ontologies et la modélisation des connaissances | 30 |

| | |
|--|-----|
| 3.1.2. Les sémantiques computationnelles d'une ontologie..... | 33 |
| 3.1.3. Nécessité d'une ontologie de l'Instruction..... | 35 |
| 3.1.4. Le design pédagogique et sa prise en compte dans la construction de systèmes tuteurs intelligents | 38 |
| 3.2 Les scénarios, un des produits des systèmes auteurs..... | 40 |
| 3.3 CIAO, un pont entre l'ontologie et le système auteur | 43 |
| 3.3.1. CIAO et les ontologies de théories | 43 |
| 3.3.2. Architecture conceptuelle..... | 44 |
| 3.3.3. Modélisation des services du système CIAO | 47 |
| 1) Cas d'utilisations..... | 47 |
| 2) Diagrammes de séquence système | 55 |
| 3.3.4. Rappel des services offerts par CIAO..... | 60 |
| CHAPITRE IV | 61 |
| LES CHOIX TECHNOLOGIQUES ET L'IMPLÉMENTATION DE CIAO | 61 |
| 4.1 L'exploitation du Web sémantique | 61 |
| 4.1.1. Origines et perspectives du Web sémantique | 61 |
| 4.1.2. Quelques applications existantes du Web sémantique..... | 65 |
| 4.1.3. CIAO et le Web sémantique..... | 67 |
| 4.2 Les choix technologiques..... | 68 |
| 4.3 L'implémentation de CIAO | 70 |
| 4.3.1. La gestion des répertoires et des utilisateurs | 70 |
| 4.3.2. Lancement de l'application | 76 |
| 4.3.3. Le service d'exploration..... | 77 |
| 4.3.4. Le service de requête..... | 85 |
| 4.3.5. Le service d'analyse..... | 91 |
| 4.3.6. Le service d'exportation..... | 98 |
| 4.3.7. Le service d'écriture..... | 100 |
| 4.4 Évaluation de CIAO par des concepteurs | 105 |
| CHAPITRE V | 107 |
| DISCUSSION ET CONCLUSION | 107 |
| 5.1 Synthèse | 107 |

| | |
|--|-----|
| 5.2 Perspectives | 108 |
| ANNEXE A..... | 111 |
| CAS D'UTILISATIONS ASSOCIÉS AU SERVICE D'ANALYSE SYNTAXIQUE ET SÉMANTIQUE..... | 111 |
| ANNEXE B..... | 114 |
| STRUCTURE HIÉRARCHIQUE GLOBALE D'UN FICHIER DE SCÉNARIO PÉDAGOGIQUE ET RÈGLES DE CARDINALITÉ..... | 114 |
| B.3 Autres erreurs vérifiées durant l'analyse syntaxique | 119 |
| B.4 Règles de formulation des recommandations pour l'analyse sémantique | 120 |
| BIBLIOGRAPHIE..... | 122 |

LISTE DES FIGURES

| Figure | Page |
|--------|---|
| 2.1 | Fonctionnement d'un système tuteur intelligent. 11 |
| 3.1 | Interactions entre CIAO et le Système Auteur.....44 |
| 3.2 | Diagramme de séquence du cas d'utilisation 1.56 |
| 3.3 | Diagramme de séquence du cas d'utilisation 2.57 |
| 3.4 | Diagramme de séquence du cas d'utilisation 3.58 |
| 3.5 | Diagramme de séquence du cas d'utilisation 4.59 |
| 4.1 | Outil de configuration "Configure Sesame !".71 |
| 4.2 | Chargement de la configuration à partir du serveur.71 |
| 4.3 | Gestion des utilisateurs.72 |
| 4.4 | Gestion des répertoires.....73 |
| 4.5 | Détail des répertoires.74 |
| 4.6 | Gestion des droits d'accès.....75 |
| 4.7 | Envoi de la configuration au serveur en exécution.76 |
| 4.8 | Page d'accueil de CIAO.....76 |
| 4.9 | Service d'exploration de CIAO.....78 |
| 4.10 | Sous-service "Documentation" du service d'exploration de CIAO.79 |
| 4.11 | Sous-service "Hierarchy" du service d'exploration de CIAO.....80 |
| 4.12 | Sous-service "Description" du service d'exploration de CIAO.81 |
| 4.13 | Sous-service "Repository" du service d'exploration de CIAO.82 |
| 4.14 | <i>Domaine et Range</i> de propriétés dans une relation.84 |
| 4.15 | Résultat d'une requête en mode débutant sur le répertoire des classes.87 |
| 4.16 | Résultat d'une requête en mode débutant sur le répertoire des instances....87 |
| 4.17 | Requête en mode intermédiaire dans le répertoire des classes.88 |
| 4.18 | Requête en mode intermédiaire dans le répertoire des instances.89 |
| 4.19 | Requête en mode expert dans le répertoire des classes.91 |
| 4.20 | Scénario pédagogique syntaxiquement incorrect.92 |
| 4.21 | Exemple de résultat d'analyse syntaxique sur un fichier erroné.93 |

| | | |
|--------|--|-----|
| 4.22 | Résultats d'une analyse syntaxique sur un fichier sans erreurs. | 95 |
| 4.23 | Exemple de scénario pédagogique sans erreurs. | 96 |
| 4.24 | Résultat d'une analyse sémantique sur un fichier exempt d'erreurs. | 97 |
| 4.25 | Résultat de l'analyse effectuée sur un fichier erroné..... | 98 |
| 4.26 | Extraction du contenu de l'ontologie dans le répertoire des classes. | 99 |
| 4.26 a | Extraction du contenu de l'ontologie dans le répertoire des instances. ... | 100 |
| 4.27 | Processus d'identification d'un usager. | 101 |
| 4.28 | Mise à jour de l'ontologie à partir d'un fichier. | 102 |
| 4.29 | Mise à jour de l'ontologie à partir d'une adresse Web. | 102 |
| 4.30 | Mise à jour de l'ontologie à partir d'une opération copier-coller. | 103 |
| 4.31 | Suppression d'éléments dans l'ontologie..... | 104 |
| 4.32 | Confirmation de la suppression du contenu entier de l'ontologie..... | 105 |

LISTE DES TABLEAUX

| Tableau | Page |
|--|------|
| 2.1 Les sept catégories de SA | 13 |
| 2.2 Forces et faiblesses des systèmes auteurs par catégorie..... | 14 |
| 2.3 Degré d'utilisation de systèmes auteurs | 24 |
| 3.1 Cas d'utilisation 1 | 47 |
| 3.2 Cas d'utilisation 2 | 49 |
| 3.3 Cas d'utilisation 3 | 51 |
| 3.4 Cas d'utilisation 4 | 53 |
| A.1 Cas d'utilisation de l'analyse syntaxique..... | 112 |
| A.2 Cas d'utilisation de l'analyse sémantique | 113 |
| B.1 Éléments obligatoires validés dans le scénario pédagogique | 116 |
| B.2 Éléments optionnels validés dans le scénario pédagogique | 116 |
| B.3 Erreurs vérifiées durant l'analyse syntaxique | 119 |
| B.4 Règles utilisées pour l'analyse sémantique | 120 |

LISTE DES ABRÉVIATIONS

| | |
|--------------|---|
| AIDA | Advanced Instructional Design Advisor |
| AIED | Artificial Intelligence and EDucation |
| CAI | Computer-Aided Instruction |
| CIAO | Conception Intelligemment Assistée par Ontologies |
| DP | Design pédagogique |
| EIAH | Environnement Informatique pour l'Apprentissage Humain |
| IA | Intelligence Artificielle |
| IC | Ingénierie de la connaissance |
| IO | Ingénierie ontologique |
| LICEF | Laboratoire d'Informatique Cognitive et d'Environnements de Formation |
| OTDP | Ontologie des Théories du Design Pédagogique |
| OWL | Web Ontology Language |
| RDF | Resource Description Framework |
| RDF-S | Resource Description Framework – Schema |
| SA | Système auteur |
| SI | Science de l'instruction |
| SABC | Systèmes À Base de Connaissances |
| SeRQL | Sesame RDF Query Language |
| STI | Systèmes Tuteurs Intelligents |
| TA | Téléapprentissage |
| TELUQ | Télé-Université du Québec |
| URI | Uniform Resource Locator |
| USAF | United States Army Forces |
| XML | eXtensible Markup Language |

RÉSUMÉ

Les systèmes auteurs (SA) constituent aujourd'hui des outils vitaux en ce qui a trait aux activités de conception de cours dans le domaine du Design Pédagogique (DP). Ces activités permettent d'élaborer des systèmes tuteurs dotés d'une intelligence permettant d'offrir un cadre d'apprentissage et de formation approprié. Cette activité souffre toutefois d'un manque crucial car les SA actuels ne prennent pas en compte l'utilisation de théories pertinentes auxquelles peut se référer le concepteur. Le concepteur n'est ainsi pas en mesure d'avoir accès à ces théories directement à partir du SA. Face à cela, il devient difficile de garantir l'effet de l'apprentissage, car la structure de cours conçue à l'aide des SA ne peut être correctement validée. C'est pour faire face à ces problèmes et fournir une assistance active durant le processus de DP que le présent projet nommé CIAO a été initié. Il a pour objectif de développer un outil d'assistance active basé sur l'usage d'ontologies de théories du Design Pédagogique.

Dans le cadre de ce projet, nous avons axé notre attention sur la mise à la disposition du concepteur en tout temps d'une base de connaissance regroupant les théories de l'instruction, de l'apprentissage et du DP. L'accès à ces théories et au savoir qu'elles contiennent permettra à l'auteur de garantir une structure de cours fiable et efficace. Pour la mise en œuvre, nous avons utilisé (1) une base de connaissances contenant les théories associées aux activités de Design Pédagogique ; Cette base de connaissance permettra au concepteur d'explorer en tout temps les théories qu'elle contient et lui permettra d'obtenir des informations clés pour la réalisation de ses activités de DP; (2) des modèles de structures représentés sous forme de scénarios pédagogiques ; Ces scénarios constituent le lien entre notre système CIAO et les SA utilisés par le concepteur. Ils pourront être analysés, validés et modifiés en vue de les rendre compatibles aux normes de construction de cours et principalement aux objectifs et attentes visés par le concepteur. L'outil développé dans le cadre de ce projet permet alors d'intégrer une assistance active basée sur des théories pertinentes du Design Pédagogique. Cette assistance est jusqu'à présent l'élément manquant dans les SA.

Mots clés : Assistance au Design Pédagogique ; Système tuteur et ontologies ; Systèmes auteurs et assistance aux concepteurs ; Conception intelligente par Ontologies ;

CHAPITRE I

INTRODUCTION

1.1 Contexte

L'apprentissage en ligne (ou E-learning) est un mode de formation auquel on recourt de plus en plus fréquemment au vu des nombreux avantages qu'il offre à l'apprenant. Nous retiendrons la définition suivante: l'apprentissage en ligne consiste à donner accès de façon formelle ou informelle un à apprentissage, à des activités de formation, à des processus et à des événements d'apprentissage, en utilisant tout médium électronique comme l'intranet, l'extranet, le Cd-rom, la cassette vidéo, le dvd, etc. Cette approche comporte de nombreux avantages de formation:

- 1- **Pour l'entreprise** : Elle permet de réduire les coûts de formation et d'augmenter son capital de savoir et de connaissances.
- 2- **Pour le formateur** : Elle facilite l'accès aux étudiants grâce à la plateforme virtuelle utilisée et met à leur disposition des ressources pédagogiques de bonne qualité.
- 3- **Pour l'apprenant** : Elle permet le recours à des ressources d'apprentissage diversifiées et de bonne qualité à différents endroits. Elle permet aussi de moduler un horaire d'apprentissage, conciliant ainsi la formation à recevoir aux autres activités.
- 4- **Pour la société** : L'un des buts majeurs de l'éducation est d'améliorer la qualité de vie des citoyens. L'apprentissage en ligne y contribue grandement en rendant les ressources pédagogiques disponibles à un plus grand nombre et en permettant un accès flexible à ces ressources, ce qui accroît de façon générale le savoir de la société, et idéalement, contribue à sa transformation.

Pour rendre disponible un contenu didactique approprié pour l'apprentissage en ligne, les concepteurs ont de plus en plus recours au design pédagogique (DP). Le DP se définit comme suit : branche de la connaissance qui traite des recherches et théories associées aux stratégies d'instruction, et du processus pour le développement et l'implémentation de ces stratégies. Le DP (Gagné, Briggs, & Wager, 1992) est une façon systématique qui permet de planifier les événements en vue de faciliter l'apprentissage. Il comprend tout un ensemble de phases interdépendantes dont l'analyse des apprenants, des contextes d'apprentissage et des buts visés par la formation, la conception des objectifs, stratégies et outils d'évaluation, la production du matériel pédagogique utilisé pour la formation, et l'évaluation de la performance de l'apprenant.

Grâce au processus de DP, les concepteurs d'un cours iront au delà du simple design. Ils assumeront bien souvent le rôle de leaders en ce qui concerne l'analyse et la planification des activités associées au E-learning. Ils ont la tâche de faire le pont entre les concepts développés de part et d'autre des deux mondes fondant le E-learning, à savoir la technologie et l'éducation. Naturellement, ils seront responsables de la phase de conception, donc du choix des stratégies appropriées afin d'atteindre les objectifs d'apprentissage.

En plus de s'appuyer sur les principes (Mueller, 1998) du DP, les concepteurs se servent d'outils (Gros & Spector, 1994) comme les systèmes auteurs (SA) pour faciliter l'élaboration du contenu de la formation et des modules de mise en œuvre de la formation. Les objectifs (Murray, Blessing et Ainsworth, 2003) visés par les concepteurs en utilisant les SA sont de minimiser les coûts de production des systèmes de formation, de minimiser l'expertise requise de la part du concepteur, d'assurer la qualité du contenu et de la pédagogie à travers des stratégies pédagogiques retenues, de permettre la participation d'acteurs de profils différents (analyste de la formation, enseignant, étudiant, etc.), de disposer d'une plateforme d'évaluation des diverses stratégies et modèles de contenu. Même si les SA incluent des outils pour le développement des modules de mise en œuvre de la formation, ils sont plus souvent utilisés en grande partie durant la phase de développement (Paquette et al, 1994; Locatis & Park, 1992; Merrill, 1997) en vue de produire un contenu (général ou spécifique, c'est-à-dire associé ou non au domaine d'apprentissage) accessible par ordinateur. Selon Merrill (1997), ils simplifient la tâche

de programmation et permettent aux utilisateurs/concepteurs expérimentés de produire un contenu visuel très attrayant. Cependant, ils requièrent de l'utilisateur un travail énorme de familiarisation avec l'ensemble des fonctionnalités. Des exemples de SA populaires à l'heure actuelle sont : Macromedia Authorwave 4.0, Aim Tech IconAuthor, WBT Systems TopClass et Asymetrix Toolbook.

Toutefois, l'assistance fournie par les SA aux concepteurs durant les activités de DP est plutôt limitée. Durant ces activités, les concepteurs produisent entre autres des scénarios pédagogiques pour élaborer les cours. Ces scénarios sont bien souvent produits selon la norme XML-IMSLD¹ mais ne réfèrent généralement pas aux théories et modèles d'apprentissage, d'instruction ou du design pédagogique pourtant essentiels pour garantir la qualité (ou validité pédagogique) des produits du DP. Le présent travail s'insère dans le cadre de l'élaboration d'un outil offrant aux concepteurs de l'appui durant la conception de ces scénarios. Dans la prochaine section, nous présentons plus en détails la problématique de notre approche.

1.2 Problématique

La conception de scénarios pédagogiques sous-tend le DP. Elle se réalise à l'aide de SA, qui souffrent d'une lacune notable : ils ne permettent pas la production de scénarios pédagogiquement valides. Ainsi, comme ils ne respectent pas les théories d'apprentissage et d'instruction, ils ne peuvent garantir l'effet de l'apprentissage selon des principes et des règles associées à ces théories. Afin de remédier à ce défaut, il faut intégrer la pédagogie dans le processus même de construction. Il nous faut s'intéresser aux théories, aux modèles, afin de rendre ces différents éléments accessibles à l'auteur de façon à lui fournir le support cognitif requis.

Nous proposerons un système permettant d'assurer au concepteur une assistance active durant l'utilisation des SA. Dans un premier temps, la construction des ontologies associées aux différentes théories d'apprentissage, de l'instruction et du design pédagogique est nécessaire. Ces ontologies ont été élaborées par Valéry Psyché,

¹ <http://www.imsglobal.org/learningdesign/>

doctorante au laboratoire de Gestion Diffusion et Acquisition des Connaissances (GDAC) de l'Université du Québec À Montréal. Nous montrerons comment l'assistance au concepteur s'appuie sur ces ontologies. Nous identifierons ensuite les procédés de production de scénarios et des SA pour produire ces scénarios. Cette étape permettra de comprendre la structure des scénarios pédagogiques produits, afin de pouvoir offrir le support adéquat au concepteur. Après quoi, nous ferons la liaison entre ce procédé de production et les ontologies retenues. Cette liaison est assurée par notre système dénommé CIAO (système de Conception Intelligemment Assistée par Ontologie).

L'idée d'exploiter les ontologies de théories pour l'aide au design pédagogique est née en 2001 d'une discussion conjointe entre les Pr. *Roger NKAMBOU* (GDAC², UQAM), *Jacqueline BOURDEAU* (LICEF, TÉLUQ), *Riichiro MIZOGUCHI* (U. d'Osaka, Japon) et la doctorante *Valéry PSYCHÉ* (GDAC ET LICEF) chargée de l'élaboration de ces idées dans le cadre d'une thèse de doctorat en informatique cognitive. Les développements effectués dans le cadre de ce mémoire ainsi que l'expérimentation ont été menés de pair avec *Valéry Psyché*, doctorante à l'UQAM et chercheure au laboratoire LICEF³ de la TÉLUQ⁴. Le LICEF est un laboratoire en informatique cognitive et environnements de formation. C'est également le Centre de recherche de la Télé-Université. Il regroupe près d'une centaine de personnes qui élaborent des méthodes, des outils de conception et de réalisation de systèmes d'apprentissage. Par le LICEF, la Télé-Université est reconnue comme un acteur important dans le domaine de la recherche en formation en ligne. Il constituait donc un cadre adéquat pour mener notre recherche. Nous avons eu accès aux recommandations provenant d'experts de ce centre. La prochaine section présente les objectifs que nous visons avec CIAO.

² Gestion, Diffusion et Acquisition des Connaissances

³ www.licefteluq.quebec.ca

⁴ www.teluq.quebec.ca

1.3 Objectifs de la recherche

Le système CIAO⁵ offrira trois services: le service de navigation et d'exploration de l'ontologie, le service de recherche (requêtes) et le service d'analyse et de validation de scénarios pédagogiques.

Le premier service permettra au concepteur d'accéder au contenu de l'ontologie des théories par quatre canaux différents: la documentation de l'ontologie, la structure hiérarchique de l'ontologie, la description des restrictions entre les différentes classes de l'ontologie et la banque de données de l'ontologie dans laquelle l'utilisateur se déplacera en utilisant des hyperliens. Chacun de ces quatre canaux permet donc au concepteur d'explorer notre ontologie sous des angles différents.

Le second service de CIAO, celui de la recherche, comporte trois sous-services. Chacun d'eux permet à l'utilisateur l'accès à différents éléments de nos ontologies des théories, de l'instruction et du design pédagogique en se servant de requêtes écrites en langage SeRQL⁶ (Sesame RDF Query Language, prononcé "circle"). Dans un premier temps, l'utilisateur fait un choix parmi une liste de requêtes prédéfinies afin d'en voir les résultats. En second lieu, il peut être aidé par le système pour élaborer une requête dont il visualisera le résultat. La troisième version est réservée à l'utilisateur expérimenté maîtrisant le langage SeRQL. Avec cette version, l'utilisateur construit sa requête en saisissant le code approprié pour en visualiser le résultat. Le langage SeRQL permet d'exploiter les fonctionnalités de la plateforme Java Sesame. Les motifs de son utilisation sont détaillés plus loin dans la section 4.2.

Grâce au troisième service offert par CIAO, l'utilisateur validera son scénario pédagogique à n'importe quel moment de sa construction. Deux types de validation seront proposés à l'utilisateur, le premier basé sur une analyse syntaxique du scénario pédagogique et le second basé sur une analyse sémantique. L'analyse syntaxique permettra au concepteur d'obtenir la liste des erreurs de son scénario. Elle le guidera pour lui permettre d'apporter les

⁵ <http://www.licef.teluq.quebec.ca/CIAO/>

⁶ <http://www.openrdf.org/doc/sesame/users/ch06.html>

changements requis. L'analyse sémantique ne s'effectue que si le scénario fourni par le concepteur ne comporte aucune erreur de syntaxe dans sa construction. Elle permet au système de faire des recommandations au concepteur et l'oriente vers les théories du DP appropriées.

En plus des trois services principaux offerts au concepteur, CIAO disposera d'un service de modification autorisant à apporter des modifications au contenu de l'ontologie.

Le système CIAO offrira ainsi une interface permettant d'assurer la connexion entre les procédés de construction de scénarios et les théories pertinentes.

1.4 Plan du mémoire

En plus des chapitres d'introduction (chapitre 1) et de conclusion (chapitre 5), ce mémoire comporte trois autres chapitres, deux annexes et une bibliographie.

Dans le chapitre 2, nous présenterons un état de la question concernant les systèmes d'apprentissage en ligne et les SA. Nous examinerons ensuite les limites des SA en terme d'assistance et nous terminerons en présentant l'ingénierie ontologique et son rôle dans l'apprentissage en ligne et plus précisément dans le DP.

Le chapitre 3 présente les ontologies des théories liées au DP et les concepts qui s'y rapportent, puis les scénarios pédagogiques, qui sont le produit résultant de l'activité de DP effectuée à l'aide des SA. Après cela, il explique le système CIAO en le situant par rapport aux ontologies des théories, de l'instruction et de l'apprentissage développées dans le cadre de ce travail. Il explique l'architecture conceptuelle et la modélisation de CIAO. Il se termine par une brève présentation des services offerts par le système.

Le chapitre 4 présente les choix technologiques pris dans le cadre de ce travail ainsi que les détails de l'implémentation de CIAO. On y discute du web sémantique et de la

manière dont il a été exploité. Il se termine par une présentation de l'évaluation effectuée par des chercheurs du laboratoire LICEF de la Télé-Université sur le système CIAO.

Le chapitre 5 (conclusion) fait la synthèse de ce travail et relève les perspectives, ainsi que les futurs axes d'amélioration ou d'exploration.

Les annexes présentent des cas d'utilisations associés à certains services de CIAO et les repères qui ont servi à la validation des fichiers de scénarios pédagogiques.

CHAPITRE II

ÉTAT DE L'ART

Ce chapitre présente l'état de la question concernant les SA qui constituent les assises du design pédagogique. Nous commencerons par des généralités concernant les systèmes d'apprentissage en ligne. Nous décrirons ensuite les SA et leurs limites en terme d'assistance dans le domaine de l'apprentissage en ligne. Cela nous permettra de présenter les aspects liés à l'utilisation de systèmes auteurs pour la conception et le développement des systèmes tuteurs intelligents (STI). Nous traiterons ensuite de l'ingénierie ontologique et de son rôle dans le téléapprentissage et la conception de systèmes de formation à distance.

2.1 Les systèmes d'apprentissage en ligne

L'apprentissage en ligne est un mode d'apprentissage basé sur l'utilisation des nouvelles technologies, permettant l'accès à des formations en ligne interactives et parfois personnalisées, diffusées par l'intermédiaire d'Internet, d'un intranet ou autre média électronique, afin de développer les compétences, tout en rendant le processus d'apprentissage indépendant de l'heure et de l'endroit (Psyché, 2004). Il est souvent mis

en œuvre à travers un STI. Les STIs sont des systèmes de diffusion de la formation dans lequel des composants logiciels essaient de jouer le rôle d'enseignant (Psyché, 2004). Un STI est comparable à tout système de diffusion d'apprentissage connu dans le domaine des Environnements Informatiques pour l'Apprentissage Humain (EIAH), à la différence que le processus de diffusion est supporté par des systèmes à base de connaissances (SABC) qui utilisent des techniques d'Intelligence Artificielle (IA) dans leur tâche.

La notion de STIs conçus à des fins de formation a pris naissance au siècle dernier lorsque Sidney L. Pressley développa un système qui proposait un ensemble de questions à choix multiples avec les réponses associées. Ce système fournissait automatiquement le feedback nécessaire à l'apprenant. Depuis, les psychologues de l'éducation ont noté le fait que la conception de systèmes tuteurs personnalisés pouvait conduire à un meilleur apprentissage.

Les STI sont des systèmes fournissant un enseignement personnalisé. Ils contiennent des modèles de contenu d'apprentissage qui précisent *quoi* enseigner, en utilisant des bases de données séparées ou bases de connaissances, et des stratégies d'enseignement qui précisent *comment* enseigner (Wenger, 1987 ; Ohlsson, 1987). Ils effectuent des inférences sur le sujet d'apprentissage et les activités qui doivent être réalisées par un apprenant afin d'adapter dynamiquement le contenu ou le style de formation proposé.

La conception des STI repose sur deux idées fondamentales concernant le processus d'apprentissage. La première est que la formation individualisée fournie par un tuteur compétent est de loin supérieure à celle fournie selon le style d'enseignement en classe parce-que aussi bien le contenu que le style de la formation peut être continuellement adapté afin de répondre au mieux aux besoins de la situation (Bloom, 1956). La deuxième idée est le fait que l'apprenant se forme mieux lorsqu'il est confronté à des situations qui lui permettent le

plus possible d'utiliser ses connaissances. Il s'agit alors de situations pendant lesquelles l'apprenant se forme par la pratique, par les erreurs qu'il commet et cela, en se construisant une connaissance d'une manière qui lui est propre (Bruner, Ginsburg & Opper, 1979). Les STI, aussi appelés tuteurs à base de connaissances, utilisent des techniques favorisant un apprentissage automatisé qui se rapproche le plus possible de l'idéal, en simulant le plus fidèlement possible des situations du monde réel et en incorporant des modèles computationnels (bases de connaissances) dans le contenu, le processus de formation et l'état d'apprentissage de l'étudiant (Wenger, 1987).

Chaque STI s'appuie sur trois composants de base :

- la connaissance du domaine;
- la connaissance de l'apprenant;
- la connaissance des stratégies du formateur ou de la pédagogie.

À cela, il faut ajouter l'interface avec lequel l'apprenant interagit. Le domaine fait référence au sujet enseigné. L'apprenant réfère à l'étudiant ou l'utilisateur du STI. Les stratégies du formateur réfèrent aux méthodes de formation et la façon dont le matériel doit être présenté. Cet ensemble d'exigences a été introduit par Derek H. Sleeman et J.R. Hartley (Hartley & Sleeman, 1973). Le but d'un STI est de pouvoir dispenser de façon efficace la connaissance qu'il contient.

Fonctionnement d'un STI

Pour ce qui est de son fonctionnement, l'étudiant apprend d'un STI en résolvant des problèmes. Le système sélectionne un problème puis compare sa solution avec celle fournie par l'apprenant et effectue ensuite un diagnostic basé sur les différences constatées entre les deux solutions. Après avoir fourni cette rétroaction, le système fait une nouvelle évaluation, met à jour le niveau d'habileté de l'apprenant et répète ce cycle. Tout en évaluant la connaissance actuelle de l'apprenant, le STI considère ce que celui-ci a besoin de savoir, quelle partie de la formation doit être dispensée avant la prochaine évaluation, et comment la présenter à l'apprenant. Le schéma suivant illustre le fonctionnement d'un STI. Suite à cela, le STI sélectionne le problème que doit résoudre l'apprenant.

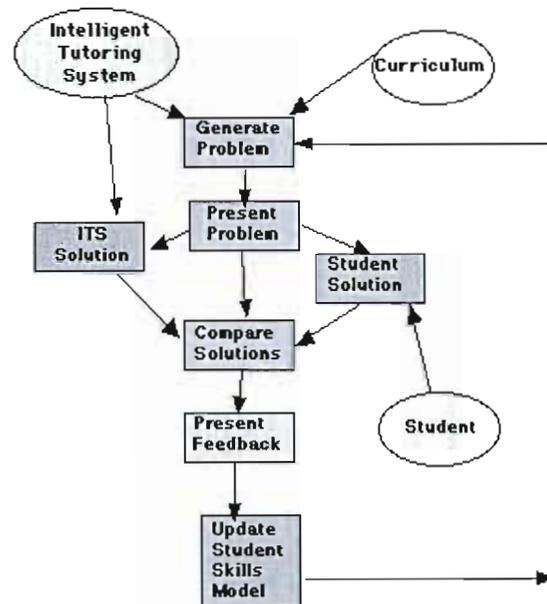


Figure 2.1 : Fonctionnement d'un système tuteur intelligent⁷.

Un STI devrait être capable de :

- diagnostiquer de façon précise les connaissances de l'apprenant et ses habiletés ;
- diagnostiquer en s'appuyant sur des principes plutôt que sur des réponses préprogrammées ;
- décider de la prochaine tâche à exécuter ;
- adapter l'enseignement aux compétences de l'apprenant ;
- fournir un feedback adéquat ;
- dialoguer efficacement avec l'étudiant ;
- encourager la réflexion de la part de l'étudiant.

Débats au sujet des STI

Les progrès effectués dans le domaine des ordinateurs ont facilité l'usage et la conception de STI. Cette prolifération des STI a donné naissance à plusieurs débats concernant leur utilité et leur efficacité :

⁷ <http://coe.sdsu.edu/eet/Articles/tutoringsystem/start.htm>

Le degré de contrôle de l'apprenant : Jusqu'à quel point doit-on permettre le contrôle à l'apprenant ?

Apprentissage collaboratif vs individuel : Les apprenants doivent-ils interagir de façon individuelle ou en groupe ?

Réalité virtuelle et apprentissage: La réalité virtuelle contribue-t-elle uniquement à l'étude ou peut-elle aller au delà de l'enseignement assisté par ordinateur ?

2.2 Les systèmes auteurs et leurs limites en termes d'assistance aux usagers

Au cours de la dernière décennie, les STIs ont migré du laboratoire vers la salle de classe et le bureau où bon nombre d'entre eux se sont révélés efficaces (Shute & Regian, 1990 ; Koedinger & Anderson, 1995). Mais ils demeurent difficiles et coûteux à développer. Les SA sont commercialisés pour la formation assistée par ordinateur mais ne disposent pas toujours de la sophistication requise pour construire des tuteurs complètement intelligents. Un système auteur est un système d'aide à la production. Lorsqu'on se situe dans le domaine de l'Intelligence Artificielle et de l'Éducation, il s'agit des systèmes d'aide à la production de tout artefact utile dans le processus d'apprentissage (système de formation, contenu, stratégie pédagogique). Il est donc utilisé par des personnes impliquées dans le processus de production (Psyché, 2004). Ces personnes regroupent les designers pédagogiques, les experts du domaine, les analystes du contenu, etc.

Les SA ont permis la construction de STI dans un large éventail de domaines d'activités, incluant les services commerciaux, les mathématiques et la maintenance d'équipements. Ces STI ciblaient différents types d'apprenants, depuis les classes des écoles primaires jusqu'aux employés des entreprises. Plus d'une douzaine de SA ont été élaborés⁸. Ils se différencient les uns des autres par les types de domaines et les tâches auxquelles ils s'adressent, par leur capacité à faciliter la construction de ST efficaces, et par le niveau de

⁸ Ces systèmes ne sont pas toujours décrits par leurs concepteurs comme étant des SA. La liste qui est fournie considère également comme SA les systèmes utilisant les techniques de l'Intelligence Artificielle comme les réseaux de règles, et aussi ceux utilisant des modèles de contenu pédagogique et/ou des règles d'enseignement.

détail et de précision employés pour représenter les connaissances ou les compétences à enseigner. Il est difficile de classer ces systèmes selon des catégories précises. Par conséquent, bien qu'un système soit présenté comme appartenant à une certaine catégorie, il contient bien souvent des caractéristiques propres à au moins une autre catégorie de SA. La classification suivante (Murray, 1999) des SA en catégories constitue essentiellement une "boîte à outils" permettant de décider adéquatement du type de SA à construire en fonction des besoins à satisfaire.

Le tableau 2.1 énumère sept catégories de systèmes auteurs, classés selon le type de systèmes tuteurs dont ils permettent la construction. Le tableau 2.2 présente un résumé des forces et faiblesses de chaque type de SA, de même que les différences entre ces différentes catégories de SA.

Tableau 2.1: Les sept catégories de SA

| CATÉGORIE | | EXEMPLES DE SA |
|-----------|---|--|
| 1 | Ordonnancement et planification des activités d'apprentissage | DOCENT, IDE, ISD Expert, Expert CML |
| 2 | Stratégies de formation | Eon, GTE, REDEEM |
| 3 | Simulation de dispositifs et entraînement d'équipements | DIAG, RIDES, SIMQUEST, XAIDA |
| 4 | Systèmes experts | Demonstr8, D3 Trainer, Training Express |
| 5 | Types multiples de connaissances | CREAM-Tools, DNA, ID-Expert, IRIS, XAIDA |
| 6 | But spécifique | IDLE-Tool/IMap, LAT |
| 7 | Hypermédia intelligent/adaptatif | CALAT, GETMAS, InterBook, MetaLinks |

On classe aujourd'hui la grande majorité des SA selon deux grandes catégories : orienté-pédagogie et orienté-performance (Murray, 1997-2003). Les systèmes orientés pédagogie (catégories 1, 2, 5 et 7 du tableau 2.1) mettent l'emphase sur l'ordonnancement des activités et les méthodes d'enseignement, tandis que ceux orientés performance (catégories 3, 4 et 6 du tableau 2.1) accordent plus d'importance au fait de mettre à la disposition de l'apprenant un environnement d'apprentissage riche dans lequel il puisse développer ses compétences par la pratique et par le feedback fourni par les STI.

Tableau 2.2: Forces et faiblesses des systèmes auteurs par catégorie

| CATÉGORIE | FORCES | FAIBLESSES | DIFFÉRENCES |
|---|---|--|--|
| Ordonnancement et planification des activités d'apprentissage | Utilisation de règles, de contraintes et de stratégies pour l'ordonnancement des leçons, des modules et des présentations. | Peu fiable du point de vue de l'apprenant ; Représentation peu profonde des compétences. | Varie selon que les règles de construction sont fixées ou modifiables ; échafaudage du processus de construction ; |
| Stratégies de formation | Stratégies d'enseignement de bas niveau ; Ensemble sophistiqué de primitives d'enseignement ; Différentes stratégies d'enseignement. | (Pareil que le précédent) | Méthode de représentation de la stratégie ; source de l'expertise d'enseignement. |
| Simulation de dispositifs et entraînement d'équipements | La construction des STI et la formation par les STI sont organisées pour l'identification, l'utilisation et le dépannage des composants d'un dispositif | Stratégies d'enseignement limitées ; modélisation de l'apprenant limitée ; sert surtout pour les compétences procédurales. | Fidélité de la simulation ; Facilité de construction ; |
| Systèmes experts | Connaissance/expertise du domaine exécutable ; Modélisation et diagnostic granuleux de l'apprenant ; Comprend des règles pour les débutants. | La construction du système expert est difficile ; limitée à l'expertise procédurale et à celle orientée vers la résolution de problèmes ; limitée aux stratégies d'enseignement. | Modèles d'expertise orientés performance vs modèles orientés connaissance. |
| Types multiples de connaissances | Représentation claire et méthodes d'enseignements prédéfinies pour les faits, concepts et procédures. | Limité à des faits, concepts et procédures relativement simples ; Stratégies d'enseignement prédéterminées | Intégration de l'ordonnancement de la formation ; types de connaissances/tâches supportées. |

| | | | |
|----------------------------------|---|---|---|
| But spécifique | Les systèmes basés sur les modèles fournissent de l'aide pendant la construction ; Un design particulier ou des principes pédagogiques peuvent être rajoutés. | Chaque SA est limité à un type spécifique de STI ; Représentation des connaissances et pédagogie non flexibles. | Degré d'inflexibilité |
| Hypermédia intelligent/adaptatif | Le Web fournit l'accessibilité et l'interface utilisateur fournit l'uniformité ; sélection et annotation adaptative des hyperliens ; | Interactivité limitée ; | Focalisation au niveau macro ou micro ; degré d'interaction ; |

Les partisans des théories constructivistes d'apprentissage (Jonassen & Reeves, 1996) critiquent souvent les STI orientés pédagogie et les théories du design pédagogique qui les sous-tendent comme étant trop "instructivistes", ignorant des aspects importants de l'apprentissage tels que la motivation intrinsèque, le réalisme du contexte ou les idées fausses communes. À l'heure actuelle, ces facteurs sont reconnus par la plupart des théoriciens du design pédagogique (Merrill, 1983 ; Gagne, 1985 ; Reigeluth, 1983), mais sont perçus comme pas suffisamment importants ou trop complexes ou complètement incompris pour être incorporés aux systèmes d'apprentissage.

Voici la description des catégories présentées dans les tableaux 2.1 et 2.2.

1- Ordonnement et planification des activités d'apprentissage

Les systèmes auteurs de cette catégorie organisent les éléments de formation (unités d'apprentissage) selon une hiérarchie de cours, de modules, de leçons, de présentations, etc., qui sont reliés par des préalables, et autres liens. Les unités d'apprentissage disposent généralement d'objectifs pédagogiques. Le contenu est entreposé dans des boîtes de texte et des graphiques. Ces systèmes sont perçus comme des outils qui aident les concepteurs

pédagogiques et les enseignants à construire leurs cours et à gérer la formation dispensée par ordinateur.

2- Stratégies d'enseignement

Les systèmes de cette catégorie excellent dans la représentation de stratégies d'enseignement diverses. Ils sont apparentés à ceux de la catégorie précédente car l'information est également emmagasinée dans des boîtes de texte et des graphiques, de plus la représentation du domaine est peu profonde. Les systèmes appartenant à la catégorie précédente se focalisent sur le "macro-niveau" d'enseignement (ex : l'ordonnancement par sujets ou modules) tandis que ceux de cette catégorie se situent à un "micro-niveau" d'enseignement. Les décisions d'enseignement à ce niveau incluent le temps et la façon de fournir les explications, les résumés, les exemples, les analogies, les trucs, les astuces, et le feedback. Elles incluent de plus le type de questions et d'exercices à offrir à l'apprenant.

3- Simulation de dispositifs et entraînement d'équipements

Pour les STI construits par des SA appartenant à cette catégorie, l'apprenant observe une pièce d'un équipement et doit en identifier les composants, effectuer des opérations, de la maintenance ou diagnostiquer le comportement du dispositif défectueux et réparer ou remplacer les parties défectueuses. Les compétences de ce type sont relativement répandues et génériques, ce qui rend largement utilisables les SA qui se spécialisent dans ce domaine. La tâche la plus difficile à réaliser lorsqu'on utilise ce type de SA est de construire la simulation du dispositif. Mais une fois qu'elle est réalisée, le reste se fait aisément. Contrairement aux deux catégories précédentes, les apprenants utilisant des STI construits avec des SA de cette catégorie feront de l'apprentissage par la pratique⁹.

⁹ Learning by doing

4- Systèmes experts

Plusieurs systèmes tuteurs intelligents incluent des modèles de connaissances basés sur des règles du domaine d'expertise. Ces systèmes, souvent appelés tuteurs de traçage de modèles (Anderson & Pelletier, 1991), observent le comportement de l'apprenant et construisent un modèle cognitif élaboré basé sur les connaissances de ce dernier qui peuvent être comparées avec celles du système expert. Les apprenants qui utilisent ces systèmes ont souvent à résoudre des problèmes dans un délai donné. Ils reçoivent du feedback lorsque leur comportement diffère de celui que propose le modèle expert.

5- Types multiples de connaissances

Les théories du design pédagogique classent connaissance et tâches selon deux catégories et proposent des méthodes d'enseignement pour chacune d'elles. Elles tendent à se limiter aux types de connaissances facilement définis, comme des faits, des concepts et des procédures. Bien que les types de connaissances et les méthodes associées varient selon les théories, elles proposent généralement un apprentissage orienté comme suit : les faits sont enseignés par la pratique répétitive et des dispositifs mnémoniques ; les concepts sont enseignés en utilisant des analogies et des exemples positifs et négatifs allant des plus simples au plus complexes ; les procédures sont enseignées étape par étape. Les SA de cette catégorie se différencient à plusieurs points de vue, mais utilisent tous un schéma de classification connaissances/compétences.

6- But spécifique

Dans cette catégorie, nous retrouvons les SA spécialisés dans des tâches ou des domaines particuliers. Les SA des catégories "Simulation de dispositifs" et "Types de connaissances multiples" fonctionnent aussi pour des types de tâches précis. Ceux de la catégorie "But spécifique" mettent l'accent sur des tâches encore plus précises. Il existe un principe selon lequel les SA construits pour des tâches ou des activités d'apprentissage précises conviennent mieux aux besoins des utilisateurs (apprenants) et aux concepteurs pour des situations données. Les systèmes de cette catégorie sont conçus en partant d'un tuteur

intelligent, puis créent un environnement de travail pour des tuteurs similaires. Le processus de construction associé s'appuie plus sur des modèles que ceux des autres catégories de SA.

7- Hypermédia intelligent/adaptatif

Au fur et à mesure que les systèmes hypermédias et les tuteurs Web deviennent sophistiqués, ils renferment un nombre grandissant de méthodes et de modèles appartenant au domaine des tuteurs intelligents. Les fonctions de ces systèmes recourent celles des systèmes appartenant aux catégories "Ordonnancement et planification des activités d'apprentissage" et "Stratégies d'enseignements" (selon que l'accent soit mis sur l'apprentissage au niveau macro ou micro). Comme c'est le cas avec bien des systèmes Web, le niveau d'interactivité et de fidélité offert à l'apprenant est faible pour les STI construits avec les SA de cette catégorie.

2.2.1 Comment sont développées les différentes parties d'un STI par un SA ?

Construction de l'interface

La conception d'interface est un domaine où les outils traditionnels de construction multimédia excellent comparativement aux SA. Ceci est probablement dû au fait que la construction d'un ensemble d'outils pour le design d'interface est une tâche qui prend énormément de temps. Etant donné que la construction de graphiques de base constitue un "problème résolu", la plupart des chercheurs en systèmes auteurs n'ont pas mis la priorité sur l'effort nécessaire pour le développement d'outils répondant à ce besoin. Cependant, l'expérience (Murray, 1998) a montré que la personnalisation de l'interface utilisateur est une priorité pour les concepteurs.

Construction du modèle du domaine

Les STI contiennent des représentations de l'expertise pour les connaissances d'activités de formation, de simulation de modèles et de résolution de problèmes. Des SA ont été construits pour chacune de ces catégories de domaines (Murray, 1999). Nous avons ainsi des outils tels que IDE, Eon, RIDES et CREAM-Tools (Nkambou, Frasson et Gauthier, 2003) pour le domaine des connaissances d'activités de formation, RIDES et SIMQUEST pour la simulation de modèles, et PUPS et Demonstr8 pour la résolution de problèmes par des systèmes experts.

Construction du modèle de l'enseignant

Les stratégies d'enseignement précisent la manière dont le contenu pédagogique est ordonné, le type de feedback à fournir, quand et comment coacher, expliquer, résumer, présenter un problème, etc. Bon nombre de méthodes de représentation sont utilisées pour modéliser l'expertise d'enseignement, incluant les procédures, les plans, les contraintes et les règles. Cependant la grande majorité des SA comportent un modèle d'enseignement fixe (non modifiable). Eon, COCA, REDEEM, IDE et GTE permettent la construction de modèles pédagogiques.

Construction du modèle de l'apprenant

Presque tous les systèmes auteurs (Murray, 1999) présentés ici utilisent le "recouvrement" de modèles de l'apprenant ; ex : Une valeur basée sur la performance de l'apprenant est assignée aux sujets ou aux étapes procédurales à exécuter. XAIDA et Eon permettent au concepteur de définir des idées fausses comme sujets d'apprentissage, de façon à ce que le STI puisse évaluer et remédier aux erreurs générales. Demonstr8 semble être le seul système utilisant un modèle de l'apprenant "exécutable" (c'est-à-dire un modèle pour prédire et simuler le comportement de l'apprenant). Diverses techniques de l'IA pour la modélisation ont été intégrées aux modèles de recouvrement, incluant la logique floue (Goodkovsky & Al., 1994) et les réseaux Bayésiens (Collins & Al., 1996). Eon semble être le seul système permettant la construction/modification du modèle de l'apprenant.

2.2.2 Quelles méthodes de conception et d'acquisition de connaissances ont été utilisées ?

De façon générale, les buts visés par les SA, classés par ordre d'importance ou de prédominance sont les suivants :

- 1- Réduire l'effort (temps, coût, et/ou autre ressource) nécessaire à la construction de STIs ;
- 2- Réduire les compétences nécessaires pour la construction de STI (ex : permettre à plus de personnes de participer au processus de conception) ;
- 3- Aider le concepteur/auteur à articuler ou organiser ses connaissances du domaine ou ses connaissances pédagogiques ;
- 4- Permettre (ex : structurer, recommander ou renforcer) l'usage de bons principes de conception (dans la pédagogie, l'interface utilisateur, etc.)
- 5- Permettre l'obtention rapide de prototypes de conception de STI (ex : Susciter des cycles de conception/évaluation rapides pour le prototype du système).

Les SA atteignent ces buts en utilisant un certain nombre de méthodes. Huit de ces méthodes (Murray, 1999) sont ici brièvement décrites.

1-Assemblage de l'articulation de la connaissance par des modèles

La construction de STI est à la fois un processus de conception et un processus d'articulation de l'information. La méthode la plus utilisée par les SA pour permettre aux non-programmeurs de construire des STI est d'assembler la tâche en y incorporant un modèle ou une plateforme de travail. Simplifier en limitant l'univers de ce qui peut être construit est une méthode intuitive.

2- Connaissance incorporée et connaissance de base

Une des façons de simplifier la construction de STI et de la rendre plus puissante est d'incorporer la connaissance à l'intérieur du système. Cette connaissance peut être active ou passive. La connaissance passive est celle qui fait implicitement partie de la structure

ou des contraintes imposées par un SA. Par exemple, les SA appartenant à la catégorie "Types de connaissances multiples" disposent de principes de design pédagogique incorporés à même leur structure.

3- Gestion de la connaissance

Les STI sont des systèmes élaborés et leur construction implique la gestion d'une quantité importante et complexe d'informations. Bon nombre de techniques de conception d'interfaces utilisateurs servent à divers SA pour aider à la gestion et à l'organisation de la connaissance. Le fait de simplifier l'entrée d'informations par l'utilisation de modèles, de formulaires de saisie de données et de menus d'aide est assez fréquent. Les outils permettant aux auteurs de faire des zooms avant et arrière entre le détail et la vue globale des images peut aider à gérer de grandes quantités d'informations.

4- Visualisation de la connaissance

La manière la plus efficace d'aider les concepteurs à mieux gérer une énorme quantité de connaissances interconnectées et complexes est de fournir de puissants outils de visualisation. Malheureusement, la construction de l'interface utilisateur est souvent la tâche la plus intensive dans la programmation d'une application interactive. À ce jour, bien que des outils permettant aux concepteurs de visualiser les stratégies d'enseignement existent dans ce domaine (Murray, 1999)

5- Éclaircissement de la connaissance et gestion des flux d'activités

L'acquisition de connaissances est largement reconnue comme étant le facteur constituant le goulot d'étranglement dans la construction de systèmes en IA. Un certain nombre de techniques ont été utilisées pour extraire la connaissance des experts du domaine. La plupart d'entre elles sont "manuelles" et nécessitent une intervention d'un ingénieur des connaissances ou l'observation de l'expert (Hoffman, 1987). Des outils logiciels ont également été développés en vue d'automatiser certaines tâches d'acquisition de la connaissance (Bosse, 1988 ; Shaw & Gaines, 1986).

6- Validation de la connaissance et de la conception

Chaque SA fait des compromis différents pour naviguer entre la conception libre et la conception régie par des contraintes. Toutefois, on ne peut pas perdre de vue que plus grande est la flexibilité offerte au concepteur, plus grandes seront les chances qu'il insère dans le système des données inconsistantes, imprécises ou des éléments qui ne respectent pas les principes actuels du bon design pédagogique (en fonction de la théorie utilisée durant la conception). Une des façons de permettre une conception flexible tout en maintenant un bon niveau de qualité est d'autoriser le concepteur à entrer ce qu'il veut de la façon qu'il le souhaite, mais d'inclure des mécanismes chargés de vérifier l'exactitude, la consistance, la complétude et l'efficacité des informations entrées.

7- Réutilisation de la connaissance

Les SA ont la capacité de rendre la construction de STI plus efficace grâce à la réutilisation des informations. À ce jour, la plupart des SA n'ont pas servi à construction d'un nombre suffisamment important de STI pour que nous soyons en mesure d'exploiter cet avantage. La mise en œuvre de la réutilisation implique le recours à une librairie structurée de ressources, où les sujets de construction, les activités, les stratégies, les composants d'interface, et/ou la connaissance du domaine pourraient être stockées indépendamment du STI et chargée au besoin dans n'importe quel STI pendant la phase de construction.

8- Création automatisée de la connaissance

Certains SA font des inférences ou créent des connaissances nouvelles ou de l'information, ce qui dispense le concepteur d'avoir à articuler et à rentrer cette information par lui-même. RIDES et Demonstr8 utilisent des techniques de programmation basées sur des exemples pour inférer des procédures générales à partir d'exemples plus précis fournis par le concepteur. Le système DIAG infère une grande quantité d'informations sur des dispositifs fautifs à partir d'une quantité relativement petite de symptômes de déficiences rentrés par le concepteur.

2.2.3 Vers des systèmes auteurs généraux ou spécialisés ?

L'un des désaccords les plus brûlants dans la communauté des chercheurs sur les SA concerne le degré approprié de généralisation pour un SA. Par exemple, le système LAT est conçu pour produire uniquement des STI chargés de former le personnel pour des services à la clientèle. C'est un outil général en ce sens qu'on ne peut y recourir pour créer un STI de service à la clientèle pour une variété de produits. Un SA qui se spécialise dans la production d'un type particulier de STI dispose de plusieurs avantages. En principe, en restreignant grandement l'univers de ce qui peut être conçu, un SA peut voir son utilité, sa fidélité, sa précision et son efficacité s'accroître. Des SA plus pointus diminuent le risque pour le concepteur de commettre des erreurs en entrant des données incorrectes, inconsistantes ou pédagogiquement dépourvues de contenu.

2.2.4 Utilisation et évaluation des systèmes auteurs

Utilisation des systèmes auteurs

Une mesure de la viabilité d'un SA est le nombre et la diversité de STI qu'il permet de construire, et le degré auquel le système développé a été utilisé, indépendamment du laboratoire dans lequel il a été élaboré. Bien sûr, le fait qu'un système n'ait pas été autant utilisé ne signifie pas que sa conception n'est pas viable. Mais étant donné que la production de systèmes utilisables requiert plusieurs étapes dans la conception basées sur la rétroaction obtenue des utilisateurs et des tests effectués, il apparaît raisonnable de penser que des systèmes qui n'ont pas été largement utilisés nécessiteront des travaux additionnels significatifs afin d'être pleinement utilisables. Le tableau suivant (Murray, 1999) montre une estimation brute du degré d'utilisation de divers systèmes. La catégorie 1 correspond à des prototypes récents qui ne sont pas encore considérés comme des SA pleinement fonctionnels, et qui ont été testés dans un petit nombre de domaines. La catégorie 2 regroupe des prototypes de systèmes qui sont des SA complets. Certains de ces derniers ont été utilisés pour la construction de plusieurs STI complets mais sous-utilisés. Les systèmes de catégorie 3 sont un peu plus robustes ou ont été plus utilisés que

ceux de la catégorie 2. Les systèmes de la catégorie 4 ont été utilisés dans la construction de plus d'une douzaine de STI, ont permis de construire des STI utilisés dans des situations réelles de formation, et ont atteint le stage de maturité en termes de robustesse et de documentation correcte de l'utilisateur. Les systèmes SIMQUEST et Training Express appartiennent à la catégorie 3 car bien que suffisamment robustes pour être commercialisés, il ne semble pas y avoir de preuve qu'ils aient été utilisés pour la construction de plus d'une douzaine de STI.

Tableau 2.3: Degré d'utilisation de systèmes auteurs

| | |
|--|--|
| 1- Prototypes récents et preuves de concepts | D3 Trainer, Demonstr8, DIAG, Expert-CML, IRIS |
| 2- Prototypes évalués ou utilisés | CREAM-Tools, DNA, Eon, GTE, IDLE-Tool, LAT, WebCT |
| 3- Prototypes modérément évalués ou utilisés | ISD-Expert/Training Express, REDEEM, SIMQUEST, XAIDA |
| 4- Prototypes grandement utilisés | IDE, CALAT, RIDES |

Productivité des systèmes auteurs

Les systèmes auteurs ont la capacité de réduire la quantité d'efforts nécessaires à la construction de STI, ou de permettre, avec cette même quantité d'efforts, d'augmenter l'adaptabilité, la précision et l'efficacité des STI. Des indications de ratios de développement pour certains SA sont disponibles (Murray, 1999). Ces résultats sont difficiles à interpréter mais laissent en la possibilité de développer des STI à un coût raisonnable. L'une des raisons qui explique cette difficulté d'interprétation est le fait que ces résultats n'incluent généralement pas le temps nécessaire à la création des graphiques.

Évaluation des systèmes auteurs

À ce jour, il y a eu relativement peu d'évaluations relatives aux SA. Ceci est en partie dû au fait que ces systèmes disposent de nombreuses fonctionnalités et qu'il est difficile de mesurer l'effet individuel de chacune d'elles, et de créer des situations de contrôle

permettant de comparer les résultats. Aussi, il serait possible d'argumenter que le seul fait de démontrer qu'une variété de concepteurs ont utilisé un SA pour produire une variété de STI actuellement utilisés par des apprenants est suffisant. Il existe un petit nombre de SA qui peuvent se "vanter" d'avoir été suffisamment utilisés pour se réclamer de cette "preuve d'existence". En plus, les preuves d'existence, bien que procurant une grande crédibilité au domaine étudié, sont d'une aide minime lorsqu'il s'agit de questions spécifiques portant sur la recherche. Compte tenu du fait que les SA sont relativement récents, des évaluations sommatives, qui montrent en apparence qu'un système "fonctionne", peuvent être de très faible valeur comparativement aux évaluations formatives, qui elles, donnent des indications sur ce qu'un système est effectivement capable de faire. Un certain nombre de méthodes d'évaluations qualitatives et formatives peuvent être utilisées (Murray, 1993) et certaines d'entre elles l'ont déjà été pour fournir un résumé d'évaluations de SA (Murray, 1999)

2.2.5 Les limites des SA

La construction de STI impliquent la création de contenu à l'aide de SA. Les limites actuelles des SA se situent essentiellement au niveau de l'assistance fournie au concepteur durant les activités de DP. Le concepteur n'est bien souvent pas encadré lorsqu'il s'agit d'obtenir une rétroaction sur la validité de certains modèles développés pour la formation. Selon (Murray, 1999), les SA ne disposent pas souvent de la sophistication requise pour construire des systèmes tuteurs complètement intelligents. C'est pour combler ce manque dans le cas de la production de scénarios pédagogiques pour la formation en ligne, que le projet CIAO a été développé. Il permettra ainsi de fournir une assistance au concepteur afin que ce dernier dispose d'un scénario répondant à des critères et des normes définies. L'activité de DP pendant laquelle le concepteur produit le contenu pour les STI pourra alors garantir la validité des éléments développés, et leur association à des théories d'instruction, d'apprentissage et du DP.

2.3 L'ingénierie ontologique et son rôle dans l'apprentissage

L'ingénierie ontologique (IO) revêt un potentiel important pour l'augmentation de l'intelligence dans les environnements de formation à distance, ainsi que pour la conception de ces environnements. Le téléapprentissage est un processus d'acquisition d'informations, de construction de connaissances et de développement de savoir-faire et d'habiletés, qui se réalise dans un environnement informatisé supporté par des réseaux, par l'intermédiaire d'interactions avec le système ou d'échanges interpersonnels répartis dans le temps et l'espace.

L'IO a son origine dans l'ingénierie des connaissances, science longtemps considérée comme le domaine de prédilection du développement d'expertise en conception des SABC. L'ingénierie des connaissances a évolué pour devenir l'ingénierie ontologique. Selon (Mizoguchi, 2004), on est passé "de l'intelligence artificielle à l'amplificateur d'intelligence", ou à l'accès à l'information, ou encore à l'assistant intelligent. Dans ce contexte, l'amplificateur d'intelligence n'est pas un logiciel de résolution de problème qui résout le problème à la place du concepteur ou de l'apprenant, mais un partenaire intelligent qui nous accompagne tout en restant invisible, et qui aide quand c'est nécessaire. C'est pour réaliser ce type de partenaire intelligent, donc pour raffiner et augmenter l'ingénierie des connaissances conventionnelle que l'ingénierie ontologique a été proposée (Mizoguchi, 2003).

Une étude élaborée (Psyché, Mendes & Bourdeau, 2003 ; Psyché et al. 2004) a permis de mettre en relief le rôle prépondérant joué par l'IO dans le domaine de l'apprentissage en ligne et dans les STI. L'IO est née du besoin éprouvé par certains chercheurs de la communauté de l'IA pour l'utilisation d'une ingénierie s'appuyant de manière solide sur des fondements et des bases théoriques et méthodologiques, afin d'améliorer la conception des systèmes intelligents : l'IO permet de spécifier la conceptualisation d'un système, c'est-à-dire de lui fournir une représentation formelle des connaissances qu'il doit acquérir, sous la forme de connaissances déclaratives exploitables par un agent (Psyché, Mendes & Bourdeau, 2003). L'ingénierie ontologique est une méthodologie qui

nous donne la logique du design d'une base de connaissances, le cœur d'une conceptualisation du monde-cible, des contraintes sémantiques de ces concepts ainsi que des théories et des technologies permettant l'accumulation de connaissances qui est indispensable pour le traitement des connaissances dans le monde réel (Mizoguchi, 2004).

Il est ainsi possible de définir l'ingénierie des connaissances comme étant la recherche d'heuristiques spécifiques au domaine pour la résolution d'un problème, et l'ingénierie ontologique comme étant la recherche de concepts généraux, réutilisables, partageables et durables pour construire un modèle de connaissances capable d'aider des personnes à résoudre des problèmes (Mizoguchi, 2004).

L'élément clé sur lequel s'appuie l'ingénierie ontologique est l'ontologie. Ce concept a été l'objet de plusieurs définitions de la part des chercheurs du domaine de l'IA. Dans ce milieu, le concept d'ontologie semble avoir été abordé pour la première fois par John McCarthy qui révéla le recoupement entre le travail fait en Ontologie philosophique et l'activité de construire des théories logiques de systèmes d'IA. La définition du terme Ontologie qui est la plus citée est celle énoncée par Gruber: « Une ontologie est la spécification d'une conceptualisation » (Gruber, 1993).

Les composantes d'une ontologie (GomezPerez, 1999) sont : les concepts, les relations, les axiomes et les instances. Les **concepts**, aussi appelés *termes* ou *classes* correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème), retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Les **relations** traduisent les associations (pertinentes) existant entre les concepts présents dans le segment analysé de la réalité. Les **axiomes** constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie. Les **instances** constituent la définition extensionnelle de l'ontologie. Elles véhiculent les connaissances (statiques et factuelles) à propos du domaine du problème.

Le rôle joué par l'informatique cognitive dans la conception des environnements de formation à distance n'est plus à démontrer (Paquette, 2002). De façon générale, l'apport spécifique de l'IO pour ces environnements serait la représentation formelle de

connaissances déclaratives, couplée à un mécanisme d'inférence exploitable par ces environnements (Psyché, Mendes & Bourdeau, 2003). Dans la perspective du Web sémantique, qui est en voie de devenir une assise pour les environnements de formation à distance, l'IO offre une sémantique riche, meilleure que toute autre méthode de représentation des connaissances connue. L'IO est également le type d'ingénierie qui se positionne le mieux pour permettre l'interopérabilité réclamée par le Web sémantique, à des fins d'opération des systèmes de formation à distance ainsi que le partage intra et inter systèmes. L'ontologie offre une meilleure performance puisqu'elle supporte une interaction directe lors de requêtes d'information par les usagers, à l'inverse d'autres structures de représentation de connaissances, comme les schémas décrivant les données stockées dans une base de données, qui sont utilisées elles, plutôt en mode différé (Psyché, Mendes & Bourdeau, 2003).

L'ingénierie ontologique tient une place prépondérante dans le projet CIAO que nous présentons car elle permet la construction de la base de connaissances contenant les ontologies des théories, de l'apprentissage et du DP développées par Valéry Psyché. Cette base de connaissances donnera accès à des concepts et des théories, en les reliant à des objets pédagogiques, offrant ainsi une base de connaissances déclaratives stable et consistante, à partir de laquelle on pourra offrir des services intelligents. C'est fort de tout cela, et en vue de fournir une assistance proactive basée sur des théories valides d'apprentissage que nous proposons le système CIAO, qui intégrera cette assistance avec un accès aux différentes bases ontologiques. Dans la suite du texte, nous emploierons le terme "ontologie des théories" pour faire référence à la base de connaissances contenant les ontologies des théories, de l'instruction et du DP.

CHAPITRE III

UNE APPROCHE D'INTÉGRATION DE L'ASSISTANCE DANS UN SYSTÈME AUTEUR BASÉE SUR LES ONTOLOGIES DE THÉORIES

Ce chapitre présente les ontologies de théories en tant que base de connaissances servant pour nos activités de Design Pédagogique. Nous verrons comment les ontologies, grâce à leurs sémantiques computationnelles aident le concepteur dans sa tâche. Nous aborderons également les fichiers de scénarios pédagogiques et montrerons que leur validation donne une assistance active au Designer Pédagogique. Finalement, le système CIAO sera présenté, et son positionnement entre les ontologies du Design Pédagogique et les scénarios pédagogiques sera précisé.

3.1 Les ontologies des théories

Au début, le développement des systèmes experts a donné des connaissances et des savoirs sans se préoccuper de la mise au point d'un processus de raisonnement et de logique formelle. Malgré le succès de ces systèmes à leurs débuts, les chercheurs du domaine ont constaté de sérieuses difficultés dans l'usage des technologies à base de

connaissances. Ces difficultés provenaient d'une part des coûts financiers de construction de ces systèmes et d'autre part de la faible possibilité de réutilisation des bases de connaissances. Pendant longtemps, les systèmes ont été construits en utilisant des connaissances heuristiques exploitant des technologies basées sur l'usage de règles simples. Il n'existait pas de méthode sophistiquée ou de théories permettant d'élucider la connaissance ou de transformer, d'organiser et de traduire la connaissance de façon à la rendre accessible et utilisable par les ordinateurs.

La modélisation de la connaissance apparaît comme une alternative aux technologies basées sur des règles (simples) et permet alors de vaincre les difficultés observées dans la construction des STI. Elle a rendue plus facile et plus claire l'observation de la connaissance et son utilisation dans une structure compréhensible par l'ordinateur. Comme exemples, nous avons le projet KADS en Europe (Breuker & De Velde, 1994), le projet PROTEGE aux États-Unis (Puerta & Musen, 1992) et le projet MULTIS au Japon (Mizoguchi, Tijerino & Ikeda, 1992) (Mizoguchi, Vanwelkenhuysen & Ikeda, 1995). L'idée maitresse est de trouver des activités qui soient indépendantes du domaine et qui précisent les rôles joués par les objets du domaine dans le processus de résolution de problèmes. C'est dans cette optique que les récentes recherches en matière de modélisation de la connaissance ont introduit l'idée de l'utilisation d'une ontologie.

3.1.1. Les ontologies et la modélisation des connaissances

Une ontologie est un système conceptuel qui permet le partage et la réutilisation des concepts grâce à une sémantique computationnelle (Mizoguchi, 2004). À l'origine, l'Ontologie était une orientation philosophique où l'on tentait de rendre compte de l'état de l'existant de façon formelle. En informatique, une ontologie s'entend comme un système de concepts fondamentaux représentés sous une forme accessible par un ordinateur. Le domaine des ontologies retient l'attention pour deux raisons :

- 1- une ontologie fournit une structure conceptuelle de base qui permet de développer des SABC partageables et réutilisables ;
- 2- une ontologie fournit l'interopérabilité entre les sources d'information et les sources de connaissances.

L'idée selon laquelle il n'existe pas de différence entre une ontologie et une base de connaissances, bien que partagée par de nombreux chercheurs en IA est une erreur. En effet, cette idée ne cherche pas à capturer une propriété essentielle de l'ontologie, à savoir l'attacher aux concepts plutôt qu'au vocabulaire, et elle porte sur ce qui existe dans le monde cible (Mizoguchi, 2004). Selon (Mizoguchi, 2004), une différenciation claire entre "ontologie" et "base de connaissances" devrait se faire à partir de son rôle, c'est-à-dire qu'une ontologie fournit un système de concepts utilisés dans la construction d'une base de connaissances; par conséquent, une ontologie peut être une spécification de la conceptualisation du monde cible que se fait l'ingénieur qui construit la base de connaissances, donc un méta-système d'une base de connaissances traditionnelle. Une ontologie n'est pas utilisée directement pour la résolution de problèmes ; elle fournit une spécification des connaissances et des modèles dans le système, tandis que les bases de connaissances conventionnelles sont utilisées pour la résolution de problèmes. Le rôle d'une ontologie vis-à-vis une base de connaissances est d'abord de donner des définitions des concepts utilisés dans la représentation des connaissances, puis de spécifier les contraintes entre les concepts, afin de rendre la base consistante et transparente, deux propriétés nécessaires pour pouvoir partager et réutiliser la connaissance – c'est le cœur des systèmes d'intelligence artificielle (Mizoguchi, 2004).

La puissance des ontologies réside en grande partie dans le fait qu'elles sont déclaratives. La représentation de connaissances déclaratives fait partie de beaucoup de systèmes d'IA, puisque c'est la garantie que le "système sait ce qu'il sait", ce qui lui permet de modifier son comportement en modifiant la connaissance qu'il possède (Mizoguchi, 2004). Si la connaissance était intégrée de façon procédurale, le système ne serait pas en mesure de changer son comportement. Les connaissances déclaratives permettent alors à un système d'expliquer son comportement. Lorsqu'elles sont représentées sous la forme d'une ontologie, elles permettent à un système de justifier ses connaissances et de garantir que le modèle produit (une instance) est conforme à l'ontologie.

Une ontologie comporte également à la fois une *ontologie des tâches*, qui décrit l'architecture de résolution de problèmes pour les SABC, et une *ontologie du domaine*, qui décrit le domaine des connaissances. Le concept d'ontologie a servi comme théorie

du vocabulaire (des concepts) utilisé pour la construction des différentes parties constituant la structure de modélisation pour la résolution des problèmes (Mizoguchi, Vanwelkenhuysen & Ikeda, 1995). L'ontologie des tâches fournit une méthodologie effective et un vocabulaire à la fois pour l'analyse et la synthèse des SABC. Elle offre une représentation des termes et concepts utilisés pour décrire la façon dont les experts effectuent les opérations de résolution de problèmes de manière complètement indépendante du domaine d'apprentissage. La recherche sur les ontologies d'un point de vue technologique est appelée ingénierie ontologique. Elle a contribué au développement de l'ingénierie de la connaissance vers l'ingénierie ontologique (Mizoguchi, Sinitsa and Ikeda, 1996a) (Mizoguchi, 1998).

La connaissance étant dépendante du domaine étudié, et l'ingénierie de la connaissance (IC) étant l'étude de celle-ci d'un point de vue technologique, il en a résulté de sérieuses difficultés dont a souffert l'IC compte tenu de cette spécificité et de cette diversité. Cependant, la recherche à partir des ontologies est différente. L'ingénierie ontologique traite de la connaissance en termes de ses origines et des éléments à partir desquels elle est construite. La nature hiérarchique des concepts et la capacité à structurer de la connaissance sont exploitées afin d'étudier les bases de la connaissance de même que les théories sur lesquelles elles s'appuient. Cette façon de procéder permet ainsi d'éviter les difficultés auxquelles a été confrontée l'ingénierie de la connaissance. Une ontologie des tâches est obtenue en analysant les structures des différentes activités intervenant dans la résolution d'un problème réel. Le but ultime de la recherche sur l'ontologie des tâches est de rendre disponible une théorie des concepts nécessaires à la modélisation du processus visant la résolution d'un problème humain.

Une ontologie des théories pour les STIs pourrait avoir d'un point de vue informatique une signification universelle (Mizoguchi, 2004). Une telle ontologie a été développée pour les besoins du projet CIAO (Psyché, 2004). Les théories de l'apprentissage n'ont pas à expliquer tous les phénomènes de l'apprentissage (Mizoguchi & Bourdeau, 2000) (Inaba, Ohkubo, Ikeda & Mizoguchi, 2003). Il suffit qu'une théorie s'applique dans un seul cas. Une ontologie des STI donne une structure conceptuelle commune pour reconstruire les théories et pour expliquer les raisons pour lesquelles une théorie

s'applique ou non (Bourdeau & Mizoguchi, 2002). Ceci implique qu'une telle ontologie peut être qualifiée d'universelle, même si elle n'est pas unique.

3.1.2. Les sémantiques computationnelles d'une ontologie

L'un des points cruciaux entourant le concept d'ontologie consiste à savoir si la sémantique qui l'entoure se réduit uniquement à un ensemble de termes juxtaposés. Force a été de constater qu'elle dispose de sémantiques informatiques très riches. Trois niveaux d'ontologie ont à cet effet été proposées (Mizoguchi, 1998).

Niveau 1 :

Une ontologie est une collection structurée de termes. La tâche fondamentale dans le développement d'une ontologie est l'articulation du "monde", du domaine auquel elle est rattachée. Cette tâche fait ainsi ressortir les différents concepts et leur hiérarchie. Des exemples d'ontologies de ce niveau incluent les hiérarchies de sujets rencontrés dans les moteurs de recherches sur le Web et les balises utilisées pour la description des métadonnées. Une légère définition des concepts est ainsi créée.

Niveau 2 :

En plus du contenu d'une ontologie de niveau 1, il est possible de rajouter des définitions formelles afin de prévenir des interprétations inattendues des concepts présents dans l'ontologie. Relations et contraintes sont formellement définies comme un ensemble d'axiomes. Les relations sont plus riches que celles présentes dans le niveau 1. Les définitions sont également déclaratives et formelles afin d'être interprétables par ordinateur. La possibilité pour l'ontologie d'être interprétée permet alors aux ordinateurs de répondre aux questions concernant les modèles construits à partir de l'ontologie. Une grande partie des efforts déployés dans la construction d'une ontologie cherchent à obtenir des ontologies de deuxième niveau.

Niveau 3 :

Rendue à ce niveau, l'ontologie est exécutable: les modèles dont elle permet la construction peuvent être exécutés en utilisant des modules fournis par quelques-uns des codes abstraits associés aux concepts de l'ontologie. L'ontologie devient capable de répondre à des questions se rapportant à la performance dans l'exécution des modèles. Des exemples typiques de ces ontologies se retrouvent dans les ontologies des tâches.

Avec ces trois niveaux, on peut se poser la question de savoir de quelle façon est-ce que l'ontologie contribue-t-elle à fournir de l'intelligence aux systèmes tuteurs dans leur construction. Premièrement, une ontologie de niveau 1 fournit un ensemble de termes qui peuvent être partagés par les chercheurs de la communauté, et ainsi être utilisés comme un vocabulaire structuré et partagé. Ce vocabulaire ainsi créé permet de convenir des spécifications associées aux fonctionnalités des composants (des STI), des stratégies d'apprentissage... . Dès lors, il devient possible de comparer différents systèmes en s'appuyant sur un ensemble d'idées partagé.

Une ontologie est également définie comme la "spécification d'une conceptualisation (Gruber, 1993)". De ce fait, elle représente de façon explicite cette conceptualisation qui a bien souvent été gardée implicite. Une ontologie de niveau 2 contient un ensemble de termes et de relations auxquelles sont associées des définitions formelles sous forme d'axiomes. Ces axiomes sont déclaratifs, ce qui permet à l'ontologie de représenter la conceptualisation de façon déclarative. Une ontologie de niveau 2 devient une source d'intelligence pour un système basé sur des ontologies.

Un des rôles d'une ontologie est de servir de méta-modèle. Un modèle propose une vue abstraite d'un objet donné. Une ontologie fournit des concepts et des relations utilisés dans la construction des parties du modèle. Les axiomes offrent les contraintes sémantiques entre les concepts. De ce fait, une ontologie spécifie les modèles à construire en fournissant des directives et des contraintes à satisfaire. L'ontologie joue donc un rôle au niveau méta-modèle. C'est une caractéristique que requiert un système auteur. Des recherches (Mizoguchi, Sinitsa & Ikeda, 1996b) (Ikeda, Seta & Mizoguchi, 1997 ; Murray, 1998) ont permis de montrer le rôle de l'ontologie en tant que méta-modèle.

Une ontologie partagée et acceptée constitue le premier pas vers une standardisation. La structure obtenue avec une ontologie de niveau 1 représente une compréhension du domaine du concepteur/développeur. En général, cette structure est plus informative et constitue plus qu'un ensemble de définitions de termes. Une ontologie ne peut pas devenir de façon instantanée un standard dans le milieu des chercheurs. Elle offrira par contre une plateforme de tests pour l'établissement d'un standard. Une fois les termes et les concepts de l'ontologie standardisés, la connaissance du domaine d'apprentissage pourra être systématisée selon ces concepts et selon les relations identifiées à l'intérieur de l'ontologie. Ceci permettra de conférer à une ontologie particulière son statut de standard. Des chercheurs (Mizoguchi & Bourdeau, 2000) du domaine de l'IA ont convenu de réaliser un projet ambitieux, celui de "la construction d'une ontologie du DP". Une telle ontologie permettra la prise en compte par un SA des théories existantes de l'instruction, de l'apprentissage et du DP. Dès lors, cela offrira une ontologie de l'Instruction que nous appelons, dans le cadre du projet CIAO, une ontologie des théories du DP.

La section suivante traite de la nécessité d'une ontologie pour les SA.

3.1.3. Nécessité d'une ontologie de l'Instruction

Bien qu'un grand nombre de chercheurs du domaine de l'Intelligence Artificielle et de l'Éducation soient bien au fait des théories de l'apprentissage, ils rencontrent toutefois des difficultés lorsqu'il s'agit de mettre en oeuvre des systèmes reposant sur un ensemble unifié, clairement défini et complet de concepts et principes. La question qui se pose est : comment à combler ce vide ?

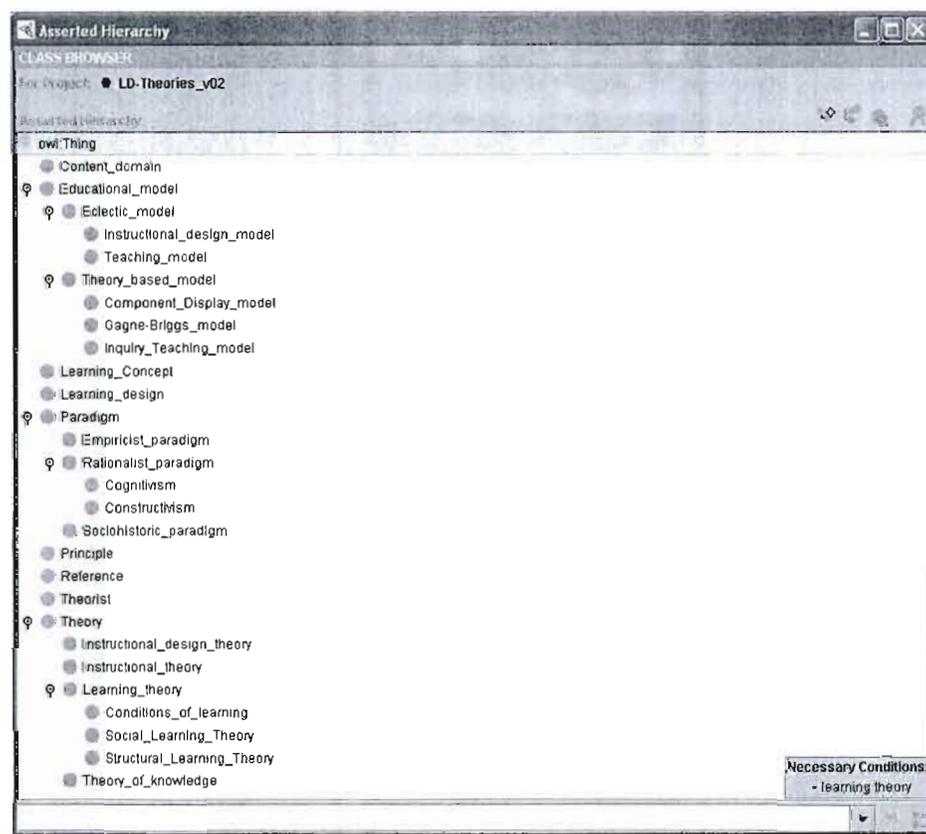
Un STI a besoin des termes et des concepts associés aux actions d'apprentissage, pour être en mesure d'amener les fonctionnalités du système vers des actions concrètes. La diffusion de la connaissance pédagogique et le partage de cette connaissance entre humains et ordinateurs est une nécessité (Mizoguchi et Bourdeau, 2000). Un accès facilité

aux théories de l'Instruction serait profitable autant aux humains qu'aux agents logiciels. L'Ingénierie Ontologique permet de spécifier les fonctionnalités de plus haut niveau d'un STI. Elle permet ainsi de combler le fossé existant entre la connaissance humaine et celle contenue dans les bases de connaissances. Les auteurs/concepteurs sont d'avis qu'une nouvelle direction doit être donnée à la recherche dans les domaines de l'IA et de l'Éducation. Celle-ci prendrait alors la forme d'une ontologie de l'Instruction (ontologie des théories de l'Instruction) qui pourrait ouvrir la voie à la mise en place d'environnements de construction pour les STIs. Ces environnements seraient alors dits "ID-aware", ce qui signifie qu'ils prendraient en compte le DP et les théories qui s'y rattachent. L'agent (logiciel) de construction des STI serait en mesure de faire référence aux théories appropriées selon le contexte, suite à des requêtes effectuées par le concepteur. Il pourrait également le conseiller pour l'usage de stratégies particulières de formation et d'apprentissage selon un point de vue théorique. Une telle approche montre l'utilité et le potentiel que nous serions en mesure de tirer de l'usage d'ontologies particulières (dont celle des théories du DP) durant la construction de STI. Dans cette optique, et selon Mizoguchi et Bourdeau, 2000, la première étape serait la construction d'une ontologie basée sur les théories de l'apprentissage existantes et sur les modèles du DP.

Toutefois, on peut penser qu'il existe un risque dans la construction des ontologies si celles-ci ne sont pas produites selon des normes précises. Si tel est le cas, l'introduction des ontologies pourrait alors générer de nouveaux problèmes et la prolifération d'ontologies idiosyncratiques conduirait à un état chaotique dans le domaine de l'IO. Toutefois, selon Mizoguchi, 2004, il est possible d'éviter cet état des lieux. Cela constitue une des préoccupations dans le domaine des ontologies : le contrôle est-il réparti ou centralisé ? Pour le Web sémantique, une ontologie pour les métadonnées peut être sous contrôle réparti ou sous aucun contrôle, tandis que pour la communauté en ingénierie des connaissances, une ontologie doit être mieux contrôlée, si on veut en faire autre chose qu'un simple vocabulaire compréhensible par l'ordinateur. L'idée de "l'ontologie universelle" n'est acceptée par personne mais il est également vrai qu'une ontologie ne peut être construite de façon arbitraire. Une stratégie d'ontologies dont le contrôle serait complètement réparti conduirait facilement à un désordre d'ontologies mal conçues. Une solution devrait se trouver entre les deux extrêmes en se rattachant à une ontologie de

haut niveau bien élaborée. En théorie, une ontologie devrait être développée par une communauté dont les membres partagent la nécessité de posséder une ontologie commune sur laquelle se reposer. En outre, les développeurs d'ontologies doivent se rattacher à une ontologie de haut niveau pour les guider dans la conduite d'ontologies "raisonnables", avec un environnement sophistiqué de construction d'ontologies. Cette façon de construire éviterait d'avoir des ontologies faites n'importe comment et plutôt "laides" (Mizoguchi, 2004).

Dans le cadre de ce projet, Valéry Psyché a développé une ontologie des théories de l'apprentissage et de l'instruction. Cette ontologie développée à l'aide des outils HOZO et Protégé, a permis la formalisation de trois théories d'apprentissage très différentes : les théories de Gagné, de Merrill et de Colins. Ces théories sont donc accessibles au concepteur et les fichiers OWL générés sont exploités par le système CIAO qui les utilise pour fournir plusieurs services utiles au concepteur. La figure suivante montre une portion de la structure hiérarchique des classes composant l'ontologie.



3.1.4. Le design pédagogique et sa prise en compte dans la construction de systèmes tuteurs intelligents

La Science de l'Instruction (SI) est née en 1966, avec la théorie de l'Instruction de Bruner et a pris de l'ampleur avec les travaux entre autres de Glaser, Ausubel, Gagné et Merrill. Elle a également accompagné les récents développements survenus dans le domaine des Sciences Cognitives¹⁰. Elle est constituée de théories, de modèles et de méthodologies portant sur l'Instruction et la Recherche sur l'Instruction. Elle dispose à la fois de composants descriptifs et prescriptifs. La partie prescriptive constitue le DP. L'objet principal sur lequel portent les études effectuées dans le domaine de la SI est l'interrelation existant entre trois types de variables : la situation d'apprentissage, le sujet d'apprentissage, et la stratégie d'apprentissage (Reigeluth, Buderson & Merrill, 1994).

Le DP est un processus systémique et systématique appliquant des stratégies et des techniques issues des théories behavioristes, cognitives et constructivistes, pour la recherche de solutions aux problèmes de l'apprentissage. Il propose une mise en œuvre systématique de théories et de connaissances structurées aux activités de conception et de développement d'apprentissage. Il fait intervenir six sous-activités que sont l'analyse, la conception, le développement, l'évaluation, la livraison et la gestion. La première étape effectuée l'analyse du sujet/domaine d'apprentissage, des buts et objectifs, des caractéristiques de l'apprenant, du contexte et des contraintes. Lors de la conception, il s'agit de prendre des décisions de haut niveau en s'appuyant sur des stratégies. Le développement comprend la prise de décisions de niveau moindre basées sur des activités et du matériel d'apprentissage et d'évaluation. La quatrième étape est d'évaluer le processus utilisé et le résultat obtenu. Par la suite, le produit obtenu est prêt pour les livraisons (étape 5). La sixième activité, qui intervient toutefois tout au long de l'activité de DP consiste à gérer la conception et le processus de livraison une fois le système complété.

¹⁰ <http://www.unc.edu/cit/guides/irg-22.htm>

Prise en compte du DP dans la construction de STI

Les connaissances en Instruction sont utilisées dans le domaine des STI depuis 25 ans et l'expérience en matière de construction des STI montre que ces connaissances sont souvent orientées vers un programme d'études ou un sujet d'apprentissage en particulier (Mizoguchi & Bourdeau, 2000). Cela a permis d'observer une faible congruence entre l'analyse des tâches et les stratégies d'apprentissage ou d'évaluation, ainsi que la faible prise en compte du contexte dans lequel évolue l'apprenant au détriment des aspects purement cognitifs. Les connaissances actuelles dans le domaine de l'Instruction sont plutôt fragmentaires et sont parfois utilisées plus pour servir les besoins de la conception technique que ceux de l'apprentissage. Des efforts récents au sein de la communauté de l'AIED (Artificial Intelligence and EDucation) ont été dirigés vers la construction de systèmes tuteurs intelligents en vue de leur amélioration et de leur facilitation (Nkambou, Frasson et Gauthier, 2003 ; Murray, 1996 ; Redfield, 1997). De plus, dans un article écrit en 1998 (Murray, 1998), Murray propose la prise en compte, ou même l'intégration de quatre composants additionnels : des outils pour le contenu, une stratégie d'instruction, un modèle de l'apprenant et une interface de conception.

Durant la dernière décennie, des travaux tenant compte de caractéristiques du DP (indépendant du domaine, basé sur des théories), ont été menés pour modéliser la connaissance contenue dans les STI (Tennyson et Barron, 1995). D'autres travaux ont également été réalisés durant les années 80 au laboratoire Amstrong des Forces Armées Américaines (USAF) pour construire un corpus de connaissances consensuel sur le Design Pédagogique, et pour automatiser le processus de DP. Ce projet dénommé AIDA (Advanced Instructional Design Advisor) a permis de comprendre la nécessité d'un ensemble de connaissances partagées. Sur cette lancée, d'autres projets ont par la suite suivi. On compte entre autre "ID Expert" de Merrill, "GUIDE" de Tennyson, "System Dynamics" à l'Université Bergen¹¹, "SAFARI" à l'Université de Montréal et "AGD" au laboratoire LICEF de la Télé-Université (Paquette & Girard, 1996). Les connaissances développées au niveau du DP ont permis de démarrer le projet d'un Environnement de Campus Virtuel orienté-objet au LICEF (Paquette et al., 1995). Les environnements intelligents de construction appuyant le développement de STI requièrent des bases

¹¹ <http://www.ii.uib.no/>

solides dans le domaine de la Science de l'Instruction. Ces bases doivent contenir un ensemble cohérent de concepts et de principes en vue de permettre la construction de produits de qualité. Ces environnements devraient permettre aux concepteurs de choisir entre une connaissance établie ou des développements plus récents. À ce propos, citons la proposition faite par Reigeluth de considérer les apprenants comme des co-concepteurs de leur formation. Cette proposition suggérerait que les "apprenants devraient avoir la capacité de demander que le système utilise des stratégies de formation particulières, de la même façon que le système était capable de décider quelles stratégies utiliser en fonction des entrées faites par l'apprenant" (Reigeluth, 1996).

3.2 Les scénarios, un des produits des systèmes auteurs

Dans le cadre du projet CIAO, nous avons fait usage de scénarios issus du DP afin de fournir un des services qui est la validation (service d'analyse). Le fichier de scénario est un fichier xml conforme au standard IMS-LD. Ce standard est établi par le consortium IMS¹², qui a pour mission de supporter l'adoption et l'utilisation de technologies universelles pour l'apprentissage. Les standards établis par l'IMS dans le cadre du Learning Design (LD) proviennent initialement de spécifications supportant l'utilisation d'une vaste gamme de pédagogies pour l'apprentissage en ligne¹³.

Le consortium IMS et les spécifications développées pour le Learning Design

Le nom formel pour IMS est IMS Global Learning Consortium (Consortium IMS pour l'apprentissage global), souvent décrit par le sigle IMS/GLC. À ses débuts en 1997, l'IMS portait le nom de "Instructional Management Systems project (IMS project)". Ce nom a par la suite été abandonné pour éviter des confusions possibles avec d'autres termes utilisés pour dénommer d'autres éléments ou concepts dans la communauté E-learning. Aujourd'hui, on dénomme le consortium par le terme "IMS". Les membres du consortium IMS proviennent de tous les secteurs de la communauté du E-learning. Cette

¹² <http://www.imsglobal.org/>

¹³ <http://www.imsglobal.org/learningdesign/index.html>

organisation fournit un forum neutre où les membres, bien qu'inspirés par des intérêts d'affaires compétitifs et de critères différents pour les prises de décisions, collaborent pour satisfaire aux exigences du monde réel en interopérabilité et en réutilisation des connaissances utilisées pour la construction de STI. Le consortium IMS développe et promeut l'adoption de spécifications techniques ouvertes pour des technologies d'apprentissage interopérables. Plusieurs des spécifications de l'IMS sont devenues des standards de facto au niveau mondial dans la création de produits et de services d'apprentissage. Les spécifications IMS et les publications associées sont mises gratuitement à la disposition du grand public.

Les spécifications IMS, définies comme intervenant dans le cadre d'un "apprentissage distribué", portent à la fois sur l'apprentissage en ligne et sur l'apprentissage "hors ligne" (off-line), qu'il soit effectué de façon synchrone (en temps réel) ou asynchrone. Les contextes d'apprentissage bénéficiant des spécifications IMS incluent aussi bien des environnements Internet spécifiques (comme les systèmes de gestion de cours basés sur Internet) que des situations d'apprentissage faisant intervenir des ressources électroniques qui ne sont pas utilisées en ligne (ressources d'apprentissage à partir d'un CD-ROM). Les apprenants peuvent se situer dans un environnement éducationnel traditionnel (salle de classe, université), dans une activité de formation d'entreprise ou gouvernementale, ou à la maison.

La spécification IMS pour le Learning Design est un langage qui recourt à une vaste gamme de stratégies pédagogiques pour l'apprentissage en ligne. Plutôt que de tenter de capturer les spécificités propres à différentes pédagogies, elle fournit un langage flexible et générique. Ce langage permet différents types de pédagogies. Cette approche est avantageuse compte tenu du fait qu'un seul ensemble d'outils pour la conception et la mise en œuvre de stratégies d'apprentissage est nécessaire pour supporter une vaste gamme de stratégies pédagogiques. À l'origine, le langage a été développé par l'Université Libre des Pays-Bas (Open University of the Netherlands – OUNL), après examen et comparaison minutieuse de plusieurs types d'approches pédagogiques et d'activités d'apprentissage associées. Plusieurs itérations dans le développement du langage ont donné un bon équilibre entre expressivité générale et expressivité pédagogique.

Processus de développement des spécifications IMS

La portée du travail effectué par l'IMS est très grande. Les besoins et conditions à satisfaire en vue d'établir les aspects critiques de l'interopérabilité sur les marchés de l'apprentissage sont rassemblés grâce à des rencontres, des groupes de discussion et d'autres sources. À partir de ces exigences, les membres de l'IMS développent des ébauches de caractéristiques relatives à la façon de construire les applications logicielles en vue de répondre aux exigences. Les spécifications sont développées pour supporter des besoins internationaux. Une fois que les spécifications sont finalisées à l'interne et qu'elles ont été validées par des tests d'interopérabilité (impliquant habituellement des membres contribuables et des participants du Réseau de Développeurs¹⁴), l'ébauche est approuvée par le comité technique de l'IMS et mise à la disposition du grand public. Par la suite, les spécifications sont remises aux groupes qui ont la responsabilité de faire accréditer les standards afin de contribuer à la reconnaissance et à l'adoption d'une base mondiale de standards techniques pour l'apprentissage distribué.

Fichier IMS-LD pour les scénarios pédagogiques

L'usage d'un fichier pour la réalisation du service de validation offert par CIAO a été fait en vue de pouvoir rendre CIAO indépendant de la plateforme utilisée pour la conception de scénarios d'apprentissage. Un avantage intéressant provient également du fait que le fichier analysé est construit par les SA en suivant des normes et standards de taille dans le E-learning, notamment ceux du Consortium IMS. L'indépendance vis-à-vis de la plateforme de construction était alors rendue possible compte tenu du fait que ce type de fichier, en tant que standard, peut être produit par différents SA et peut être ainsi être analysé sans recourir au SA qui a permis sa création.

Un modèle de fichier est fourni dans la section 4.3.5 (le service d'analyse).

¹⁴ <http://www.imsglobal.org/developersnetwork.html>

Nous venons de présenter les scénarios pédagogiques qui constituent un produit des systèmes auteurs. Dans la section 2.2, les systèmes auteurs ont été présentés et nous avons montré qu'ils disposent présentement de certaines limites en ce qui a trait à l'assistance qu'ils peuvent fournir au concepteur durant les activités de design pédagogique. Nous avons également présenté, dans la section 3.1, les ontologies des théories et le rôle qu'elles jouent dans la modélisation des connaissances. Fort de tout cela, nous présenterons dans la section suivante le système que nous développons, CIAO, en vue de répondre aux limites exposées par les systèmes auteurs, et servant de pont entre les théories, le système auteur et le concepteur.

Le système CIAO est développé pour servir de support aux experts (concepteur pédagogique) au moment de la création de scénarios pédagogiques. Il exploite les ontologies à l'intérieur desquelles les théories sont formalisées. Cette formalisation permet ainsi de donner accès à ces théories lors de la conception des scénarios pédagogiques. CIAO permet de faire un pont entre le concepteur, le système auteur et les théories, dans un processus de création de scénarios pédagogiques qui soient valides par rapport à une ou à plusieurs théories. Ce pont est nécessaire car bénéfique au concepteur pour la construction de ces scénarios, mais surtout, il permet à l'agent CIAO de procurer au concepteur, une assistance profonde durant la conception.

La section suivante présente plus en détail le système CIAO.

3.3 CIAO, un pont entre l'ontologie et le système auteur

3.3.1. CIAO et les ontologies de théories

Le système CIAO exploite la base de connaissances constituée des ontologies des théories, de l'instruction, et du design pédagogique en vue de fournir des services au concepteur de scénarios pédagogiques. CIAO sert essentiellement d'interface entre la base ontologique et le SA. Dans notre cas, l'interface sera assurée entre l'ontologie et le fichier de scénario, qui correspond au produit fourni pour le SA. CIAO permettra au

concepteur de naviguer à travers l'ontologie des théories afin d'en visualiser la structure et d'examiner les éléments qu'il juge pertinents.

Bien souvent, les concepteurs produisent leurs environnements d'apprentissage (ou STI) en se basant sur leur intuition, leur expérience, leur expertise ou d'autres facteurs personnels. Il arrive qu'ils se basent sur des théories, mais celles-ci sont dans la plupart des cas installées à même le SA utilisé pour la construction du STI. Ainsi, les SA, et donc les STI, se commettent déjà en matière de théories. Un des avantages de CIAO sera de placer les théories à l'extérieur du système (dans l'ontologie), ce qui permet de faire abstraction des théories dans un premier temps et encouragerait le développement de STI dans un contexte de liberté par rapport aux théories de référence. CIAO exploitera également la base ontologique développée par Valéry Psyché (Psyché, 2004), et permettrait alors aux concepteurs d'accéder à ces théories pour prendre des décisions relatives à la conception. CIAO permettra ainsi aux concepteurs de baser leurs connaissances heuristiques sur des connaissances théoriques en vue de supporter la conception de STI, et de pouvoir ainsi mieux expliquer et justifier les décisions prises lors de la conception.

3.3.2. Architecture conceptuelle

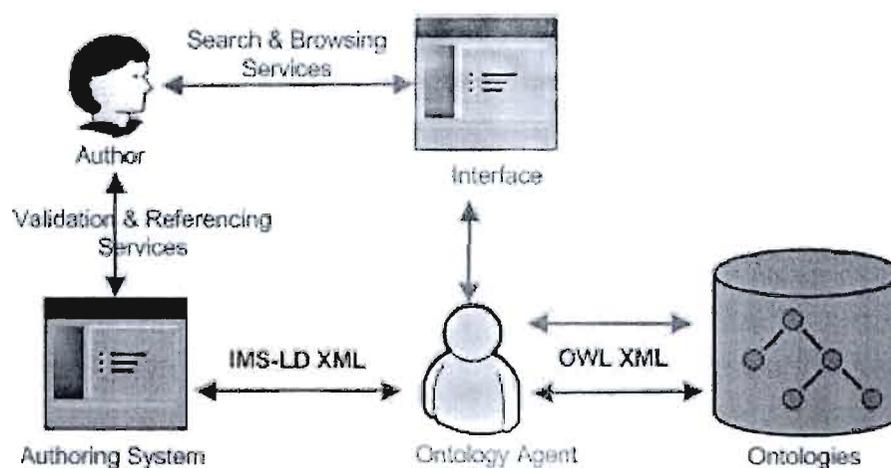


Figure 3.1 : Interactions entre CIAO et le Système Auteur.

La figure 3.1 (Psyché, Bourdeau, Nkambou et Mizoguchi, 2005) illustre l'interaction entre le système CIAO et le SA qui produit les fichiers de scénarios pédagogiques. Sur cette figure, CIAO est représenté par l'agent d'ontologie (Ontology Agent) et offre une interface.

À partir de CIAO, nous avons trois directions dans lesquelles nous pouvons nous déplacer sur le graphique. Une de ces directions indique que CIAO offre une interface à l'utilisateur ou à l'auteur des activités de DP. Cette interface lui permet ainsi de bénéficier de services de recherche et d'exploration à travers l'ontologie. Le service de recherche s'effectue grâce à des requêtes définies en langage ScRQL (Voir section 4.3.4). Il permet au concepteur d'accéder aux informations contenues dans la base de connaissances. Le service d'exploration permet au concepteur d'accéder à l'ensemble des informations contenues dans l'ontologie et d'y naviguer à sa guise, en fonction de ses besoins. Ce service est présentement offert sous trois versions différentes (Voir section 4.3.3). Une autre interaction dans laquelle intervient le système CIAO sur la figure 3.1 est celle avec le SA. Cette interaction spécifie que CIAO communique principalement avec le SA en échangeant des données au format XML IMS-LD¹⁵. Ces données, ou plus précisément les données provenant du SA et dont CIAO se sert, sont rassemblées dans un fichier de scénario pédagogique (Voir section 3.2) décrivant les différentes entités qui interviennent dans la représentation d'une activité pédagogique.

La troisième interaction dans laquelle intervient le système CIAO est celle avec l'ontologie des théories du DP. Cette interaction permet à CIAO de puiser dans les informations que contient l'ontologie afin de répondre aux demandes du concepteur. Elle permet aussi à l'administrateur des données de modifier le contenu de l'ontologie à exploiter pour les différents services directement à partir de l'interface de CIAO (Voir section 4.3.7). Les données qui transitent à ce niveau sont des données au format OWL (Web Ontology Language). Le OWL est un vocabulaire basé sur RDF, et permet de spécifier ce qui peut être compris : il fournit un langage pour définir des ontologies Web structurées. Le RDF (Resource Description Framework) est au centre de la plupart des travaux du Web Sémantique. C'est un vocabulaire XML pour décrire des ressources, ou des relations entre ressources, en leur affectant des métadonnées. Les machines peuvent

¹⁵ Données XML respectant les normes pour le standard IMS-LD (Voir section 3.2)

dès lors classer ces ressources et modéliser les liens existant entre elles. RDF permet donc de définir les règles qui relient les informations entre elles.

Le langage OWL est conçu pour être utilisé par des applications ayant besoin de traiter le contenu de l'information plutôt que d'en faire une simple présentation. OWL facilite l'interprétation du contenu Web par des ordinateurs, d'une meilleure façon que le permet tant XML, RDF et RDF-S¹⁶ (RDF-Schema), en fournissant un vocabulaire additionnel couplé à une sémantique formelle. OWL offre un langage pouvant s'utiliser pour décrire les classes et leurs relations entre elles qui sont inhérentes aux documents Web et aux applications.

Le langage OWL est défini en trois sous-langages de plus en plus expressifs, chacun étant une extension du précédent :

- **OWL Lite** : pour les utilisateurs ayant besoin d'une classification hiérarchique et d'un ensemble simplifié de contraintes pouvant être exprimées sur les données. Par exemple, bien que OWL Lite supporte les contraintes de cardinalité, il ne permet que des valeurs de cardinalités de 0 et 1 au niveau des relations.
- **OWL DL** : pour les utilisateurs voulant bénéficier d'une expressivité maximum sans pour autant perdre la complétude au niveau du traitement des données et la capacité de décision effectuée par les systèmes utilisant le raisonnement. OWL DL inclut l'ensemble des éléments du langage OWL Lite en plus de restrictions telles que celles portant sur la séparation des types (une classe ne peut pas être à la fois un individu ou une propriété, ou encore une propriété ne peut pas être à la fois un individu et une classe).
- **OWL Full** : destiné aux utilisateurs voulant bénéficier de l'expressivité maximum et de la liberté syntaxique du RDF, mais sans garanties de traitement. Par exemple, en OWL Full, une classe peut être traitée simultanément comme une collection d'individus et comme un individu à part entière.

La figure 3.1 décrit également un lien existant entre le concepteur et le SA. Ce lien traduit le fait que le concepteur utilise le service de validation (analyse) et de référence offert par

¹⁶ <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>

CIAO, mais à travers le système auteur. Le concepteur peut bénéficier directement de ce service à partir de l'interface de CIAO car celui-ci a été rendu indépendant du SA utilisé pour la création des fichiers de scénarios pédagogiques. Ce sont finalement ces fichiers qui sont analysés et validés.

La section suivante décrit quelques fonctionnalités de CIAO pour les différentes options associées à la recherche par requête des informations relatives aux classes de l'ontologie des théories. La section suivante présente le cas d'utilisation associé à l'action d'écriture consistant à modifier le contenu de l'ontologie dont se sert CIAO.

D'autres cas d'utilisation sont fournis en annexes.

3.3.3. Modélisation des services du système CIAO

1) Cas d'utilisations

Cas d'utilisation 1 "Chercher des informations à partir de la liste de requêtes prédéfinies"

Tableau 3.1: Cas d'utilisation 1

| | |
|--|--|
| <p>Acteur principal : L'utilisateur du système CIAO (concepteur d'activités pédagogiques)</p> <p>Parties prenantes et intérêts :</p> <p>L'utilisateur du système : il veut voir des informations contenues dans l'ontologie en faisant un choix dans une liste de possibilités de requêtes.</p> <p>Le système CIAO : il veut offrir à l'utilisateur les informations demandées sur la recherche.</p> <p>Préconditions :</p> <p>Le contenu de l'ontologie n'a pas été altéré suite à des</p> | |
|--|--|

| | | | |
|--|--|----------------------------|--|
| modifications effectuées par l'administrateur de l'application | | | |
| <p>Garanties de succès (postconditions) :</p> <p>Le choix de l'utilisateur est enregistré. Les données sont correctement affichées, selon le choix effectué par l'utilisateur.</p> <p>Les données sont affichées dans un format choisi par l'utilisateur.</p> | | | |
| <p>Scénario principal (succès) :</p> <p>Action Acteur (ou intention)</p> | | Responsabilités du système | |
| 1. | L'utilisateur pointe l'option de recherche d'informations dans l'ontologie des théories. | 2. | Le Système affiche les trois possibilités de recherche dans l'ontologie. |
| 3. | L'utilisateur choisit l'option de recherche à partir d'une liste de requêtes prédéfinies. | 4. | Le Système présente la liste des requêtes à l'utilisateur. |
| 5. | L'utilisateur fait un choix de données qu'il souhaite voir à partir de la liste présentée par le Système. | 7. | Le Système affiche les résultats à l'utilisateur dans un format de données HTML, ce qui permet à l'utilisateur de pouvoir naviguer dans l'ontologie en utilisant les liens hypertextes présents. |
| 6. | L'utilisateur actionne la commande pour lancer la recherche. | | |
| <p>Extensions (ou scénarios alternatifs) :</p> | | | |
| 6a. | L'utilisateur sélectionne un autre format pour l'affichage des données présentées par le Système, notamment le format XML. | | |
| 1. | L'utilisateur actionne la commande pour lancer la recherche. | 2. | Le Système affiche les résultats à l'utilisateur dans un format de données XML. |
| 6b. | L'utilisateur sélectionne le format de données RDF pour l'affiche des données. | | |
| 1. | L'utilisateur actionne la commande pour lancer la recherche. | 2. | Le Système demande à l'utilisateur de confirmer la sauvegarde des données au format RDF dans un fichier. |
| 3. | L'utilisateur confirme la sauvegarde des données. | 4. | Le Système ouvre une liste d'applications installées sur le poste de l'utilisateur et lui demande d'en sélectionner une pour |

| | | | | |
|--------|----|---|----|--|
| 6b.3.a | 5. | L'utilisateur choisit une application pour l'affichage des données. | 6. | l'affichage des données sauvegardées. Le Système affiche les données au format RDF en utilisant l'application sélectionnée par l'utilisateur. |
| | 1. | L'utilisateur annule la sauvegarde des données dans un fichier. | 2. | Le Système revient à l'écran d'affichage de la liste de requêtes prédéfinies. |

Cas d'utilisation 2 "Chercher des informations en construisant partiellement une requête"

Tableau 3.2: Cas d'utilisation 2

| | |
|--|--|
| <p>Acteur principal : L'utilisateur du système CIAO (concepteur d'activités pédagogiques)</p> <p>Parties prenantes et intérêts : L'utilisateur du système : il veut voir des informations en précisant les données à inclure dans la requête, et en se faisant aider par le système pour l'écriture de la requête. Le système CIAO : il veut offrir à l'utilisateur les informations demandées sur la recherche.</p> <p>Préconditions : Le contenu de l'ontologie n'a pas été altéré suite à des modifications effectuées par l'administrateur de l'application.</p> <p>Garanties de succès (postconditions) : Le choix de l'utilisateur est enregistré. Les données sont correctement affichées, selon le choix effectué par l'utilisateur. Les données sont affichées dans un format choisi par l'utilisateur.</p> <p>Scénario principal (succès) :</p> | |
|--|--|

| Action Acteur (ou intention) | Responsabilités du système |
|--|--|
| <ol style="list-style-type: none"> 1. L'utilisateur pointe l'option de recherche d'informations dans l'ontologie des théories. 3. L'utilisateur choisit l'option de recherche utilisant la construction partielle d'une requête. 5. L'utilisateur fait un choix de données qu'il souhaite voir comme résultat de la requête. 6. L'utilisateur actionne la commande pour lancer la recherche. | <ol style="list-style-type: none"> 2. Le Système affiche les trois possibilités de recherche dans l'ontologie. 4. Le Système présente les éléments dont peut se servir l'utilisateur pour construire sa requête. 7. Le Système affiche les résultats à l'utilisateur dans un format de données HTML, ce qui permet à l'utilisateur de pouvoir naviguer dans l'ontologie en utilisant les liens hypertextes présents. |
| Extensions (ou scénarios alternatifs) : | |
| <ol style="list-style-type: none"> 6a. L'utilisateur sélectionne un autre format pour l'affichage des données présentées par le Système, notamment le format XML. <ol style="list-style-type: none"> 1. L'utilisateur actionne la commande pour lancer la recherche. | <ol style="list-style-type: none"> 2. Le Système affiche les résultats à l'utilisateur dans un format de données XML. |
| <ol style="list-style-type: none"> 6b. L'utilisateur sélectionne le format de données RDF pour l'affiche des données. <ol style="list-style-type: none"> 1. L'utilisateur actionne la commande pour lancer la recherche. 3. L'utilisateur confirme la sauvegarde des données. 5. L'utilisateur choisit une application pour l'affichage des données. | <ol style="list-style-type: none"> 2. Le Système demande à l'utilisateur de confirmer la sauvegarde des données au format RDF dans un fichier. 4. Le Système ouvre une liste d'applications installées sur le poste de l'utilisateur et lui demande d'en sélectionner une pour l'affichage des données sauvegardées. 6. Le Système affiche les données au format RDF en utilisant l'application sélectionnée par l'utilisateur. |
| <ol style="list-style-type: none"> 6b.3.a <ol style="list-style-type: none"> 1. L'utilisateur annule la sauvegarde des données dans un fichier. | <ol style="list-style-type: none"> 2. Le Système revient à l'écran d'affichage de la liste de requêtes prédéfinies. |

| | | |
|-----|---|--|
| 6d. | <ol style="list-style-type: none"> 1. L'utilisateur actionne la commande lui permettant de voir le code généré par le Système suite aux choix qu'il a fait. 3. L'utilisateur actionne la commande pour lancer la recherche. | <ol style="list-style-type: none"> 2. Le Système affiche le code source généré par les choix effectués par l'utilisateur. 4. Le Système affiche les résultats à l'utilisateur dans un format de données XML. |
|-----|---|--|

Dans le cas d'utilisation qui précède, l'utilisateur peut également à tout moment décider le masquer le code source généré par le système suite aux choix qu'il a effectués.

Cas d'utilisation 3 "Chercher des informations en créant le code pour une requête"

Tableau 3.3: Cas d'utilisation 3

| | |
|---|-----------------------------------|
| <p>Acteur principal : L'utilisateur du système CIAO (qui est un concepteur d'activités pédagogiques)</p> <p>Parties prenantes et intérêts :</p> <p>L'utilisateur du système : il veut voir des informations en écrivant la requête que le Système utilisera. Généralement, il s'agit d'un expert en langage SeRQL.</p> <p>Le système CIAO : il veut offrir à l'utilisateur les informations demandées sur la recherche.</p> <p>Préconditions :</p> <p>Le contenu de l'ontologie n'a pas été altéré suite à des modifications effectuées par l'administrateur de l'application</p> <p>Garanties de succès (postconditions) :</p> <p>Le choix de l'utilisateur est enregistré. Les données sont correctement affichées, selon le choix effectué par l'utilisateur.</p> <p>Les données sont affichées dans un format choisi par l'utilisateur.</p> | |
| <p>Scénario principal (succès) :</p> <p>Action Acteur (ou intention)</p> | <p>Responsabilités du système</p> |

| | | | |
|--|--|---|--|
| <p>1.</p> <p>3.</p> <p>5.</p> <p>6.</p> | <p>L'utilisateur pointe l'option de recherche d'informations dans l'ontologie des théories.</p> <p>L'utilisateur choisit l'option de recherche lui permettant de construire complètement la requête par lui-même.</p> <p>L'utilisateur saisit sa requête.</p> <p>L'utilisateur actionne la commande pour lancer la recherche.</p> | <p>2.</p> <p>4.</p> <p>7.</p> | <p>Le Système affiche les trois possibilités de recherche dans l'ontologie.</p> <p>Le Système affiche l'interface de saisie à l'utilisateur, avec un modèle de requête opérationnelle, afin de lui donner une sorte d'aide à la saisie.</p> <p>Le Système affiche les résultats à l'utilisateur dans un format de données HTML, ce qui permet à l'utilisateur de pouvoir naviguer dans l'ontologie en utilisant les liens hypertextes présents.</p> |
| <p>Extensions (ou scénarios alternatifs) :</p> | | | |
| <p>6a.</p> <p>1.</p> <p>6b.</p> <p>1.</p> <p>3.</p> <p>5.</p> <p>6b.3.a</p> <p>1.</p> <p>7a.</p> | <p>L'utilisateur sélectionne un autre format pour l'affichage des données présentées par le Système, notamment le format XML.</p> <p>L'utilisateur actionne la commande pour lancer la recherche.</p> <p>L'utilisateur sélectionne le format de données RDF pour l'affiche des données.</p> <p>L'utilisateur actionne la commande pour lancer la recherche.</p> <p>L'utilisateur confirme la sauvegarde des données.</p> <p>L'utilisateur choisit une application pour l'affichage des données.</p> <p>L'utilisateur annule la sauvegarde des données dans un fichier.</p> | <p>2.</p> <p>2.</p> <p>4.</p> <p>6.</p> <p>2.</p> | <p>Le Système affiche les résultats à l'utilisateur dans un format de données XML.</p> <p>Le Système demande à l'utilisateur de confirmer la sauvegarde des données au format RDF dans un fichier.</p> <p>Le Système ouvre une liste d'applications installées sur le poste de l'utilisateur et lui demande d'en sélectionner une pour l'affichage des données sauvegardées.</p> <p>Le Système affiche les données au format RDF en utilisant l'application sélectionnée par l'utilisateur.</p> <p>Le Système revient à l'écran d'affichage de la liste de requêtes prédéfinies.</p> |
| | <p>* la requête saisie par l'utilisateur est syntaxiquement incorrecte.</p> | | |

| | | | |
|--|--|--|--|
| | | | <ol style="list-style-type: none"> 1. Le Système affiche un rapport d'erreur indiquant à l'utilisateur le type d'erreur, les informations que le Système s'attendait à recevoir, et la chaîne de caractères qui a produit l'erreur en question. 2. Le Système affiche simultanément l'interface de saisie à l'utilisateur. |
|--|--|--|--|

Cas d'utilisation 4 "Modifier le contenu de l'ontologie dont se sert CIAO"

Tableau 3.4: Cas d'utilisation 4

| | | | |
|--|--|----------------------------|---|
| <p>Acteur principal : L'administrateur du système CIAO (qui peut être un concepteur d'activités pédagogiques)</p> <p>Parties prenantes et intérêts : L'utilisateur du système : il veut mettre à jour le contenu de l'ontologie dont se sert le Système pour les différentes opérations.. Le système CIAO : il veut permettre à l'utilisateur d'effectuer la mise à jour nécessaire.</p> <p>Préconditions : Aucune.</p> <p>Garanties de succès (postconditions) : Les modifications de l'utilisateur sont enregistrées. L'ontologie des théories dont se sert le Système est mise à jour..</p> | | | |
| <p>Scénario principal (succès) : Action Acteur (ou intention)</p> | | Responsabilités du système | |
| 1. | L'utilisateur sélectionne l'option lui permettant de s'identifier et de s'authentifier afin d'obtenir les droits d'écriture nécessaires. | 2. | Le Système affiche l'interface d'identification/authentification à l'utilisateur. |
| 3. | L'utilisateur inscrit son identifiant et son mot de passe. | | |

| | | | |
|--|--|--|--|
| <p>4.</p> <p>6.</p> <p>8.</p> <p>10.</p> <p>11.</p> <p>14.</p> | <p>L'utilisateur valide les informations en actionnant la commande lui permettant de se connecter.</p> <p>L'utilisateur choisit la partie de l'ontologie qu'il souhaite explorer, à savoir les classes ou les instances.</p> <p>L'utilisateur choisit l'option lui permettant d'ajouter des données au contenu actuel de l'ontologie, à partir d'un fichier de données.</p> <p>L'utilisateur spécifie le fichier à utiliser.</p> <p>L'utilisateur actionne la commande de mise à jour de l'ontologie.</p> <p>L'utilisateur actionne la commande de retour à l'écran principal.</p> | <p>5.</p> <p>7.</p> <p>9.</p> <p>12.</p> <p>13.</p> <p>15.</p> | <p>Le Système indique à l'utilisateur qu'il vient d'être connecté et l'envoie à la fenêtre principale à partir de laquelle il choisira l'action qu'il souhaite exécuter.</p> <p>Le Système affiche l'écran correspondant au choix fait par l'utilisateur.</p> <p>Le programme affiche l'écran permettant à l'utilisateur de définir l'emplacement du fichier à partir duquel les données seront extraites.</p> <p>Le Système met à jour l'ontologie des théories.</p> <p>Le Système indique à l'utilisateur le nombre de nouveaux enregistrements qui viennent d'être rajoutés.</p> <p>Le Système affiche l'écran principal à l'utilisateur.</p> |
| Extensions (ou scénarios alternatifs) : | | | |
| <p>3a.</p> <p>5a.</p> <p>6b.</p> | <p>L'utilisateur annule le processus d'authentification</p> <p>1. L'utilisateur actionne la commande d'annulation du processus d'authentification</p> <p>Les informations d'identifications saisies par l'utilisateur sont incorrectes.</p> <p>L'utilisateur rentre des informations d'identification correctes mais ne dispose pas des droits</p> | <p>2.</p> <p>1.</p> <p>2.</p> | <p>Le Système renvoie l'utilisateur à l'écran principal de l'application.</p> <p>Le Système indique à l'utilisateur que son identifiant ou son mot de passe est incorrect.</p> <p>Le Système affiche simultanément deux options permettant à l'utilisateur de refaire un autre choix ou d'annuler le processus d'identification/authentification.</p> |

| | | |
|-----|--|--|
| 12a | <p>d'écriture sur l'ontologie.</p> <p>L'ontologie ne peut être mise à jour car elle contient des données erronées.</p> | <ol style="list-style-type: none"> 1. Le Système indique à l'utilisateur qu'il vient d'être correctement identifié. 2. Le système affiche les fonctions de CIAO et indique à l'utilisateur qu'il ne dispose pas de droits de modifications sur l'ontologie des théories. 1. Le système indique à l'utilisateur la nature de l'erreur qui vient d'être rencontrée et interrompt le processus de mise à jour. |
|-----|--|--|

2) Diagrammes de séquence système

Les cas d'utilisation présentés dans la section précédente offraient une description de l'interaction des acteurs externes avec le système CIAO. Nous pouvons donc y voir les événements générés par les acteurs, et qui affectent le système, de même que les réponses générées par ce dernier. Les opérations demandées par les acteurs constituent un élément important dans la compréhension du système. Cette section présente des diagrammes de séquence illustrant les interactions présentées à la section précédente et les opérations qu'elles déclenchent. Les diagrammes de séquences présentés sont produits pour les scénarios principaux de chacun des cas d'utilisation de la précédente section.

Diagramme de séquence pour le cas d'utilisation 1 "Chercher des informations à partir de la liste de requêtes prédéfinies "

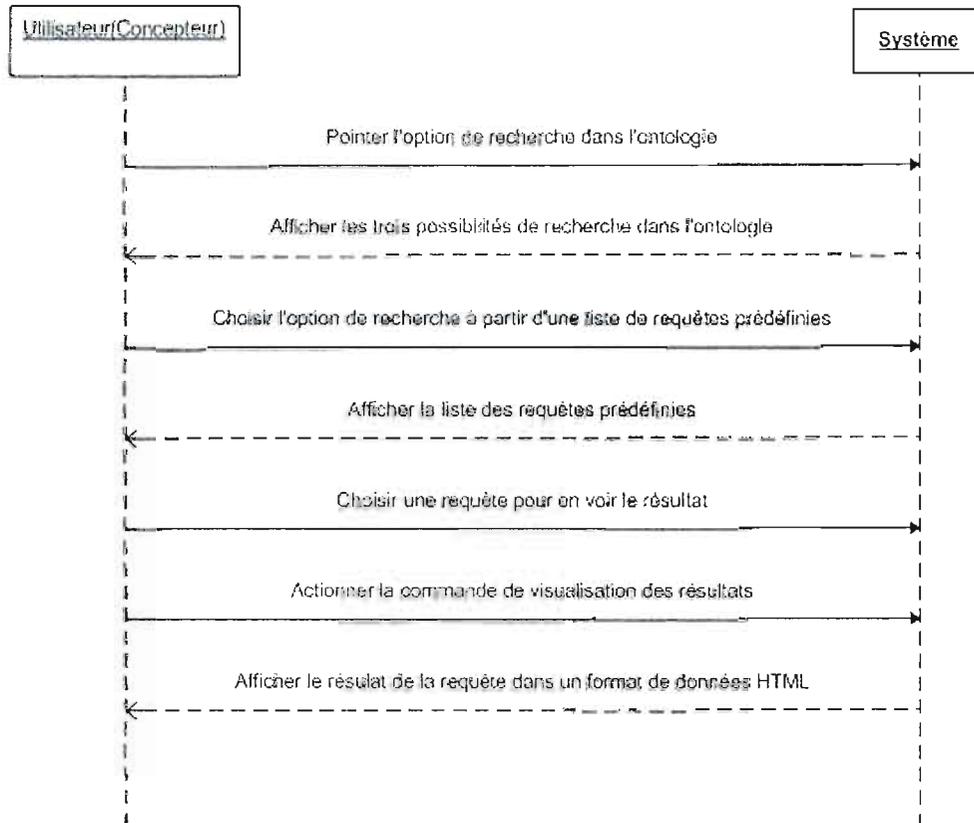


Figure 3.2: Diagramme de séquence du cas d'utilisation 1.

Diagramme de séquence pour le cas d'utilisation 2 "Chercher des informations en construisant partiellement une requête"

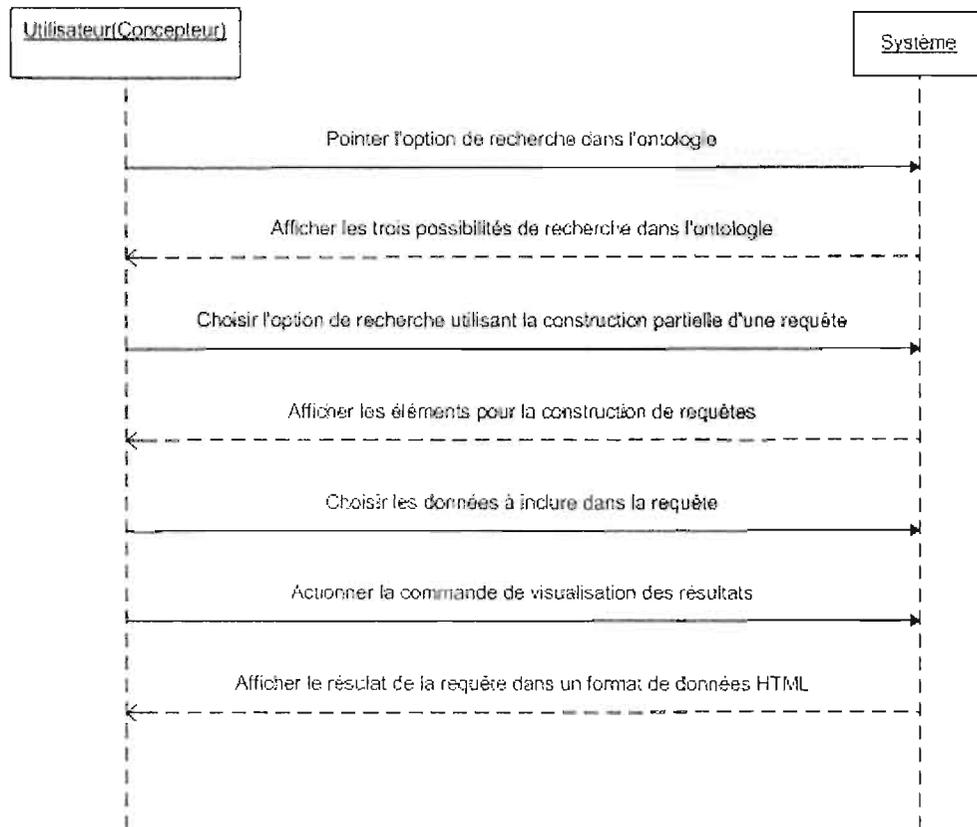


Figure 3.3: Diagramme de séquence du cas d'utilisation 2.

Diagramme de séquence pour le cas d'utilisation 3 "Chercher des informations en créant complètement le code pour une requête"

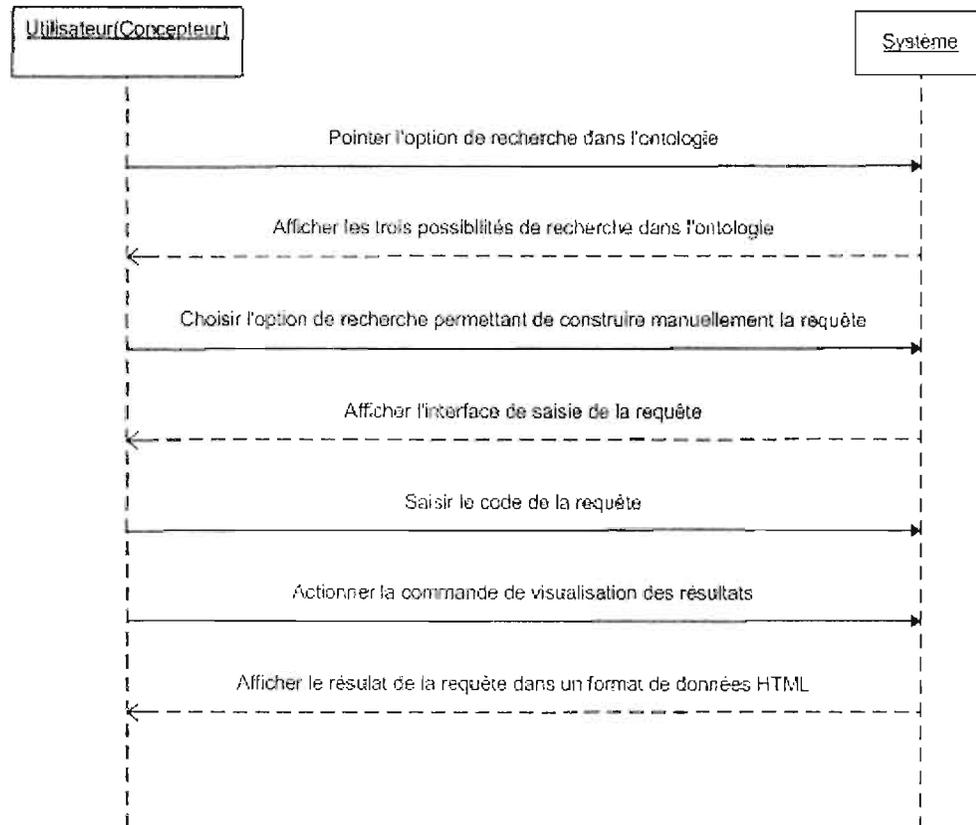


Figure 3.4: Diagramme de séquence du cas d'utilisation 3.

Diagramme de séquence pour le cas d'utilisation 4 "Modifier le contenu de l'ontologie dont se sert CIAO "

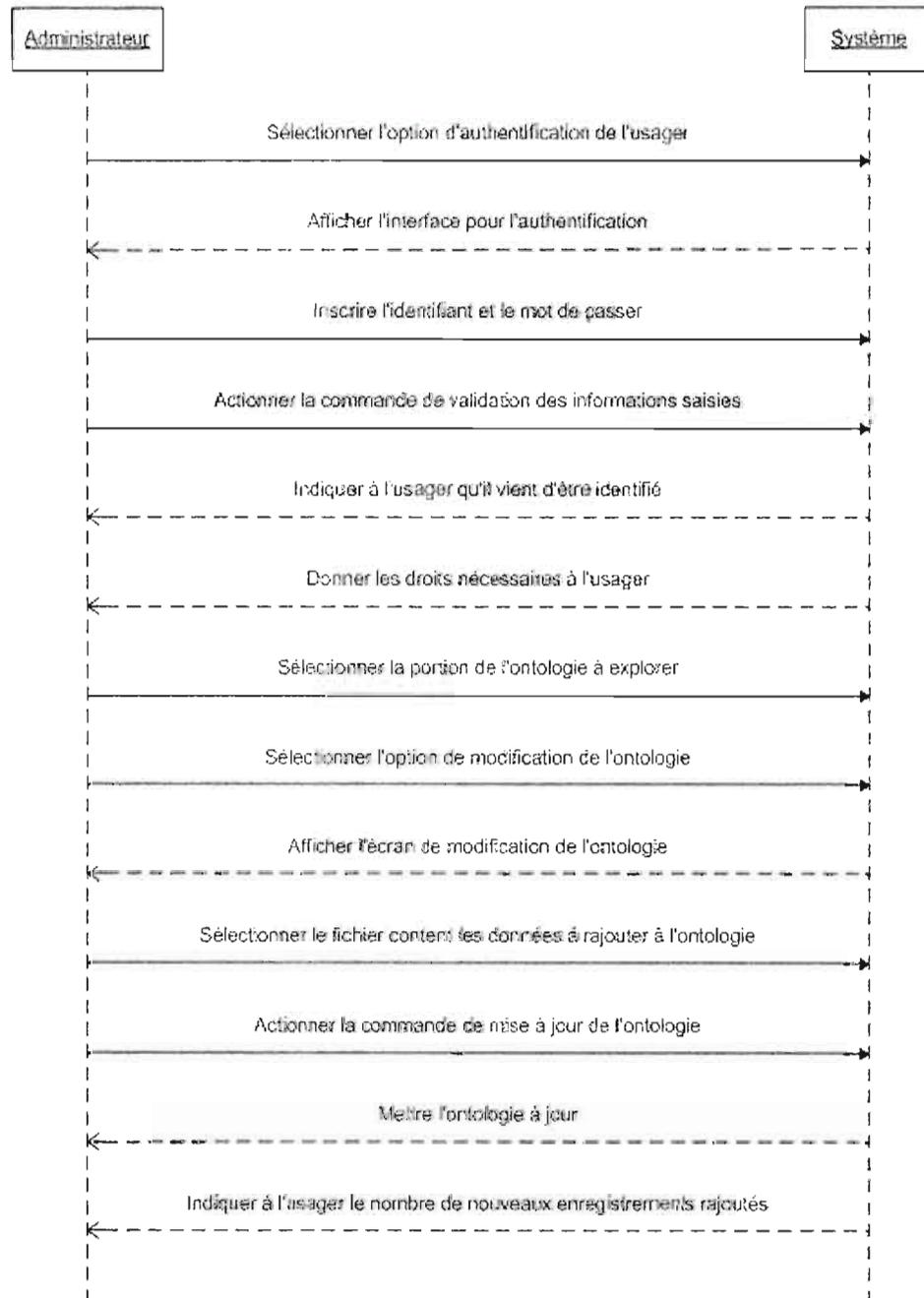


Figure 3.5: Diagramme de séquence du cas d'utilisation 4.

3.3.4. Rappel des services offerts par CIAO

Le Système offre quatre principaux services: le service d'exploration de l'ontologie des théories, le service de requêtes, le service d'analyse et le service d'exportation.

Le service d'exploration permet à l'utilisateur d'obtenir différentes visions de l'ontologie des théories, et de pouvoir naviguer à travers celle-ci grâce à des liens hypertextes. Le service de requêtes permet à l'utilisateur d'afficher différentes informations provenant de l'ontologie en construisant des requêtes. Il dispose de trois versions. La première est destinée à un usager débutant et lui permet de choisir parmi une liste de requêtes prédéfinies. La seconde est destinée à un usager intermédiaire et l'aide dans la construction d'une requête. La troisième est destinée à un usager expert et lui offre la possibilité d'écrire sa requête. Le service d'analyse permet à un usager de faire valider/analyser un scénario pédagogique afin d'y déceler des erreurs syntaxiques. Il offre également une analyse sémantique dans laquelle le système effectue des recommandations à l'usager selon le contenu de son scénario pédagogique. Le service d'exportation permet à l'usager d'exporter le contenu de l'ontologie dans un format de fichier particulier et de le sauvegarder sur disque. Le format par défaut est le RDF (Resource Description Framework).

Ces services sont présentés dans le prochain chapitre.

CHAPITRE IV

LES CHOIX TECHNOLOGIQUES ET L'IMPLÉMENTATION DE CIAO

Grâce à l'utilisation d'ontologies, nous pouvons rassembler une information compréhensible et interprétable par les ordinateurs. De plus, l'intégration du Web sémantique aux différentes activités effectuées sur la toile Internet a été réalisée. Ce chapitre traite de l'exploitation du Web sémantique à l'intérieur du projet CIAO. Nous y discutons des choix technologiques effectués et présentons des éléments d'évaluation du système.

4.1 L'exploitation du Web sémantique

4.1.1. Origines et perspectives du Web sémantique

Durant des années, le Consortium¹⁷ du World Wide Web a travaillé à la mise en place du réseau mondial en élaborant des normes axées sur la présentation des contenus et leur proximité symbolique par les liens hypertextes. Ces normes, bien qu'elles ne soient que

¹⁷ <http://www.w3.org/>

partiellement respectées, ont rapidement permis d'accéder de façon performante à une quantité d'informations toujours croissante. Cette croissance a mis en évidence un déficit de maîtrise des contenus en raison de leur volume. Le World Wide Web est de ce fait autant la base que la raison du Web sémantique.

Parler de Web sémantique, c'est admettre l'existence d'un potentiel prometteur centré sur les possibilités de maîtrise des contenus. Le mot *sémantique* souligne qu'étant donné le caractère textuel des documents, les progrès du World Wide Web doivent nécessairement passer par un affranchissement et une maîtrise des contraintes des langues ou de leurs usages. Aussi, plusieurs disciplines ayant acquis une longue expérience de ces problèmes sont-elles mobilisées dans la poursuite de cet objectif.

Actuellement, presque tout le contenu du Web est destiné à être lu, et non pas à être manipulé de façon intelligente par des programmes informatiques, car les ordinateurs n'ont en général pas de méthode fiable pour s'occuper de sémantique. Le Web sémantique utilisera la structure pour donner du sens au contenu des pages Web, en créant un environnement où les agents logiciels pourront réaliser rapidement des tâches compliquées pour les utilisateurs en parcourant les pages (Tim Berners, Hendler et Lassila, 2001). Une telle activité sera rendue possible par l'encodage de la sémantique des informations à l'intérieur des pages Web au moment de leur construction et de leur mise en forme.

À ce jour, le Web s'est développé plus rapidement en tant que médium pour des documents destinés à des gens que comme ensemble de données et d'informations pouvant être traitées de manière automatique. Pour que le Web sémantique fonctionne, des collections structurées d'informations et des ensembles de règles d'inférence utilisables pour parvenir à un raisonnement automatisé doivent être mis à la disposition des ordinateurs. De tels systèmes de représentation et d'accès à la connaissance ont été étudiés par les chercheurs en Intelligence Artificielle (IA), bien avant le développement du Web.

Avec le Web sémantique, les chercheurs construisent un langage de règles aussi significatif que nécessaire pour permettre au Web de raisonner. Le défi du Web sémantique est de fournir un langage qui exprimerait autant les données auxquelles nous avons accès par le Web classique, que les règles pour raisonner sur ces données. Deux importantes technologies déjà existantes nous y aident. Ce sont le eXtensible Markup Language¹⁸ (XML) et le Resource Description Framework¹⁹ (RDF). Le Web sémantique peut donc être vu comme le Web de demain, en tant que vaste espace permettant d'échanger des ressources et capables d'exploiter, d'une façon qualitativement supérieure, de grands volumes d'informations et de services. Ce Web devrait alors décharger les utilisateurs d'une bonne partie de leurs tâches de recherche, de construction et de combinaison des résultats, grâce aux capacités des machines à accéder aux contenus des ressources et à raisonner sur ceux-ci.

XML

Le XML, grâce à sa faculté de représenter des documents et des données très diverses d'une manière uniforme et "sérialisable" (sous forme d'une séquence de caractères), est devenu le modèle de données standard pour la publication et l'échange d'informations sur le Web. Cependant, il ne répond que partiellement aux problèmes rencontrés quand on veut partager des informations sur le Web. Cette limite n'est pas d'ordre structurel ou syntaxique, mais a trait à la description du *sens* de l'information représentée.

RDF

Le langage RDF utilise une syntaxe XML pour représenter de l'information sur le Web. Il permet de décrire des ressources et les relations existant entre ces ressources. Grâce au RDF, il est possible d'annoter des documents non structurés et de fournir également une interface pour des applications et des documents structurés. Se faisant, le RDF permet de faire interagir des applications qui échangent de l'information sur le Web, qu'elle soit formalisée et structurée ou non. RDF décrit les ressources et les relations qui les lient en

¹⁸ <http://www.w3.org/XML/>

¹⁹ <http://www.w3.org/RDF/>

utilisant un triplet sous la forme {sujet, objet, prédicat}, un peu comme la langue française utilise la notation {sujet, verbe, complément}. On parle alors de *proposition RDF*. Par exemple, le sujet peut être un document à commenter, l'objet, une propriété de ce document comme son titre, et le prédicat, la valeur de cette propriété. Un document RDF correspond alors à un graphe orienté et étiqueté. Chaque triplet correspond alors à un arc orienté dont le label est le prédicat, le nœud source est le sujet et le nœud cible est l'objet.

Chaque élément du triplet peut être un URI, un littéral ou une variable. L'URI (Uniform Resource Identifier) est une technologie du Web pour le nommage et l'adressage de ressources. Les URI correspondent ainsi à de courtes chaînes de caractères qui identifient les ressources sur le Web. Ils peuvent ainsi identifier des documents, des images, des services, des adresses électroniques, ou autres.

RDF n'est pas en tant que tel un langage XML mais plutôt une structure de données constituées de nœuds, permettant d'aller d'une information à une autre. On pourrait faire du RDF sous d'autres formes²⁰. La sérialisation XML du RDF n'est qu'une des représentations possibles et est souvent dénommée par RDF/XML. Ce mode de représentation de l'information correspond à la syntaxe créée par le Consortium W3C pour l'utilisation du RDF au sein du Web Sémantique.

Le modèle RDF dépend d'un certain formalisme qui assure son efficacité en toute circonstance, mais qui le rend un peu "lourd", car il ne peut se permettre d'omettre un des éléments du triplet. Ce formalisme est nécessaire pour l'analyse automatique et sans ambiguïté des propositions. Grâce à RDF, il devient possible d'écrire des programmes informatiques capables d'opérer des traitements sur des connaissances : indexation, classement, diffusion, formatage, comparaison, inférence, etc. Il convient toutefois de préciser que RDF, bien que disposant de nombreuses fonctionnalités, n'est pas un langage de programmation et ne possède pas une syntaxe unique, mais peut au contraire se traduire sous plusieurs formes. RDF n'est également pas un format de données, et n'est

²⁰ <http://xmlfr.org/documentations/tutoriels/041015-0001>

pas fait pour présenter des données, car il constitue quand même un modèle conceptuel très élémentaire.

OWL

Il y a également le langage d'ontologies Web, OWL²¹. Ce langage étend RDF en y ajoutant des primitives de modélisation plus riches et en permettant de définir de nouveaux concepts à partir d'*expressions logiques* sur des concepts et des propriétés existants. L'utilisation de ces expressions logiques permet dès lors de déterminer des liens d'inclusions et de détecter automatiquement s'il y a des incohérences dans les schémas et les descriptions de ressources. OWL est aujourd'hui le standard pour les ontologies du Web et permet de représenter la sémantique des connaissances d'une façon accessible et manipulable pour les ordinateurs.

4.1.2. Quelques applications existantes du Web sémantique

Cette section décrit sommairement quelques applications existantes du Web sémantique.

Le projet SIMILE

Le projet SIMILE (Semantic Interoperability of Metadata and Information in unLike Environments) est conjointement développé par les groupes W3C, MIT Libraries (<http://libraries.mit.edu/>) et MIT CSAIL (MIT Computer Science and Artificial Intelligence Laboratory - <http://www.csail.mit.edu>). Il est disponible à l'adresse <http://simile.mit.edu/>. Cette application vise à améliorer l'interopérabilité entre les capitaux/informations numériques, les ontologies, les métadonnées et les services qu'il est possible d'en tirer. L'un des défis principaux est le fait que les différentes collections de données qui doivent interagir sont souvent réparties à travers des institutions gérées par des individus, des communautés et même des gouvernements, ce qui les rend souvent difficiles d'accès et non uniformes.

²¹ <http://www.w3.org/TR/owl-features/>

Le Projet Haystack

Disponible à l'adresse <http://haystack.lcs.mit.edu/>, cette application tente de mettre en place des approches permettant à l'utilisateur de manipuler de l'information de la manière qu'il juge la plus compréhensible. Bon nombre d'applications disposent de barrières intégrées pendant la construction qui ne permettent que la gestion de certains types d'informations et des relations. Avec l'application *Haystack*, un utilisateur peut définir les liens de son choix entre les différentes vues d'informations qu'il manipule. L'un des buts visés est d'accroître la capacité de l'utilisateur à retrouver en tout temps l'information dont il a besoin.

Conference

Disponible à l'adresse <http://www.ldodds.com/conference/Wiki.jsp>, cette application permet à l'utilisateur d'utiliser un environnement dans lequel il lui est possible de créer et d'éditer des pages Web décrivant des conférences, leurs animateurs et leurs participants. Cet environnement permet également de créer des liens entre les différents documents et des métadonnées partout sur le Web, en vue d'annoter l'information saisie. Cette application est basée sur un environnement *wiki*. Un *wiki* est un site Web dynamique dont les pages peuvent être modifiées à volonté par un usager.

Platypus Wiki

Disponible à l'adresse <http://platypuswiki.sourceforge.net/index.html>, cette application offre une interface utilisateur très commode pour la création de pages wiki utilisant les technologies du Web sémantique. Ainsi, lorsqu'un site est construit en utilisant cette application, les pages sont considérées comme étant des ressources (RdfResource) et les liens entre les pages sont considérés comme des propriétés (RdfProperty) décrivant les ressources.

FOAFRealm

Cette application est inspirée du projet FOAF (Friend of a friend) qui compose un vocabulaire RDF permettant de décrire des personnes et les relations qu'elles entretiennent entre elles. Cette application permet de gérer des profils utilisateurs, en utilisant un réseau social créé et maintenu par les usagers. Elle est disponible à l'adresse <http://www.foafrealm.org/>

Piggy Bank

Il s'agit d'une extension du navigateur Firefox²², qui a pour but de transformer ce navigateur en "navigateur Sémantique Web", ce qui permettrait aux utilisateurs de profiter de l'information existante sur le Web d'une manière plus utile et de façon flexible. Cette application est disponible à l'adresse <http://simile.mit.edu/piggy-bank/>

4.1.3. CIAO et le Web sémantique

La figure 3.1 (voir section 3.3.2) illustrant les interactions entre CIAO et le SA montre que CIAO utilise des technologies telles que le XML et le OWL pour la représentation des données. Le XML est utilisé pour les données provenant des fichiers de scénarios pédagogiques générés par le SA, et le OWL est utilisé pour la représentation des données au niveau de l'ontologie des théories du Design Pédagogique (DP). De ce fait, CIAO dispose d'une plus grande portabilité et d'une ouverture vis-à-vis du Web sémantique compte tenu de l'utilisation de standards qui le composent. Les ontologies valorisent le fonctionnement du Web de plusieurs façons. Elles utilisent de façon simple pour améliorer la pertinence des recherches effectuées par les programmes. Dans ce cas, ces programmes rechercheront des pages qui font référence à un concept précis au lieu de celles qui utilisent des mots-clés ambigus. De plus, des applications plus sophistiquées utilisent les ontologies pour associer l'information provenant d'une page à des structures de connaissance et à des règles d'inférence, ce qui permettrait de marquer des

²² <http://www.mozilla.com/firefox/>

informations afin de les retrouver plus facilement et de déduire certaines associations. Dans son utilisation actuelle, CIAO retrace des informations grâce aux liens existant entre les données de l'ontologie. Les règles d'inférence et de navigation contenues dans l'ontologie permettent alors à un usager d'accéder à des données en parcourant une structure définie dans l'ontologie.

4.2 Les choix technologiques

Dans le cadre de ce travail, nous avons analysé de plusieurs outils avant d'arrêter notre choix. Dans cette section, nous présentons les outils qui ont retenu notre attention. Par la suite, nous mentionnerons ceux que nous avons choisis. Comme outils possibles, nous avons examiné: Jade, Eclipse SDK (Software Development Kit), Jena, Sesame, SeRQL, Jsp, Tomcat.

JADE²³ (Java Agent DEvelopment Framework) est une plateforme logicielle écrite en Java. Elle simplifie l'implémentation de systèmes multi-agents en utilisant des interfaces compatibles avec les spécifications portant sur les agents intelligents. Ces spécifications proviennent du FIPA²⁴ (Foundation for Intelligent Physical Agents). JENA²⁵ est une plateforme en Java et sert aux applications pour le Web sémantique. Elle est distribuée en code ouvert et permet de manipuler des données en RDF, RDFS et OWL. ECLIPSE²⁶ est une communauté de code libre dont les projets sont axés sur la production de plateformes de développement et d'environnements de travail pour la construction de logiciels. Elle comprend un kit de développement de logiciels portant le nom d'*Eclipse SDK* qui permet la production d'applications en Java. SESAME²⁷ est un environnement de travail libre en Java. Il permet l'entreposage, l'interrogation et le raisonnement effectué sur des données RDF ou RDFS (RDF Schema). Il peut être utilisé comme librairie Java pour des applications qui ont besoin de manipuler du RDF. SeRQL est le langage de requête utilisé par la plateforme Sesame pour la manipulation de documents RDF. La technologie JSP²⁸

²³ <http://jade.tilab.com/>

²⁴ <http://www.fipa.org/>

²⁵ <http://jena.sourceforge.net/>

²⁶ <http://www.eclipse.org/>

²⁷ <http://www.openrdf.org/>

²⁸ <http://java.sun.com/products/jsp/>

permet la création de contenus Web dynamique. Elle rend possible la création d'applications Web complètement indépendantes du serveur ou de la plateforme sur laquelle elles sont exécutées. TOMCAT est une technologie développée par Apache²⁹. Il s'agit d'un conteneur de modules Java exécutés sur une application serveur en vue de répondre aux requêtes des clients. Ces requêtes sont effectuées à partir des pages Web dynamiques créées en JSP.

Après analyse de ces différentes technologies, nous avons opté pour les outils suivants : Sesame, SeRQL, Tomcat, JSP, Jena.

L'environnement de travail JADE n'a pas été retenu car le système CIAO n'est pas un système multi-agents. Dès lors, nous n'aurions pas été mesure de tirer profit des avantages offerts par cet outil. ECLIPSE SDK a été examiné car le développement de CIAO avait été amorcé en tant qu'application Java qui utiliserait les services requis sous forme d'APIs. Cet outil était donc comparé à l'environnement de programmation JBuilder que nous avons retenu car il est accessible dans les laboratoires du LICEF. Une autre raison qui a motivé le choix de JBuilder comparativement à Eclipse est le fait que nous avons accès à des développeurs du LICEF qui travaillaient avec cet outil et qui auraient donc pu nous aider en cas de problèmes techniques. Au final, nous avons opté pour le développement d'une application Web qui serait facilement accessible aux utilisateurs du LICEF. Cette application a ainsi été hébergée sur les serveurs du LICEF à l'adresse <http://www.liceftelug.quebec.ca/CIAO/>. En nous orientant vers un développement Web, nous avons eu accès services libres fournis par la plateforme Sesame qui s'intégrait parfaitement à la manipulation de données OWL et RDF contenues dans notre ontologie des théories du Design Pédagogique. Les autres choix ont alors suivi de façon logique. Nous avons alors utilisé le serveur d'application Tomcat de Apache afin d'héberger notre système, le langage SeRQL pour le développement des requêtes à exécuter sur notre ontologie, la technologie JSP pour l'utilisation de pages Web dynamiques dont le contenu pouvait être automatiquement mis à jour en fonction des changements dans notre base de connaissances et la technologie Jena qui se prêtait à la manipulation de données Web sémantique. Cette dernière technologie nous a permis de fournir deux modules de notre

²⁹ <http://tomcat.apache.org/>

service d'exploration de l'ontologie. Nous l'avons notamment utilisé pour le parcours hiérarchique de notre ontologie (développée par Valéry Psyché) et sa description sous forme de classes et de sous-classes.

L'ensemble des choix que nous avons arrêté a été fait en gardant à l'esprit l'usage d'un langage de programmation portable comme Java. Cette portabilité nous permet alors de pouvoir faire exécuter le système et ses sous-services sur différents types de plateformes. Une fois ces choix technologiques effectués, nous arrivons à l'implémentation de notre système. Cette activité est décrite dans la prochaine section.

4.3 L'implémentation de CIAO

Cette section présentera de façon détaillée les différents services offerts par l'agent CIAO, avec des illustrations de l'implémentation qui a été réalisée.

4.3.1. La gestion des répertoires et des utilisateurs

CIAO donne accès à deux répertoires contenant et les classes et les instances de classes de l'Ontologie des Théories du Design Pédagogique (OTDP). Chaque répertoire est géré de façon indépendante. De plus, il est possible de gérer les utilisateurs en leur associant des droits particuliers sur chacun des répertoires. L'outil de configuration de Sesame permet ainsi de gérer autant d'utilisateurs que souhaité et de créer autant de fichiers associés aux différentes bases de connaissances.

Dans le cas du projet CIAO, nous avons créé deux répertoires car les éditeurs d'ontologie utilisés pour la création de l'OTDP, Hozo³⁰ et Protégé³¹, permettent d'extraire un fichier pour les classes et objets de l'ontologie et un autre pour les instances de l'ontologie. CIAO permet dès lors d'accéder à ces deux ensembles de données.

³⁰ <http://www.hozo.jp/>

³¹ <http://protege.stanford.edu/>

1- Chargement de la configuration du système

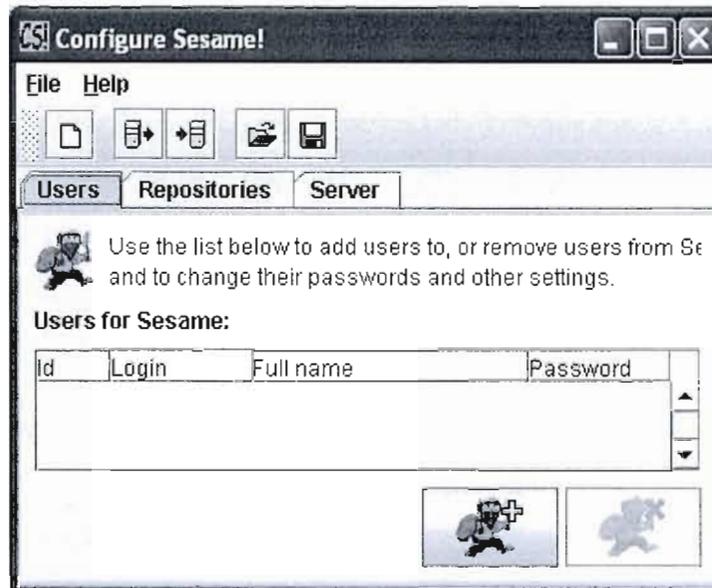


Figure 4.1: Outil de configuration "Configure Sesame !".

La figure 4.1 présente l'outil de configuration de Sesame juste après son ouverture. Cet outil édite le fichier de configuration de Sesame (*system.conf*) localement sur notre serveur Sesame ou en l'installant comme utilitaire sur un poste Client. Une fois l'utilitaire "Configure Sesame !" lancé, on configure le serveur en cours d'exécution ou en mode arrêt. Dans le premier cas, on charge directement le fichier de configuration à partir du serveur (menu [File] --> [Load from server...]). Dans le second cas, on le charge à partir du disque (menu [File] --> [Open file...]).

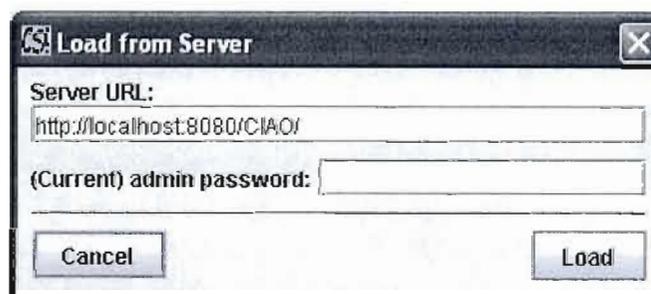


Figure 4.2: Chargement de la configuration à partir du serveur.

Une fois la fenêtre ouverte, on spécifie le mot de passe par défaut qui est "admin" s'il s'agit d'une nouvelle installation de Sesame, ou alors on rentre le nouveau mot de passe convenu.

2- Modification de la configuration système

Une fois le fichier de configuration chargé, on peut modifier les paramètres du système. Dans le cas de CIAO, on obtient alors la fenêtre suivante :

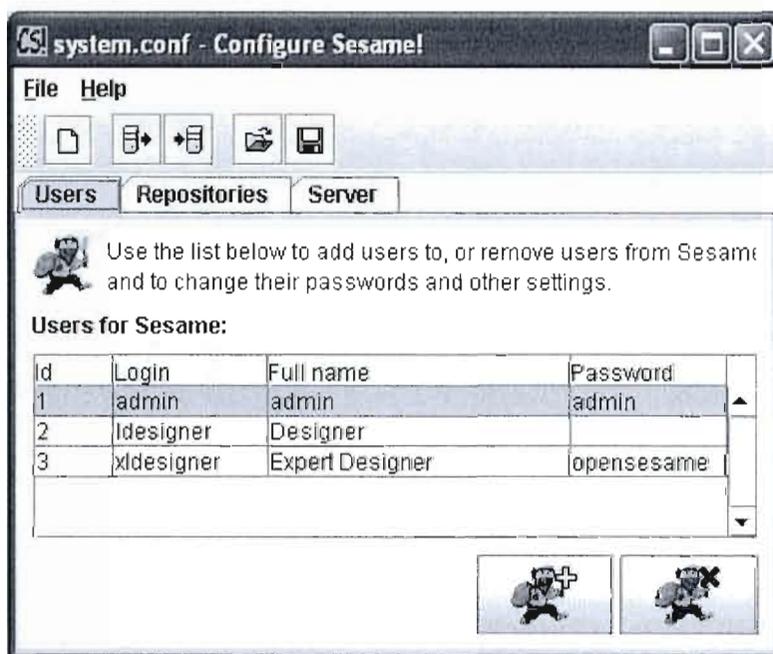


Figure 4.3: Gestion des utilisateurs.

Ici, nous avons créé trois utilisateurs afin de leur affecter des caractéristiques particulières pour l'accès au contenu de l'application.

L'onglet "**Repositories**" de la fenêtre précédente permet de gérer les répertoires pour chacune des deux bases de connaissances.

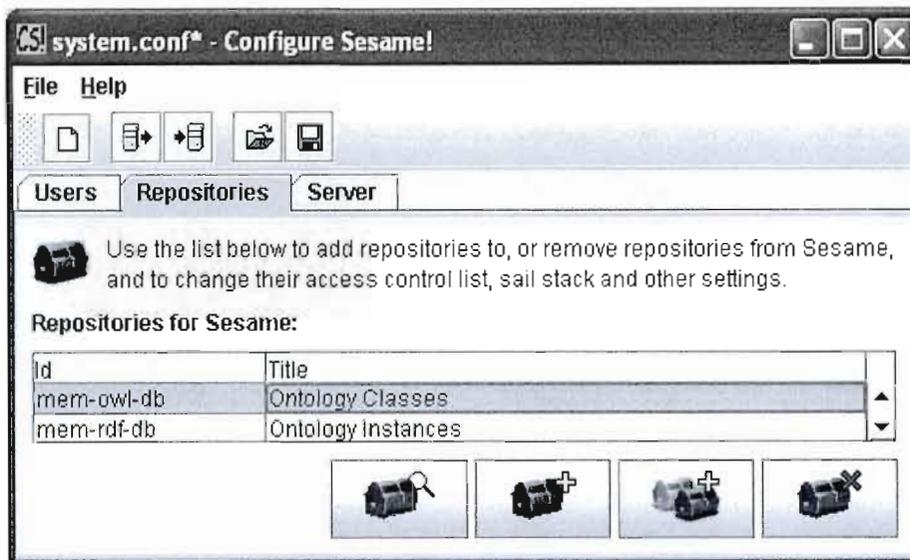


Figure 4.4: Gestion des répertoires.

Chaque répertoire dispose d'un identifiant unique et d'un titre. Dans le cadre de notre application, il y a deux répertoires nommés **mem-owl-db** et **mem-owl-rdf**, respectivement pour les classes et les instances de l'ontologie des théories du DP. La gestion des propriétés de chaque répertoire et des droits d'accès pour chacun des utilisateurs se fait en sélectionnant un répertoire et en cliquant sur le premier bouton de la barre d'outils située au bas de la fenêtre. Une fois cette manœuvre effectuée, on accède à fenêtre de la figure suivante qui liste les propriétés des répertoires.

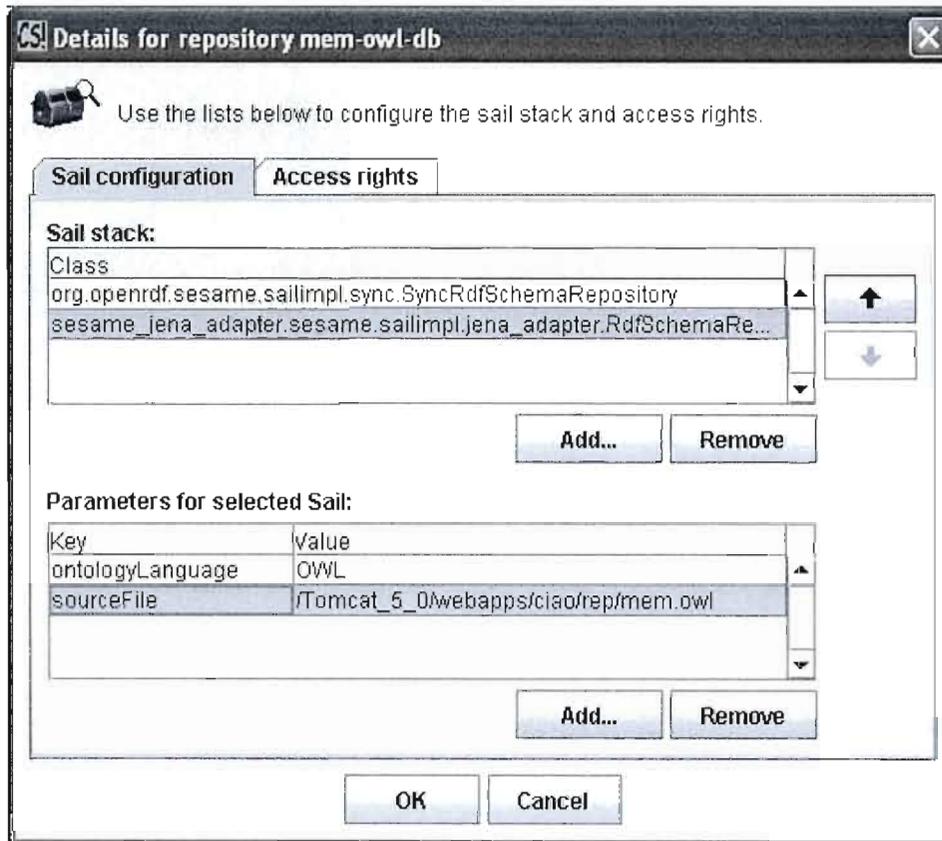


Figure 4.5: Détail des répertoires.

La fenêtre précédente nous donne les propriétés de l'objet **mem-owl-db**. Nous pouvons y voir que le fichier sur disque qui contient les données de classes se nomme **mem.owl**. Son contenu est généré par exportation à partir de l'éditeur d'ontologies HOZO. L'onglet "Access rights" de la précédente fenêtre permet de préciser les droits d'accès pour chaque type d'utilisateur.

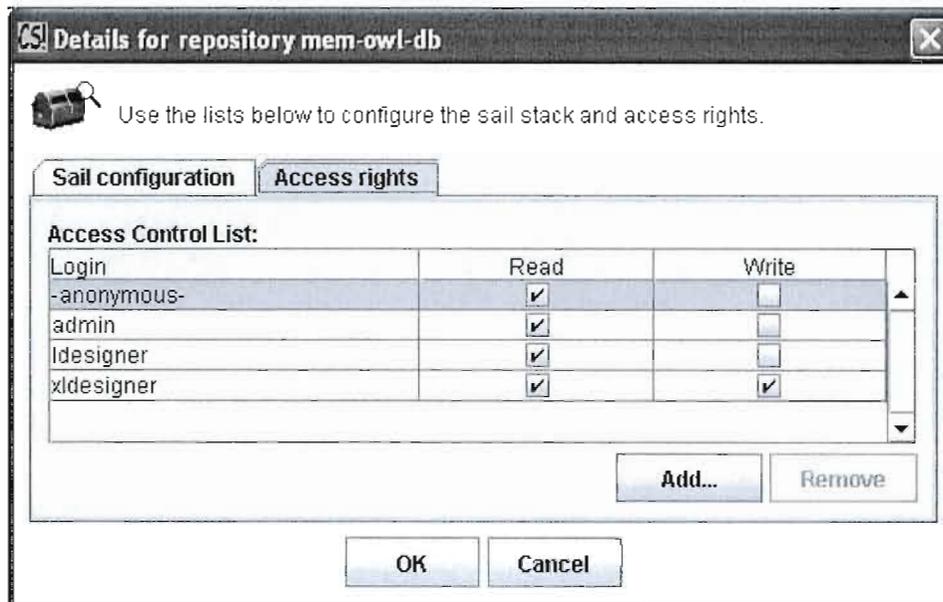


Figure 4.6: Gestion des droits d'accès.

La figure 4.6 montre qu'en plus des utilisateurs agréés, il y a un utilisateur nommé "**anonymous**". Les droits affectés à cet utilisateur sont utilisés lorsqu'un usager accède à l'application CIAO sans s'identifier. Dans ce cas, les seuls accès qu'il aura seront des droits de lecture des différents répertoires.

3- Sauvegarde de la configuration système

Une fois la configuration modifiée, on l'enregistre sur disque (menu [File] --> [Save file as...]), ou on l'envoie au serveur CIAO en cours d'exécution (menu [File] --> [Send to server...]). Lorsque la deuxième option est utilisée, les changements apportés au fichier de configuration sont automatiquement appliqués au serveur en exécution, sans besoin de le redémarrer ou de le rafraîchir.

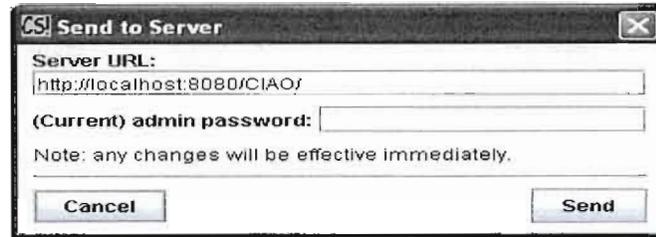


Figure 4.7: Envoi de la configuration au serveur en exécution.

Une fois la gestion des répertoires et droits d'accès effectués, on peut rendre disponible l'application CIAO avec les nouveaux paramètres.

4.3.2. Lancement de l'application

Une fois CIAO lancé à partir du navigateur, apparaît la fenêtre d'accueil suivante :

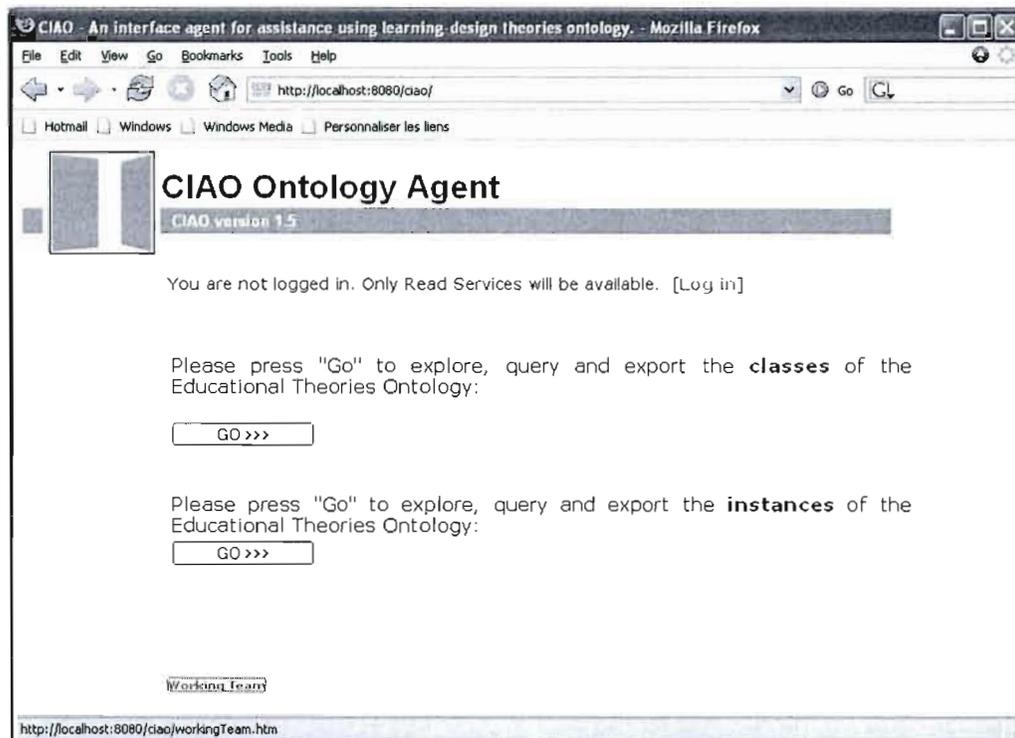


Figure 4.8: Page d'accueil de CIAO.

Cette page d'accueil permet à l'utilisateur d'accéder à l'application à partir des classes ou des instances de l'ontologie. Les mêmes services sont offerts mais les résultats diffèrent selon le service demandé. Un usager qui dispose de droits d'accès particuliers peut alors se

connecter en s'identifiant. Les seuls services nécessitant des droits d'accès et une identification de la part de l'utilisateur sont les services d'écriture qui permettent à l'utilisateur de modifier le contenu des bases de connaissances. CIAO, dans sa version actuelle, confère ces droits qu'à un administrateur ou un concepteur expert.

Parmi les quatre services principaux de CIAO, on retrouve le service d'exploration (voir section 4.3.3) qui fournit les mêmes résultats, peu importe le répertoire auquel l'utilisateur accède. Il en est de même pour le service d'analyse et validation. Par contre, les services de recherche par requête et celui de l'exportation fournissent des résultats différents selon le répertoire exploré par l'utilisateur.

Dans les sections qui suivent, les services d'exploration et d'analyse seront sollicités à partir du répertoire des classes de l'ontologie. Les services de recherche et d'exportation seront présentés selon chacun des répertoires. Chacun de ces quatre services est accessible à l'utilisateur dans le menu des services de lecture.

4.3.3. Le service d'exploration

Ce service comporte quatre sous-services représentés par les options suivantes : *Documentation, Hierarchy, Description, Repository.*

Cette sous-section permet à l'utilisateur de visualiser la documentation de l'ontologie sous la forme donnée à la figure 4.10.

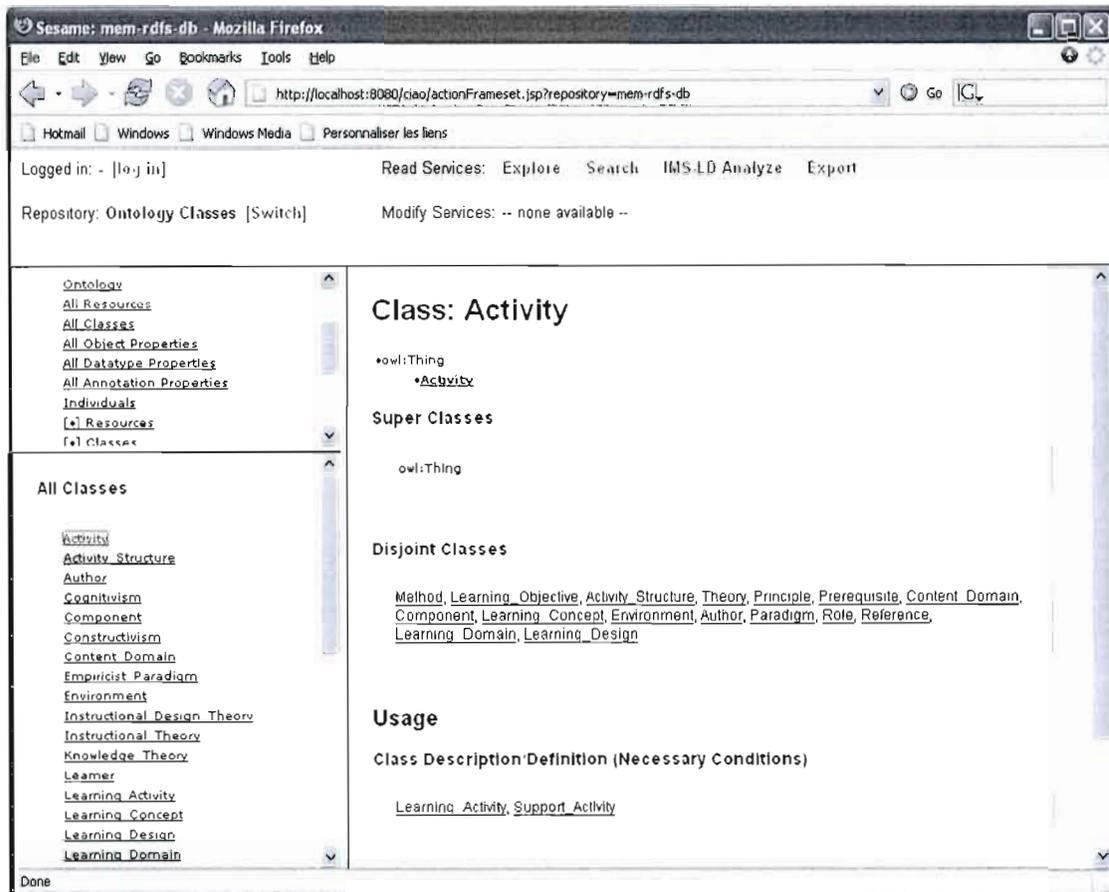


Figure 4.10: Sous-service "Documentation" du service d'exploration de CIAO.

L'utilisateur sélectionne un élément dans le cadre supérieur gauche, puis un sous-élément dans le cadre inférieur gauche et la description associée apparaît dans le cadre de droite. Cette représentation du contenu de l'ontologie est conviviale et fonctionne selon les fichiers d'aide des interfaces de programmation d'applications (API) Java.

Hierarchy

Ce second sous-service de l'exploration offre une représentation textuelle de la hiérarchie des classes et sous-classes de l'ontologie des théories. Elle fournit l'interface de la figure 4.11.

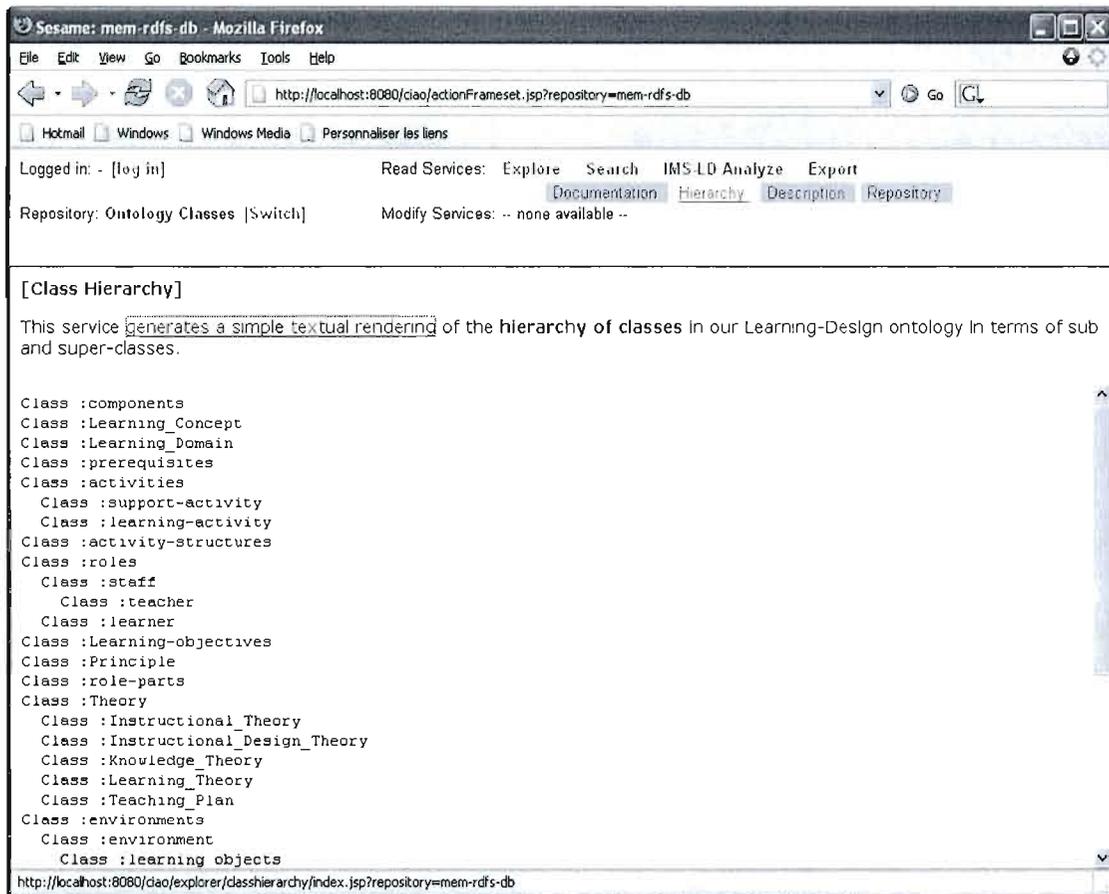


Figure 4.11: Sous-service "Hierarchy" du service d'exploration de CIAO.

Cette fonctionnalité a été développée en utilisant l'outil Jena pour la manipulation des données contenues dans l'ontologie des théories.

Description

Ce troisième sous-service génère une description textuelle de chacune des classes de l'ontologie des théories, et indique les restrictions qui s'appliquent à chacune d'elles. Pour chaque classe, l'application indique les classes auxquelles elle est rattachée, et les types de liens existant pour chaque relation. Cette fonctionnalité a été mise au point en utilisant l'API de Jena pour la manipulation des données de l'ontologie.

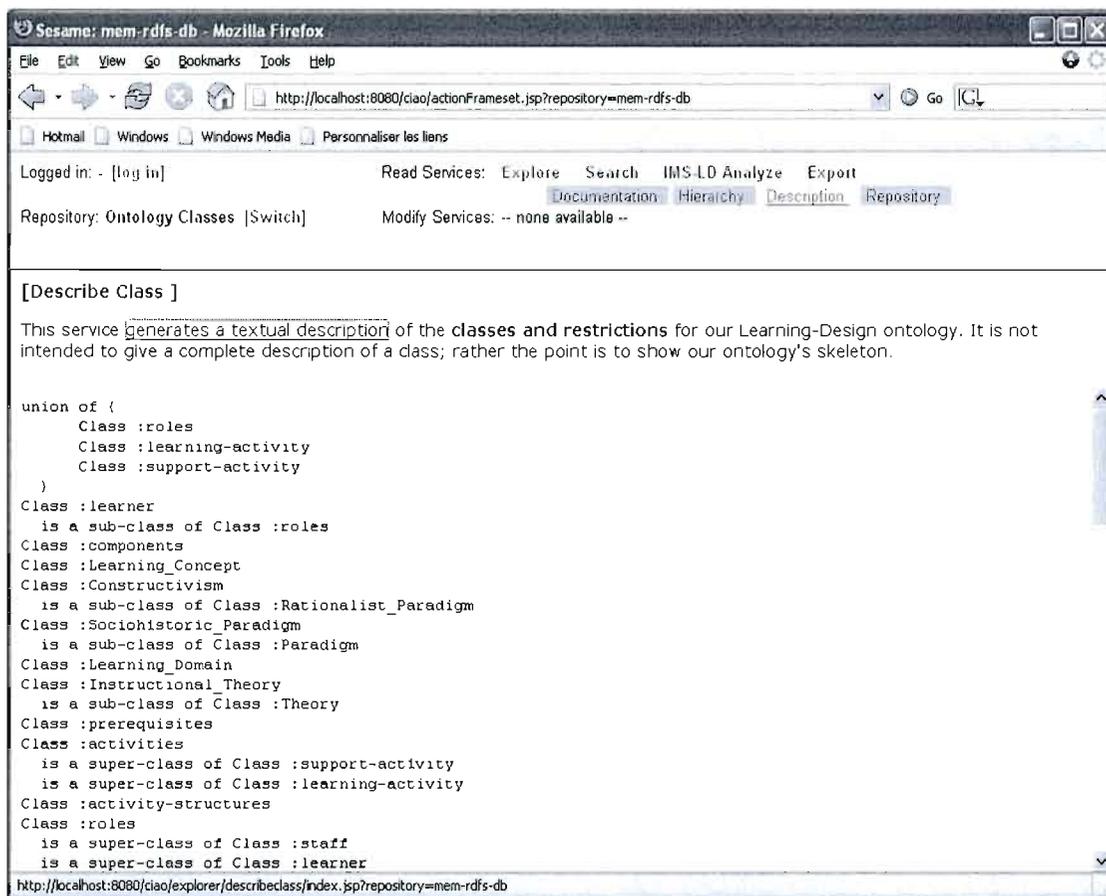


Figure 4.12: Sous-service "Description" du service d'exploration de CIAO.

Repository

Ce quatrième et dernier sous-service de l'exploration permet à l'utilisateur de naviguer dans l'ensemble de l'ontologie des théories en utilisant une visualisation sous forme de triplets. Il permet donc à l'utilisateur de percevoir chaque objet de l'ontologie comme un sujet, un prédicat, ou un objet, comme le définit le standard RDF. Le déplacement s'effectue à partir de liens hypertextes. Pour chaque élément de l'ontologie, le système affiche l'ensemble des triplets dans lesquels cet élément joue le rôle de sujet, l'ensemble des triplets dans lesquels cet élément joue le rôle de prédicat et l'ensemble des triplets dans lesquels cet élément joue le rôle d'objet. L'activation de cette option offre d'abord un premier niveau de visualisation dans lequel l'utilisateur aperçoit chaque classe et chaque propriété de l'ontologie. L'écran associé apparaît à la figure 4.13.

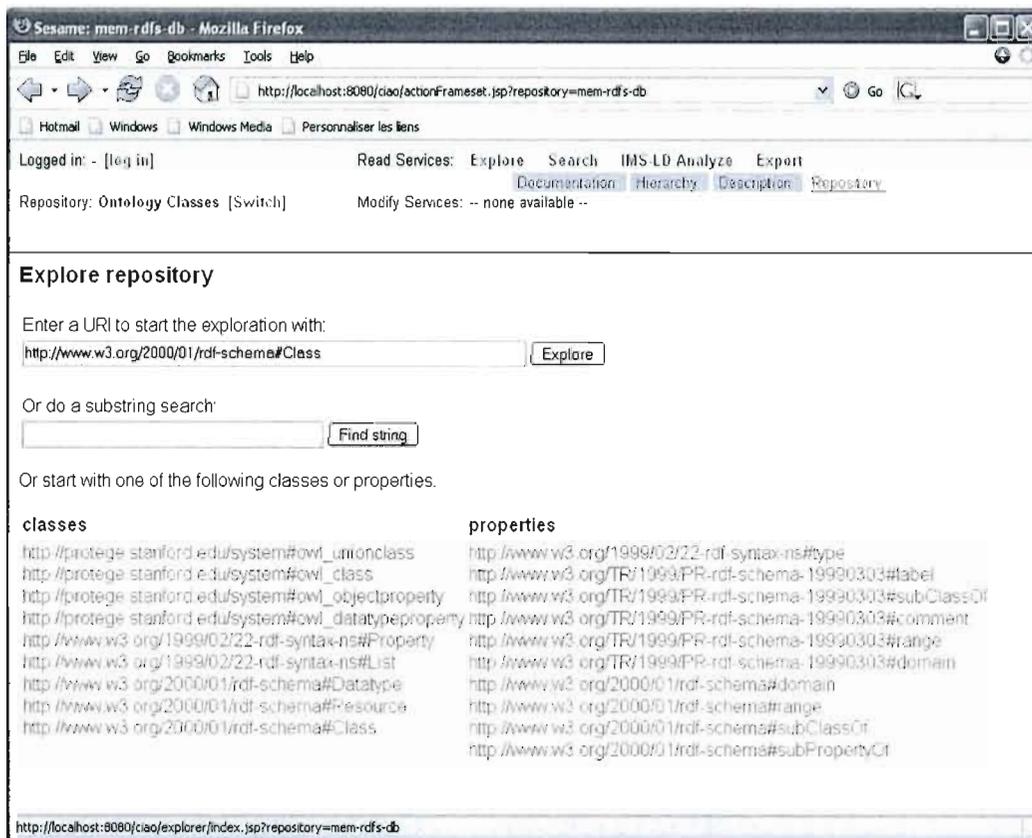
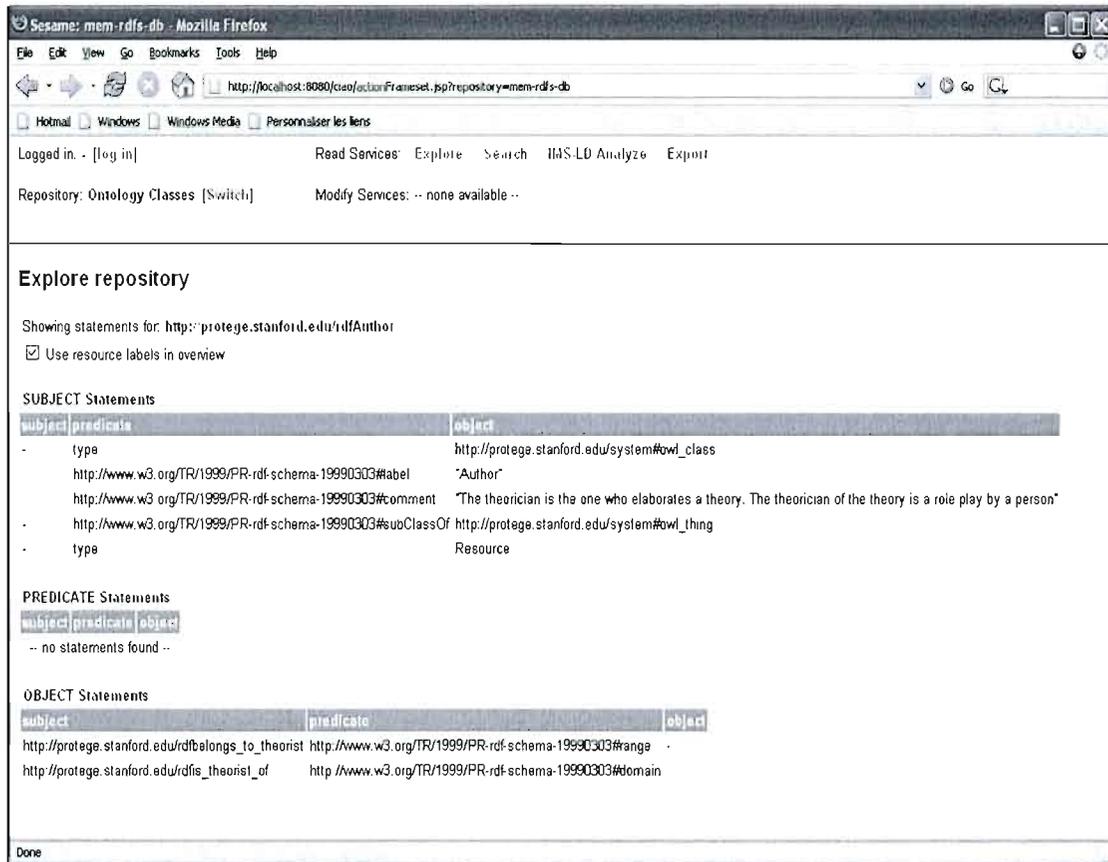


Figure 4.13: Sous-service "Repository" du service d'exploration de CIAO.

La figure suivante montre l'état du système après une série de déplacements effectués par l'utilisateur. Sur cet écran, l'utilisateur veut connaître quels sont les triplets dans lesquels intervient l'objet "rdfAuthor". Cet objet correspond à la classe utilisée par l'ontologie pour identifier les auteurs des différentes théories.



Sur la figure précédente, on voit que l'objet "rdfAuthor" dont l'URI est "`http://protege.stanford.edu/rdfAuthor`" apparaît dans cinq triplets RDF en tant que sujet, dans aucun triplet RDF en tant que prédicat, et dans deux triplets RDF en tant qu'objet.

Les triplets dans lesquels cet objet apparaît en tant que sujet montrent respectivement que:

- l'objet "rdfAuthor" est de type "owl_class" ;
- l'objet "rdfAuthor" est géré par le système sous l'étiquette "Author" ;
- l'objet "rdfAuthor" dispose d'un commentaire indiquant ceci : "The theorician is the one who elaborates a theory. The theorician of the theory is a role play by a person" ;

- l'objet "rdfAuthor" est une sous-classe de "owl_thing". Cette classe représente la classe de base pour toutes les classes d'une ontologie décrite avec l'éditeur Protégé.
- l'objet "rdfAuthor" est de type "Ressource", ceci pour dire qu'il constitue une ressource de l'ontologie, comme tout autre objet.

Les triplets dans lesquels l'objet "rdfAuthor" apparaît en tant qu'objet nous montrent respectivement que:

- l'objet "rdfAuthor" a pour *champ* la propriété "rdfbelongs_to_theorist"
- l'objet "rdfAuthor" a pour *domaine* la propriété "rdfis_theorist_of" ;

Le *champ* et le *domaine* sont des éléments qu'il est possible de préciser en tant que caractéristiques d'une propriété au moment de sa création dans l'éditeur Protégé. La figure 4.14 suivante utilise les propriétés *belongs_to_theorist* et *is_theorist_of* pour illustrer la relation entre un auteur et une théorie, respectivement représentés par les objets "rdfAuthor" et "rdfTheory" travers les liens/propriétés

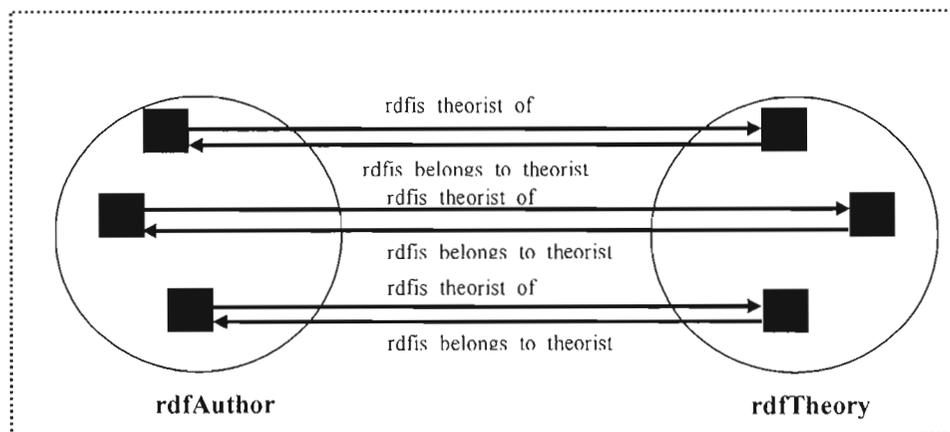


Figure 4.14: *Domain* et *Range* de propriétés dans une relation.

La figure 4.14 explique l'idée selon laquelle l'objet de type *rdfAuthor* participe à une relation avec un objet de type *rdfTheory*. Les liens unissant ces deux objets sont alors des liens *rdfis_theorist_of* et *rdf_belongs_to_theorist*. Ainsi, un auteur donné, représenté par un objet *rdfAuthor* est le théoricien associé à une théorie donnée, représentée par *rdfTheory*. Inversement, une théorie appartient à un auteur. Dans chacune de ces relations, l'élément *domain* réfère à l'ensemble de départ et le *champ* à l'ensemble

d'arrivée. Dans une relation *rdfis_theorist_of*, l'objet *rdfAuthor* représente le domaine et l'objet *rdfTheory* représente le *champ*.

4.3.4. Le service de requête

Ce service permet à l'utilisateur d'extraire des informations de l'ontologie des théories en formulant des requêtes selon trois modes distincts. Ces modes sont désignés par *mode débutant*, *mode intermédiaire* et *mode expert*, en fonction des connaissances et habiletés de l'utilisateur dans la manipulation du langage de requêtes SeRQL³³. C'est le langage utilisé pour l'écriture de requêtes sur la plateforme Sesame.

1- Mode débutant

Pour ce type de requêtes, l'utilisateur choisit une requête dans une liste prédéfinie dans le système CIAO. Le contenu de la liste diffère selon qu'on se trouve dans le répertoire des classes ou dans celui des instances. Cette fonctionnalité est accessible à partir du sous-menu *Query list* de *Search* de CIAO. Dans le répertoire des classes, il est ainsi possible d'obtenir la liste des sous-classes de la classe THEORY. Cette requête fournirait ainsi les différents types de théories contenues dans l'ontologie. Dans le répertoire des instances, on peut obtenir des instances de théories, qui correspondent alors à différents objets de type "rdfTheory".

Le code source écrit en SeRQL pour la première requête est le suivant :

```
'select Sub_class from {S} ns3:subClassOf {ns4:Theory}; ns3:label {Sub_class} using
namespace ns3 = <http://www.w3.org/TR/1999/PR-rdf-schema-19990303#>, ns4 =
<http://protege.stanford.edu/rdf>';
```

³³ <http://www.openrdf.org/doc/sesame/users/ch06.html>

Le code source pour la seconde requête est :

```
'select Theory from {T} rdf:type {ns4:Theory}; ns3:label {Theory} using namespace
ns3 = <http://www.w3.org/TR/1999/PR-rdf-schema-19990303#>,
ns4 = <http://protege.stanford.edu/rdf>';
```

Ces requêtes, comme toutes les autres du système CIAO utilisent différents espaces de noms, ou namespaces, dont *ns3*, *ns4* et *rdf*. Ces espaces de noms sont automatiquement créés par l'éditeur d'ontologies Protégé lors de la génération des fichiers RDF associés à l'ontologie.

Le RDF utilise une syntaxe de type XML est permet de retracer chaque élément en lui associant un identifiant unique, l'URI (Uniform Resource Identifier), qui a été appelé IRI (Internationalized Resource Identifier) à partir de la version 1.1 de XML. Un namespace ou espace de nom permet alors de définir des éléments et attributs pouvant porter des noms identiques, tout en appartenant à des IRI différentes. Ces IRI permettent donc de localiser de façon plus précise un élément dans une structure XML ou RDF.

La requête 1 utilise l'expression **ns3:subClassOf {ns4:Theory}; ns3:label {Sub_class}** afin de trouver toutes les étiquettes (*ns3 :label*) des théories (*ns4 :Theory*) qui sont de type sous-classe (*ns3 :subClassOf*), en faisant référence aux IRI dans lesquelles chaque objet se trouve. La requête 2 utilise l'expression **rdf:type {ns4:Theory}; ns3:label {Theory}** afin de trouver toutes les étiquettes (*ns3 :label*) correspondant à des objets qui sont de type (*rdf :type*) théorie (*ns4 :Theory*).

Les deux prochaines figures (4.15 et 4.16) montrent le résultat produit par chacune des requêtes en mode débutant.

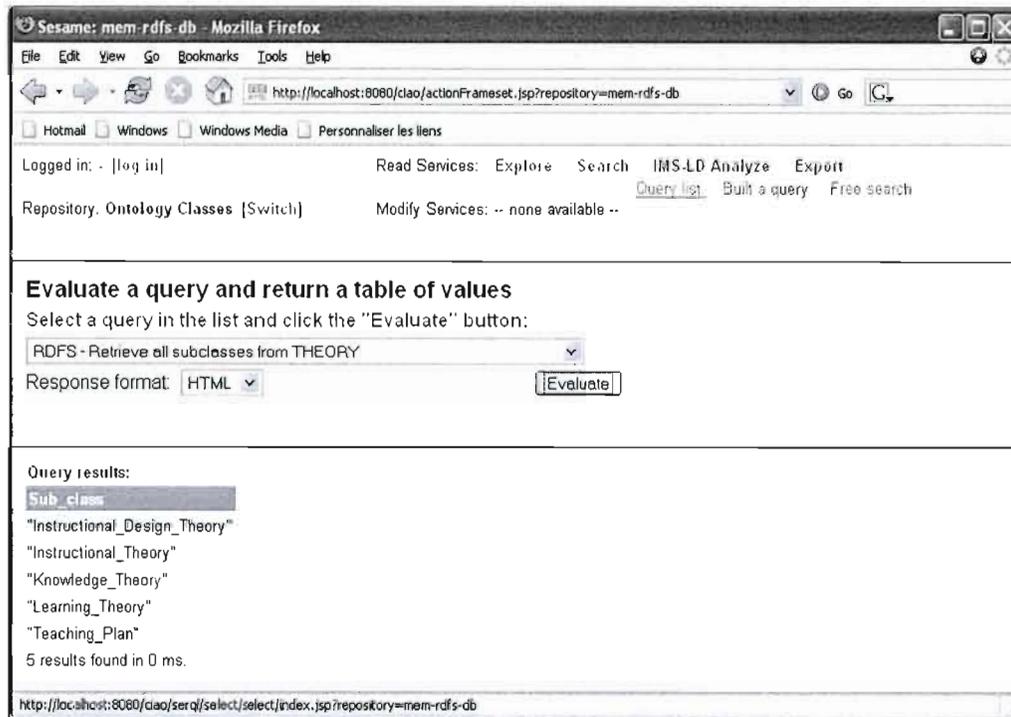


Figure 4.15: Résultat d'une requête en mode débutant sur le répertoire des classes.

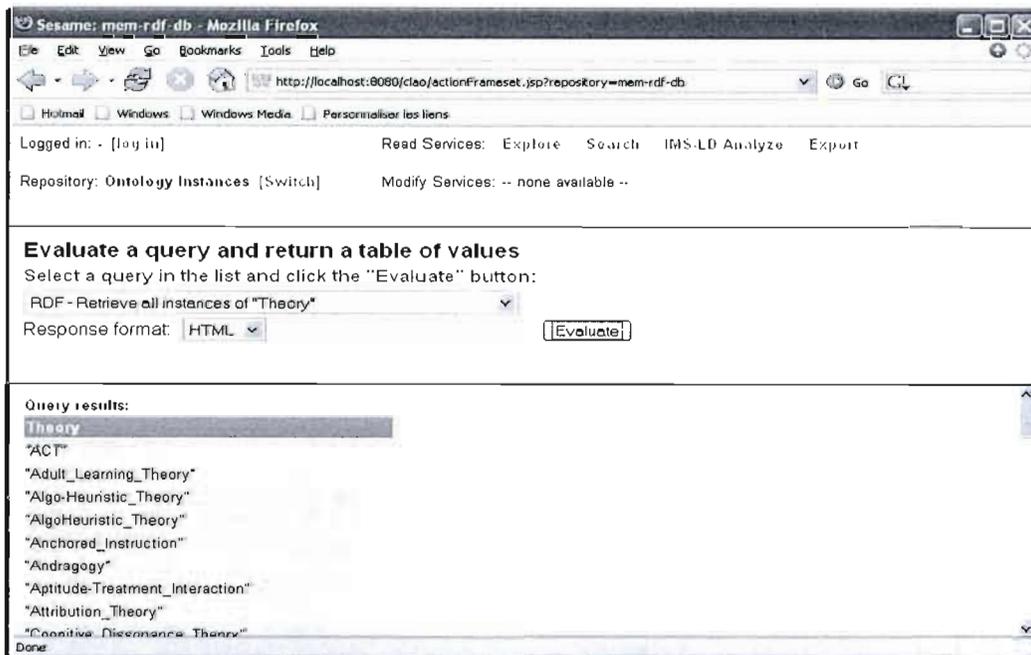


Figure 4.16: Résultat d'une requête en mode débutant sur le répertoire des instances.

2- Mode intermédiaire

Pour ce type de requêtes, le système aide l'utilisateur à choisir les éléments qu'il souhaite inclure dans sa requête et en construit automatiquement le code SeRQL. Cette fonctionnalité est accessible à partir du sous-menu *Built a query* du menu *Search* de CIAO. Dans le répertoire des classes, l'utilisateur peut ainsi choisir de sélectionner les *sous-classes* ou les *commentaires* associés à une classe donnée. Dans le répertoire des instances, l'utilisateur peut demander au système de lui fournir toutes les *instances* d'une classe donnée. Ces deux fonctionnalités prévoient l'utilisation de critères dans la formation de la requête afin de mieux cibler le résultat de la requête.

Les deux prochaines figures (4.17 et 4.18) illustrent des exemples de requêtes obtenues à partir des différents répertoires. Sur la figure 4.17, l'utilisateur demande la liste des sous-classes de la classe *Paradigm*. Le système lui fournit une liste des paradigmes d'apprentissages existant dans le domaine du Design Pédagogique (DP). Sur la figure 4.18, l'utilisateur demande la liste des instances de la classe *Author*. Le programme lui fournit l'ensemble des auteurs inscrit dans l'ontologie.

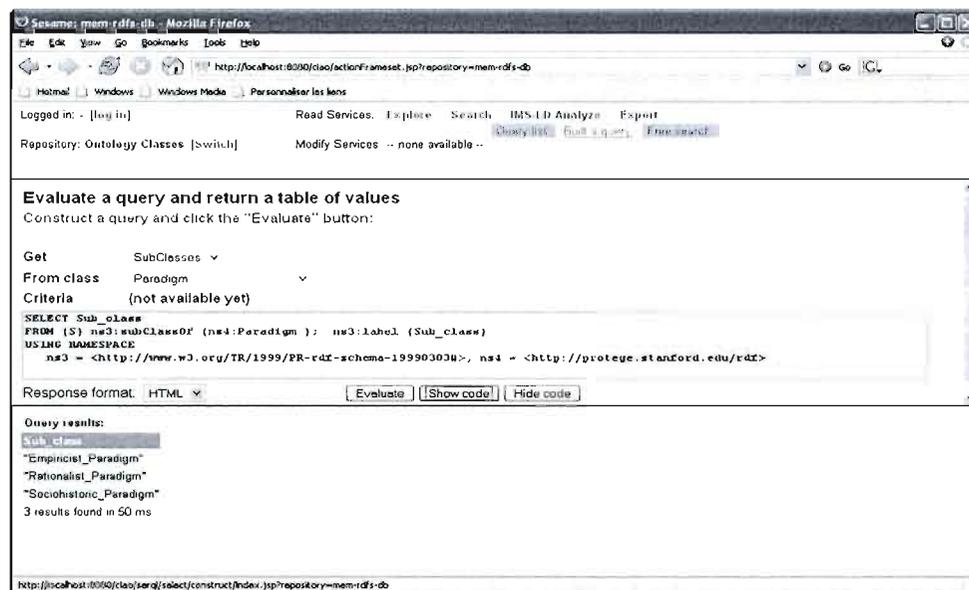


Figure 4.17: Requête en mode intermédiaire dans le répertoire des classes.

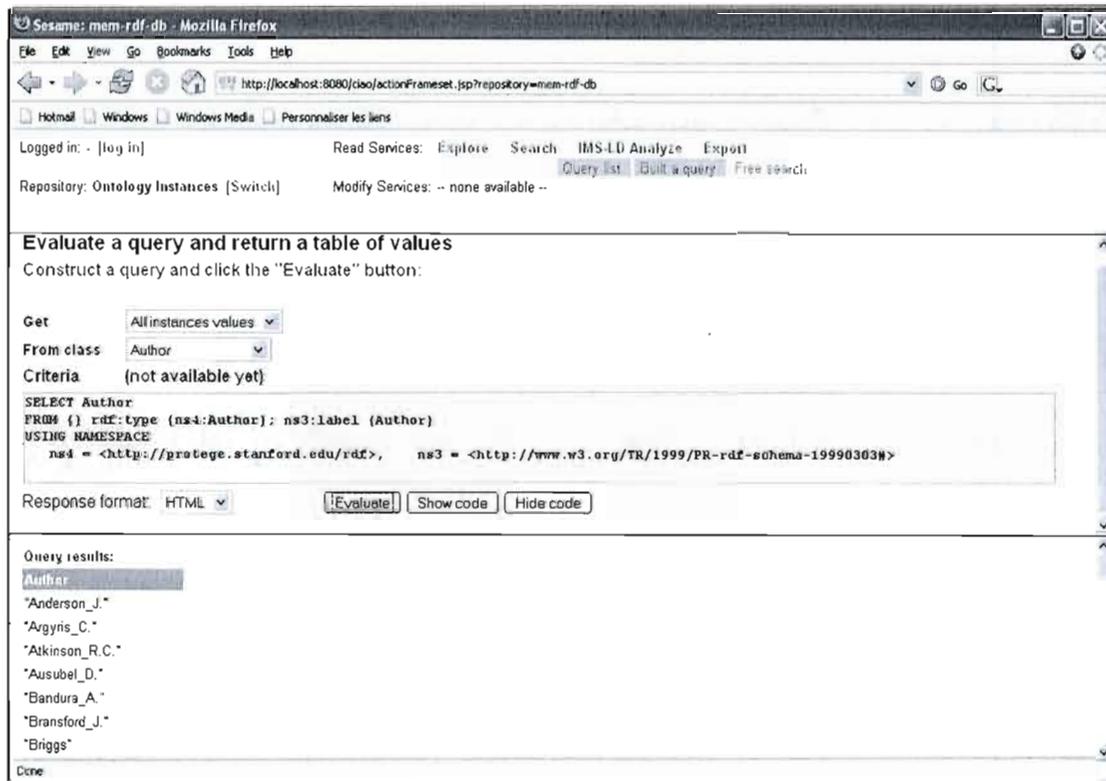


Figure 4.18: Requête en mode intermédiaire dans le répertoire des instances.

Pour obtenir les résultats, l'utilisateur clique sur le bouton *Evaluate*. Par défaut, les différents éléments fournis dans la liste de résultats sont dans le format HTML. Dans ce cas, l'utilisateur peut alors explorer chaque élément par un clic, et obtenir la liste des triplets RDF dans lesquels cet élément intervient. L'utilisateur peut décider d'afficher le résultat de la requête dans un format XML ou RDF en sélectionnant l'option appropriée devant la case *Response Format*.

Sur chaque écran, le système offre la possibilité à l'utilisateur d'afficher ou de masquer le code SeRQL produit pour la requête. À cet effet, l'utilisateur dispose de deux boutons nommés *Show code* et *Hide code*.

En plus des deux modes que nous venons de discuter, CIAO dispose d'un troisième mode qui permet à l'utilisateur d'écrire sa requête.

3- Mode expert

Cette fonctionnalité est accessible à partir du sous menu *Free search* de *Search* de CIAO. Elle suppose que l'utilisateur est familier avec le langage SeRQL et est donc capable de rédiger des requêtes complètes. Une fois l'option sélectionnée, la fenêtre correspondante s'affiche avec un exemple de requête rédigé en fonction du répertoire dans lequel se trouve l'utilisateur. Par la suite, l'utilisateur peut modifier la requête dans l'espace approprié et obtenir le résultat en cliquant sur le bouton *Evaluate*. En cas d'erreur, un message explicite est affiché et indique à l'utilisateur la source de l'erreur rencontrée.

La figure 4.19 illustre la fenêtre apparaissant après le choix du mode expert alors que l'utilisateur se trouve dans le répertoire des classes. Par défaut, elle contient également dans le volet inférieur des indications sur des sites Web où l'utilisateur peut obtenir le manuel de référence du langage SeRQL ou des exemples de requêtes produites en SeRQL.

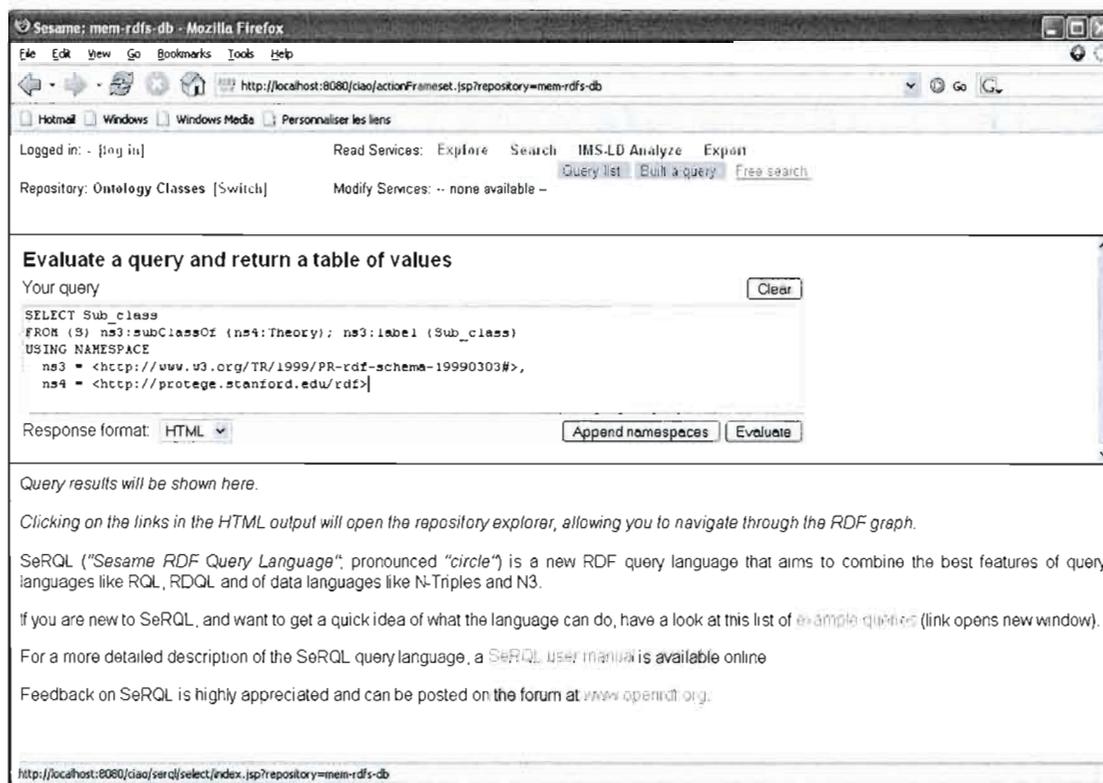


Figure 4.19: Requête en mode expert dans le répertoire des classes.

4.3.5. Le service d'analyse

Ce service permet à l'utilisateur de valider un scénario pédagogique conforme au standard IMS-LD. Deux types d'analyses sont effectués : une analyse syntaxique et une analyse sémantique. L'analyse syntaxique assure que le scénario de l'utilisateur respecte les normes de construction définies par le Consortium IMS pour le Learning Design (LD). Un ensemble de règles vérifient qu'un scénario contient les éléments obligatoires et respecte la bonne cardinalité pour chacun d'eux. L'analyse sémantique permet au système de faire des recommandations à l'utilisateur selon le contenu de son scénario pédagogique. Elle est effectuée uniquement si le scénario est syntaxiquement correct.

1- Analyse syntaxique

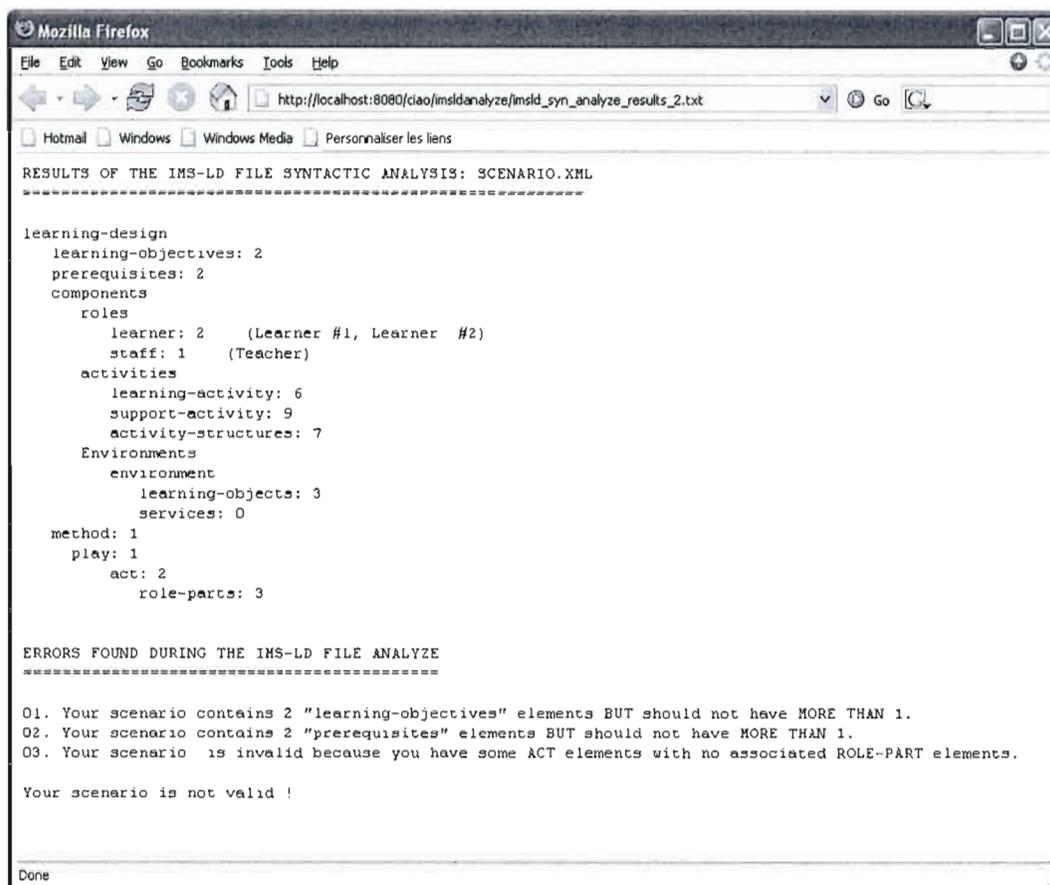
Pour cette analyse, l'utilisateur fournit au système un scénario pédagogique pour s'assurer que les règles de construction sont respectées. L'annexe B fournit la structure hiérarchique globale pour un fichier de scénario pédagogique. L'annexe B.4 fournit les règles utilisées pour effectuer cette validation. Pour tout scénario pédagogique, il existe une liste d'éléments obligatoires et une liste d'éléments recommandés par le Consortium IMS. En plus de ces éléments, ont été incluses d'autres recommandations faites par les experts du laboratoire LICEF de la Télé-Université en ce qui concerne la construction des scénarios pour le Design Pédagogique. La figure 4.20 suivante montre un scénario pédagogique contenant des "erreurs" où sera effectuée une analyse syntaxique.

```

- <unscp module et xsi:schemaLocation=' http://www.imsglobal.org/xsd/imsdcp_v1p1 http://www.imsglobal.org/xsd/imsdcp_v1p1.xsd http://www.imsglobal.org/xsd/imsmd_v1p2
http://www.imsglobal.org/xsd/imsmd_v1p2p2.xsd http://www.imsglobal.org/xsd/imsd_v1p0 http://www.imsglobal.org/xsd/IMS_LD_Level_A.xsd'
  identifier="Domain1-manifest-4CD851CA-3FF5-4128-9039-A750D757DA38">
  <!-- ORGANIZATIONS -->
  - <unsty:organizations>
  <!-- THE LEARNING DESIGN (SAMPLE WITH ERRORS)-->
  - <imsld learning-design identifier="Domain1" uri="URI" level="A" version="" sequence-used="false">
    <imsld title>Presentation (expository)</imsld title>
    + <imsld learning-objectives></imsld learning-objectives>
    + <imsld learning-objectives></imsld learning-objectives>
    + <imsld prerequisites></imsld prerequisites>
    <imsld prerequisites></imsld prerequisites>
    <!-- COMPONENTS -->
    + <imsld components></imsld components>
    <!-- METHOD -->
    - <imsld method>
    <!-- PLAY -->
    - <imsld play identifier="Domain1Node1">
      <imsld title>Play for objective #1</imsld title>
      <!-- ACT -->
      - <imsld act identifier="Domain1Node2">
        <imsld title>Act for objective #1</imsld title>
        <!-- ROLE-PARTS -->
        + <imsld role-part></imsld role-part>
        + <imsld role-part></imsld role-part>
        + <imsld role-part></imsld role-part>
        </imsld act>
      - <imsld act identifier="Domain1Node2">
        <imsld title>Act for objective #1</imsld title>
        <!-- ROLE-PARTS -->
        </imsld act>
      </imsld play>
    </imsld method>
    </unsty:organizations>
  <!-- RESOURCES -->
  + <unscp resources></unscp resources>
  </unscp module et
  
```

Figure 4.20: Scénario pédagogique syntaxiquement incorrect.

Une fois l'analyse syntaxique effectuée, le système CIAO affichera la fenêtre de résultats indiquant que le scénario pédagogique contient trois (03) erreurs de syntaxe. La figure 4.21 suivante montre la fenêtre produite par CIAO pour ce scénario.



```
RESULTS OF THE IMS-LD FILE SYNTACTIC ANALYSIS: SCENARIO.XML
=====

learning-design
  learning-objectives: 2
  prerequisites: 2
  components
    roles
      learner: 2      (Learner #1, Learner #2)
      staff: 1      (Teacher)
    activities
      learning-activity: 6
      support-activity: 9
      activity-structures: 7
    Environments
      environment
        learning-objects: 3
        services: 0
  method: 1
  play: 1
  act: 2
  role-parts: 3

ERRORS FOUND DURING THE IMS-LD FILE ANALYZE
=====

O1. Your scenario contains 2 "learning-objectives" elements BUT should not have MORE THAN 1.
O2. Your scenario contains 2 "prerequisites" elements BUT should not have MORE THAN 1.
O3. Your scenario is invalid because you have some ACT elements with no associated ROLE-PART elements.

Your scenario is not valid !

Done
```

Figure 4.21: Exemple de résultats d'analyse syntaxique pour un fichier erroné.

Lors de la validation syntaxique (et sémantique également), CIAO dresse un rapport qui indique le nombre d'occurrences pour chaque élément du scénario. Une valeur numérique inscrite en face de l'élément indique le nombre de fois que celui-ci apparaît dans le fichier de scénario pédagogique.

Sur la figure 4.21, le programme indique que le scénario pédagogique examiné contient deux objets "*learning-objectives*" et deux objets "*prerequisites*". Cependant les règles de

construction données à l'annexe B.3 empêchent un scénario pédagogique de contenir plus d'un (01) objet "*learning-objectives*" et plus d'un (01) objet "*prerequisites*". La fenêtre de résultats indique également que le scénario contient des objets "*ACT*" qui n'ont pas d'objets "*ROLE-PART*" associés. Ceci constitue une erreur car les règles de construction précisant qu'à chaque objet de type "*ACT*", il doit y avoir au moins un objet "*ROLE-PART*" associé.

Lorsqu'un fichier de scénario pédagogique est soumis à une analyse syntaxique et ne contient aucune erreur, la fenêtre des résultats affiche des résultats comme le montre la figure 4.22.

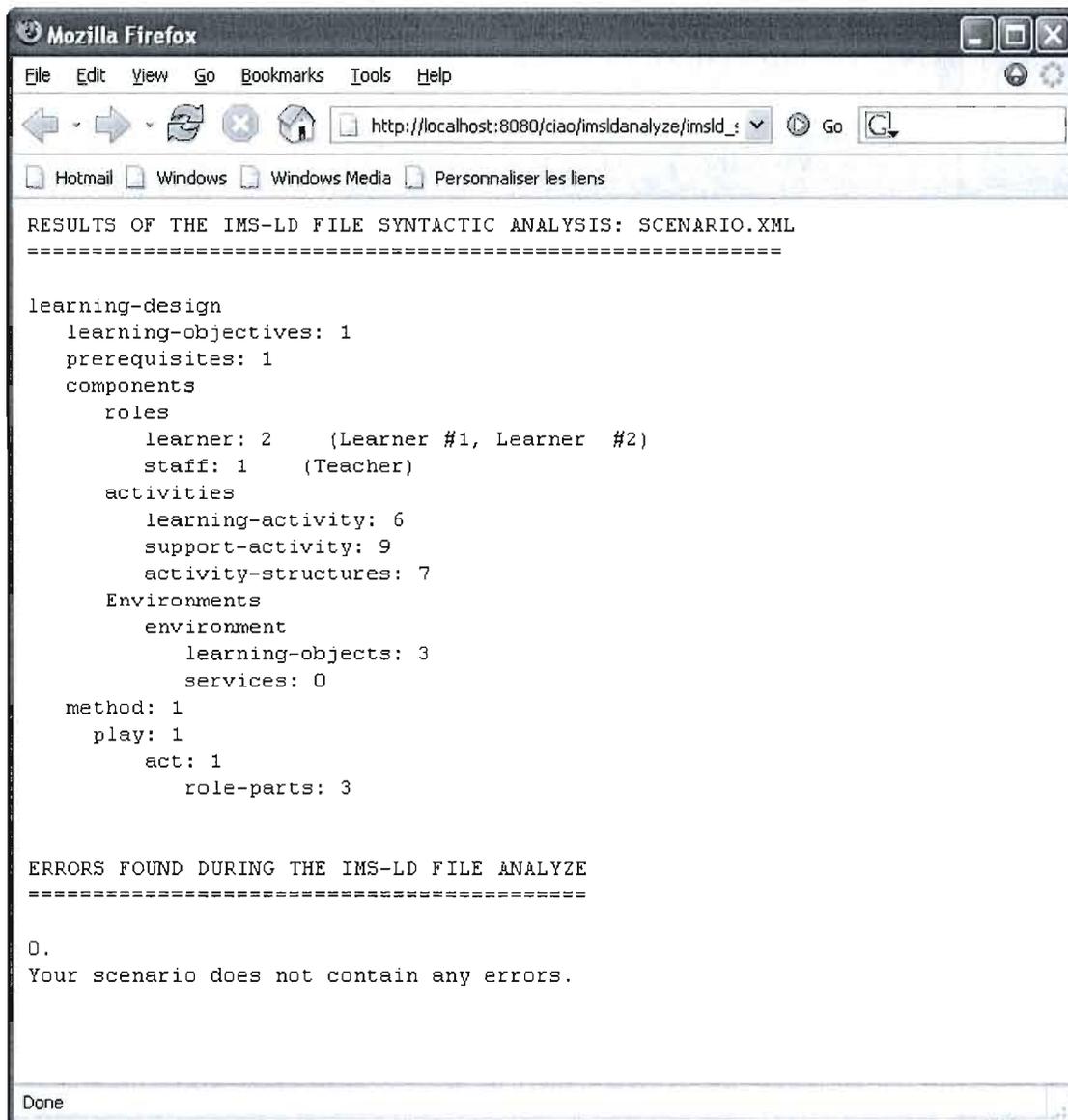
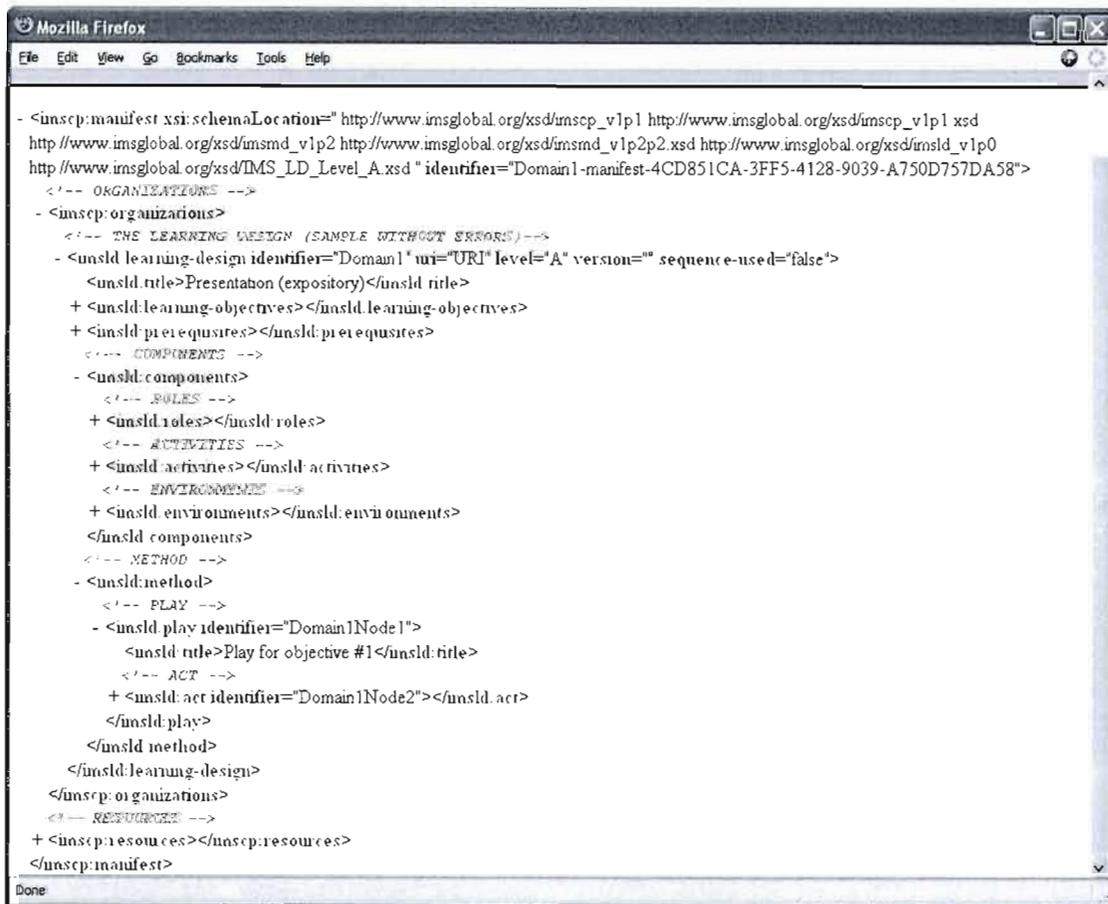


Figure 4.22: Résultats d'une analyse syntaxique d'un fichier sans erreurs.

En plus de l'analyse syntaxique, l'utilisateur peut effectuer une analyse sémantique afin d'observer les recommandations faites par le système en fonction du contenu du scénario pédagogique.

2- Analyse sémantique

Cette analyse n'est effectuée que si le fichier de scénario pédagogique soumis ne contient aucune erreur de syntaxe. Si le fichier contient des erreurs, le Système affiche un message d'erreur et indique que la validation ne peut se poursuivre. La figure 4.23 suivante montre un scénario pédagogique syntaxiquement correct où sera effectuée une analyse sémantique.



```

- <unscp:manifest xsi:schemaLocation="http://www.imsglobal.org/xsd/unscp_v1p1 http://www.imsglobal.org/xsd/unscp_v1p1 xsd
http://www.imsglobal.org/xsd/unscmd_v1p2 http://www.imsglobal.org/xsd/unscmd_v1p2p2.xsd http://www.imsglobal.org/xsd/unscmd_v1p0
http://www.imsglobal.org/xsd/TMS_LD_Level_A.xsd" identifier="Domain1-manifest-4CD851CA-3FF5-4128-9039-A750D757DA58">
  <!-- ORGANIZATIONS -->
  - <unscp:organizations>
    <!-- THE LEARNING DESIGN (SAMPLE WITHOUT ERRORS)-->
    - <unscd:learning-design identifier="Domain1" uri="URI" level="A" version="" sequence-used="false">
      <unscd:title>Presentation (expository)</unscd:title>
      + <unscd:learning-objectives></unscd:learning-objectives>
      + <unscd:prerequisites></unscd:prerequisites>
      <!-- COMPONENTS -->
      - <unscd:components>
        <!-- ROLES -->
        + <unscd:roles></unscd:roles>
        <!-- ACTIVITIES -->
        + <unscd:activities></unscd:activities>
        <!-- ENVIRONNEMENTS -->
        + <unscd:environments></unscd:environments>
      </unscd:components>
      <!-- METHOD -->
      - <unscd:method>
        <!-- PLAY -->
        - <unscd:play identifier="Domain1Node1">
          <unscd:title>Play for objective #1</unscd:title>
          <!-- ACT -->
          + <unscd:act identifier="Domain1Node2"></unscd:act>
        </unscd:play>
      </unscd:method>
    </unscd:learning-design>
  </unscp:organizations>
  <!-- RESOURCES -->
  + <unscp:resources></unscp:resources>
</unscp:manifest>

```

Figure 4.23: Exemple de scénario pédagogique sans erreurs.

La figure 4.24 illustre la fenêtre de résultats apparaissant suite à l'analyse sémantique effectuée sur un scénario sans erreurs tel que celui présenté à la figure 4.23. Les recommandations sont faites en tenant compte des éléments fournis à l'annexe B.4.

```

Mozilla Firefox
File Edit View Go Bookmarks Tools Help
RESULTS OF THE IMS-LD FILE SEMANTIC ANALYSIS: SCENARIO.XML
=====
learning-design
  learning-objectives: 1
  prerequisites: 1
  components
    roles
      learner: 2      (Learner #1, Learner #2)
      staff: 1       (Teacher)
    activities
      learning-activity: 6
      support-activity: 9
      activity-structures: 7
    Environments
      environment
        learning-objects: 3
        services: 0
  method: 1
  play: 1
  act: 1
  role-parts: 3

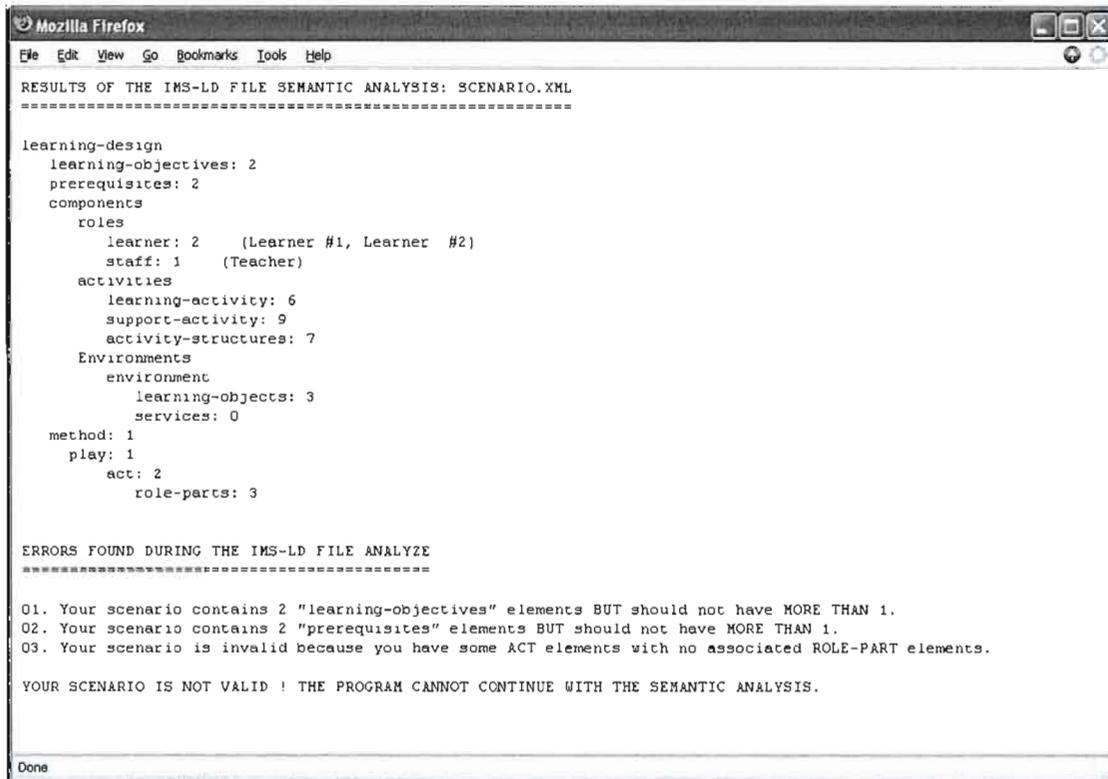
ERRORS FOUND DURING THE IMS-LD FILE ANALYZE
=====
0.
Your scenario does not contain any errors.

SOME REMARKS FOR YOUR THE IMS-LD FILE
=====
01. You have some activities IN PARALLEL and some others IN SEQUENCE.
02. Your scenario seems to be more TEACHER-CENTERED.
   Here is a non-exhaustive list of educational theory which integrate this concept:
   -> GARDNER THEORY
   -> GAGNE THEORY
   -> MERILL THEORY.
Done

```

Figure 4.24: Résultat d'une analyse sémantique d'un fichier exempt d'erreurs.

Dans le cas d'un fichier contenant des erreurs, tel que celui présenté à la figure 4.24, CIAO affiche un message d'erreur. Ce message est affiché à la suite des erreurs de syntaxe. Autrement dit, lorsque l'utilisateur demande une analyse sémantique, CIAO effectue préalablement une analyse syntaxique afin de s'assurer que le fichier ne contient aucune erreur. Un exemple de message d'erreur affiché pour une analyse sémantique est fourni à la figure 4.25.



```
RESULTS OF THE IMS-LD FILE SEMANTIC ANALYSIS: SCENARIO.XML
=====

learning-design
learning-objectives: 2
prerequisites: 2
components
  roles
    learner: 2 (Learner #1, Learner #2)
    staff: 1 (Teacher)
  activities
    learning-activity: 6
    support-activity: 9
    activity-structures: 7
  Environments
    environment
      learning-objects: 3
      services: 0
method: 1
play: 1
  act: 2
    role-parts: 3

ERRORS FOUND DURING THE IMS-LD FILE ANALYZE
=====

O1. Your scenario contains 2 "learning-objectives" elements BUT should not have MORE THAN 1.
O2. Your scenario contains 2 "prerequisites" elements BUT should not have MORE THAN 1.
O3. Your scenario is invalid because you have some ACT elements with no associated ROLE-PART elements.

YOUR SCENARIO IS NOT VALID ! THE PROGRAM CANNOT CONTINUE WITH THE SEMANTIC ANALYSIS.

Done
```

Figure 4.25: Résultat de l'analyse sémantique effectuée sur un fichier erroné.

4.3.6. Le service d'exportation

Ce service permet à l'utilisateur d'exporter le contenu de l'ontologie dans un fichier selon différents formats. L'utilisateur a ainsi la possibilité d'exporter l'ontologie dans un format RDF/XML ou encore sous forme de Triplets RDF. Selon que l'utilisateur se trouve dans le répertoire des classes ou des instances, une fenêtre différente lui est proposée pour l'extraction des fichiers. La figure 4.26 suivante montre la fenêtre apparaissant lorsque l'utilisateur décide d'extraire le contenu de l'ontologie à partir du répertoire des classes. L'utilisateur définit les différents éléments qu'il souhaite extraire en cochant les cases appropriées et choisit sous quelle forme les données seront représentées à la sortie.

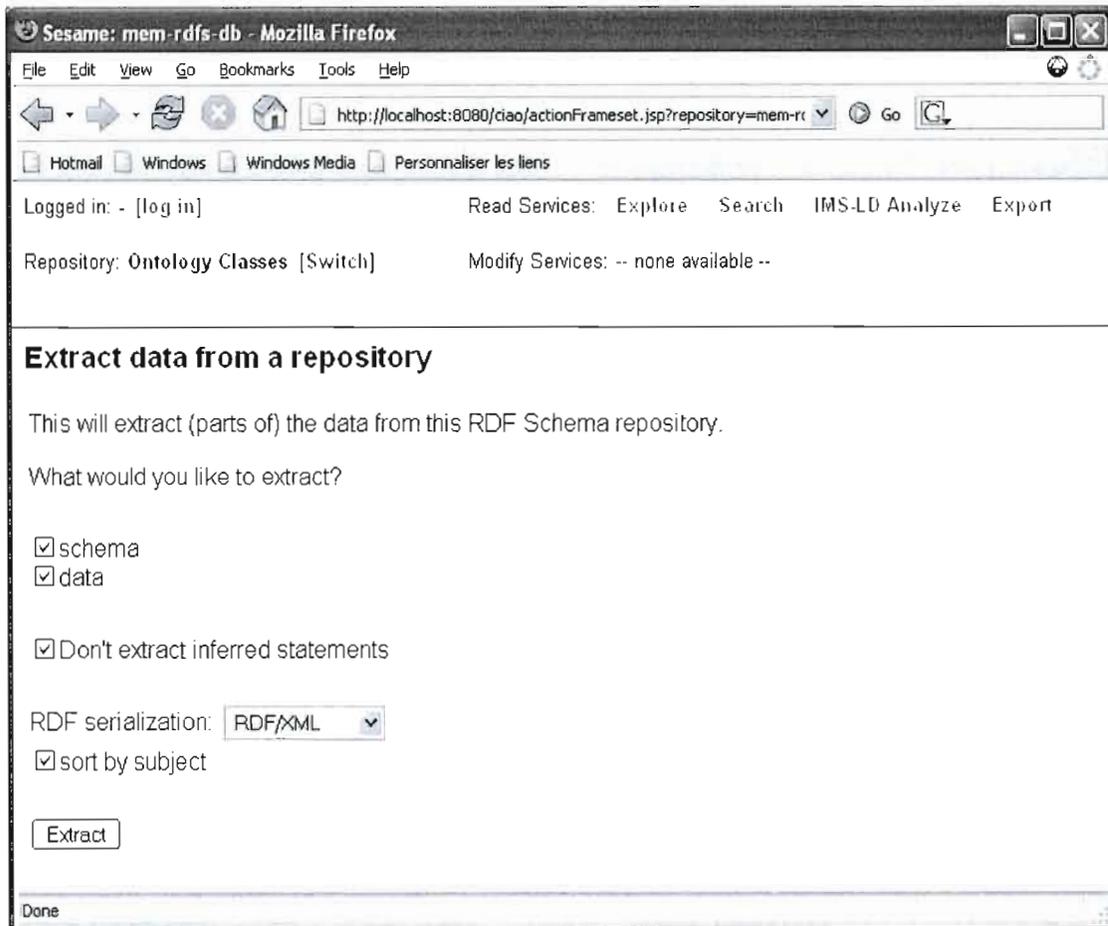


Figure 4.26: Extraction du contenu de l'ontologie dans le répertoire des classes.

Lorsque l'utilisateur se trouve dans le répertoire des instances, la fenêtre proposée est celle illustrée sur la figure 4.26 a.

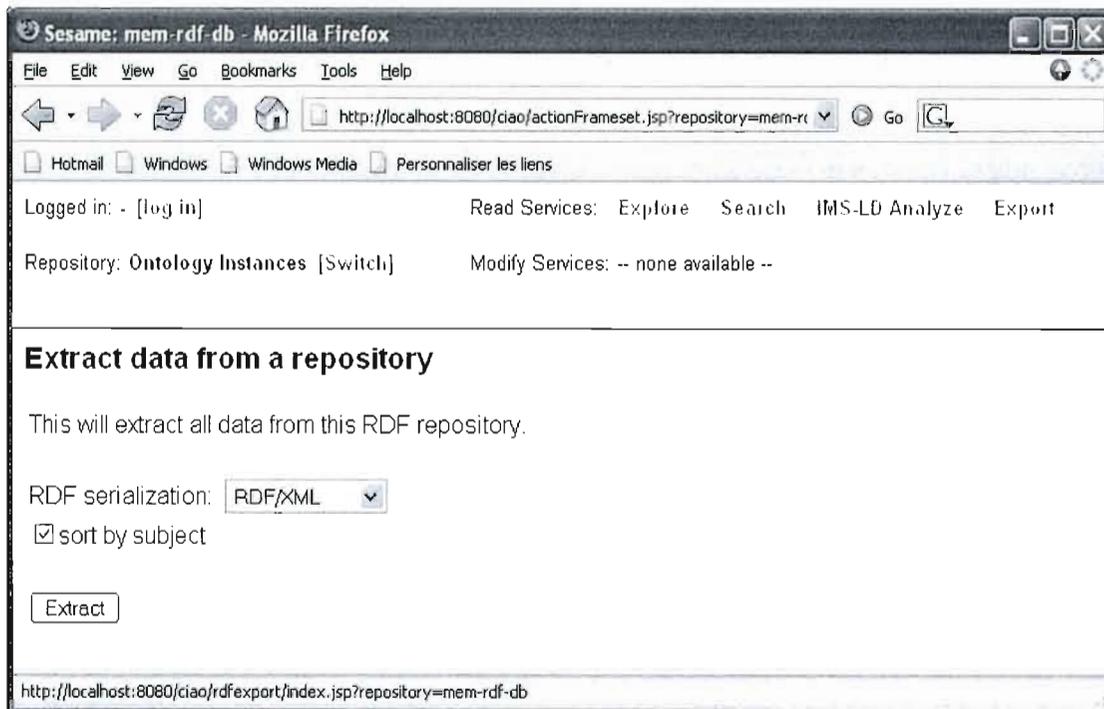


Figure 4.26a: Extraction du contenu de l'ontologie dans le répertoire des instances.

L'utilisateur n'a pas le choix des éléments à extraire car ces éléments sont les données relatives aux différentes instances des objets de l'ontologie créés lors de la construction de l'ontologie.

4.3.7. Le service d'écriture

Ce service permet à l'utilisateur de modifier le contenu de l'ontologie. Pour pouvoir y accéder l'utilisateur doit s'identifier auprès du système et posséder les droits requis. Dans la version actuelle de CIAO, cette option n'est accessible qu'à un utilisateur nommé *xldesigner*. La figure 4.27 suivante présente la fenêtre d'authentification de CIAO à laquelle il est possible d'accéder en tout temps en cliquant sur l'option *Log in* située en haut dans la barre de menus.

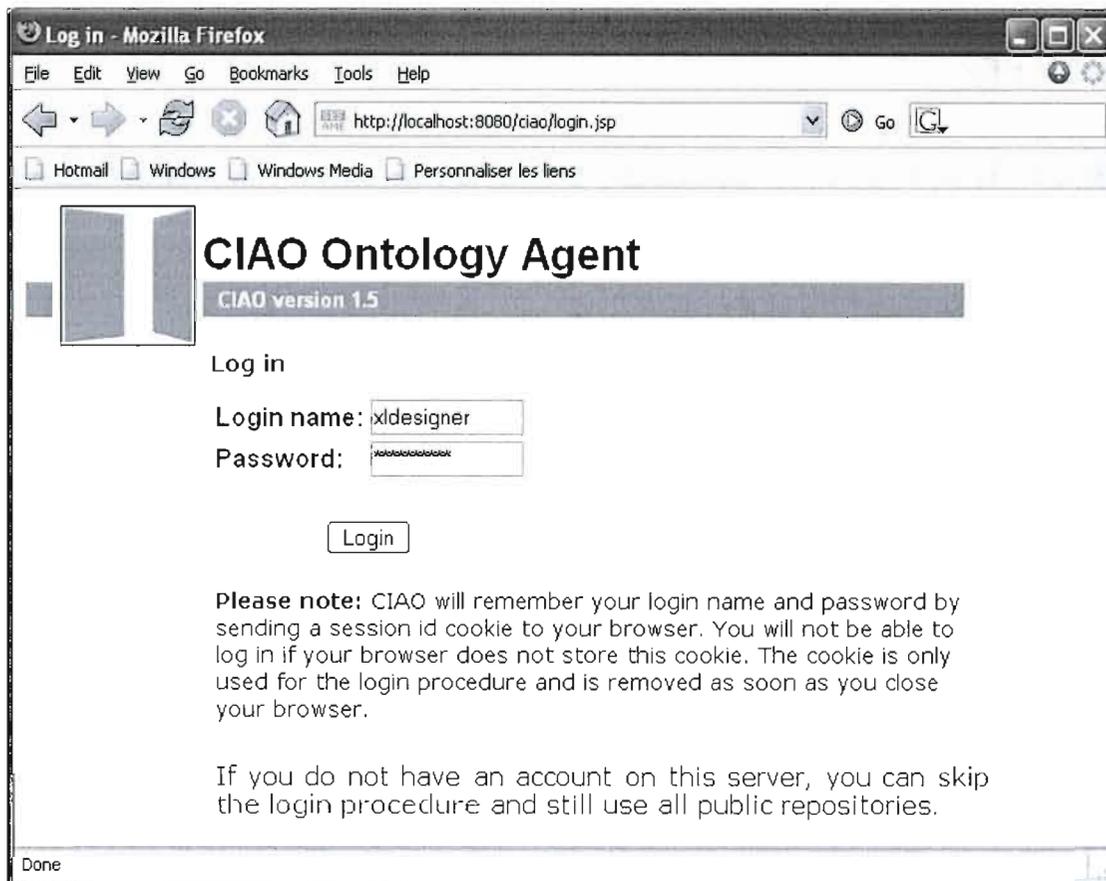


Figure 4.27: Processus d'identification d'un usager.

Une fois l'authentification effectuée, les services d'écriture apparaissent dans la barre d'outils de CIAO en face du texte "**Modify services**". Ces services sont au nombre de cinq et portent respectivement les noms de **Add(file)**, **Add(copy-paste)**, **Add(www)**, **Remove** et **Clear**. Les trois premiers permettent d'ajouter de différente manière des données supplémentaires au contenu de l'ontologie. La première manière permet de localiser un fichier sur le disque afin que le contenu soit extrait dans l'ontologie à la suite des données préexistantes. La deuxième option présente une zone dans laquelle on peut coller du texte à rajouter au contenu de l'ontologie. La troisième permet de fournir une adresse Web où se trouvent les données à rajouter dans l'ontologie. Les trois prochaines figures illustrent respectivement chacune des trois options de modification de l'ontologie.

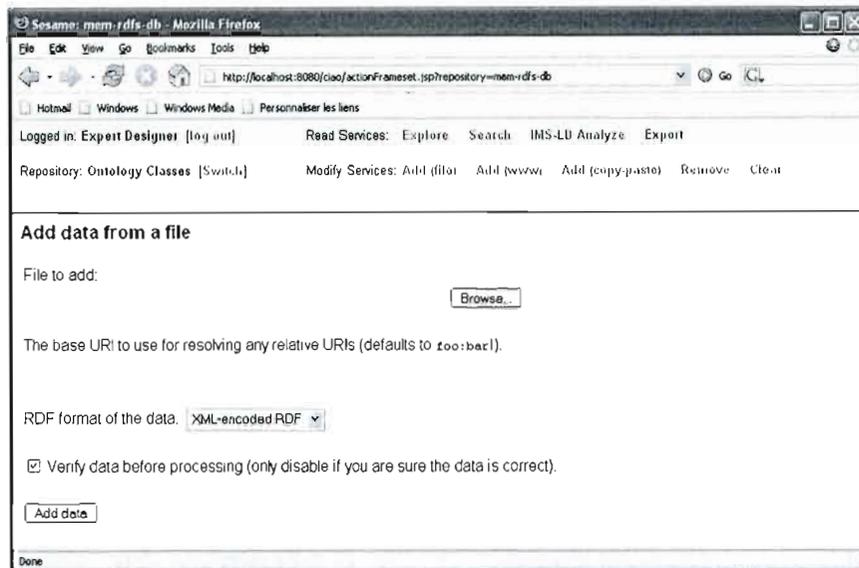


Figure 4.28: Mise à jour de l'ontologie à partir d'un fichier.

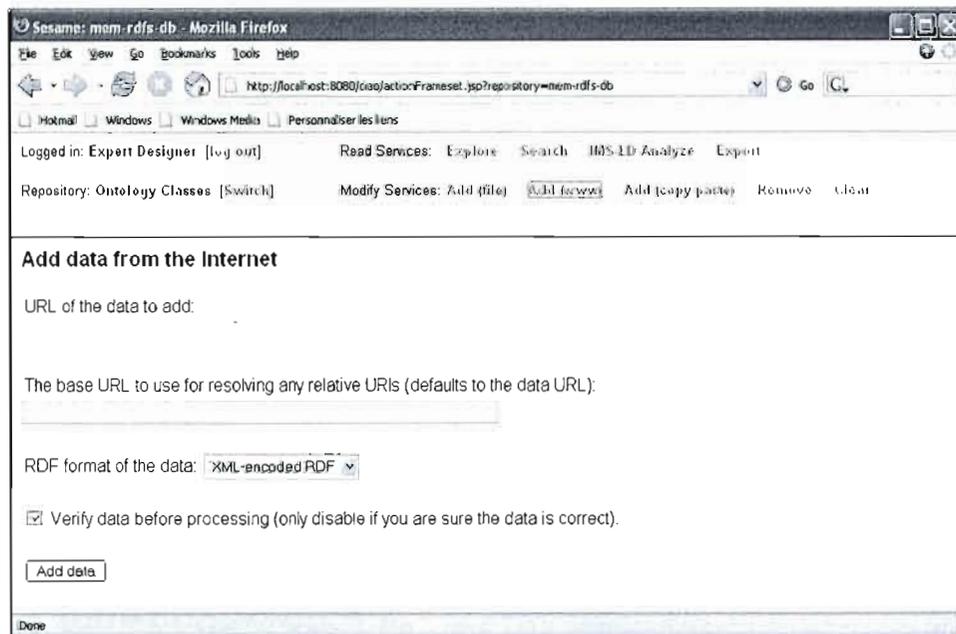


Figure 4.29: Mise à jour de l'ontologie à partir d'une adresse Web.

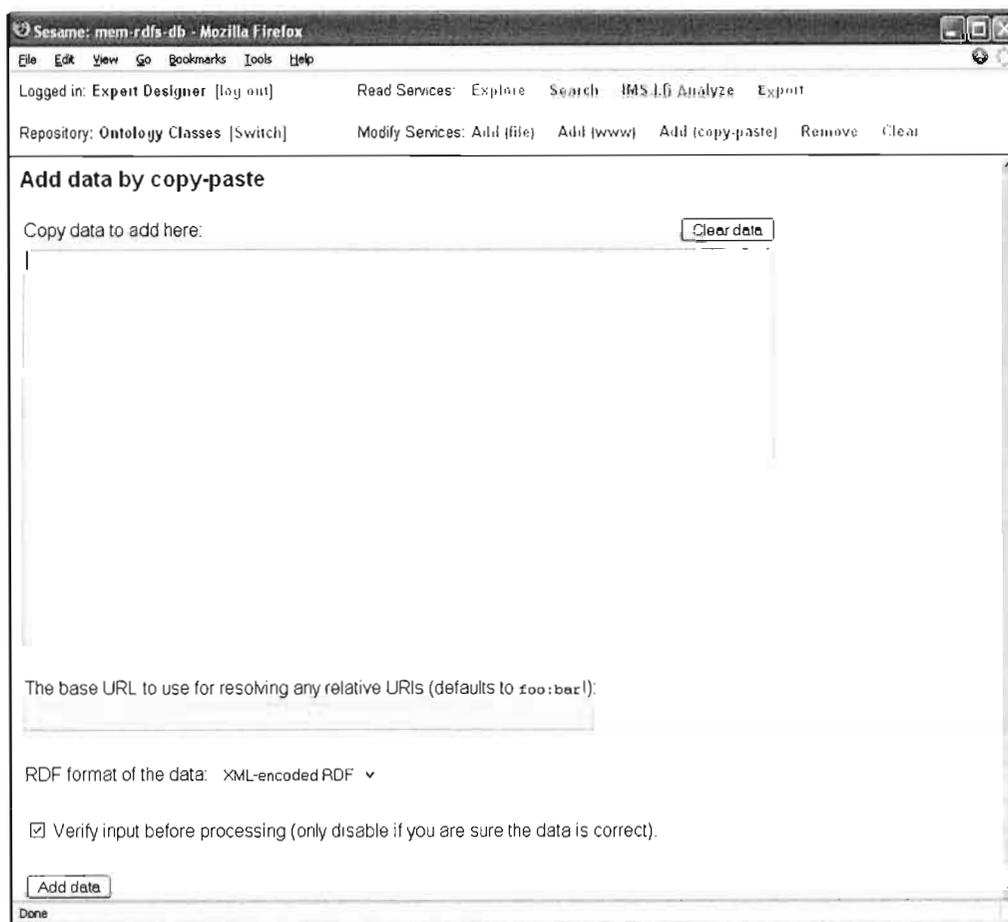


Figure 4.30: Mise à jour de l'ontologie à partir d'une opération de copier-coller.

La commande de modification **Remove** permet à l'utilisateur d'enlever un triplet RDF donné de l'ontologie. Afin de pouvoir exécuter cette opération, l'utilisateur doit préciser les composantes de tout triplet RDF, à savoir, le sujet, le prédicat et l'objet. La figure 4.31 illustre la fenêtre d'exécution de cette commande.

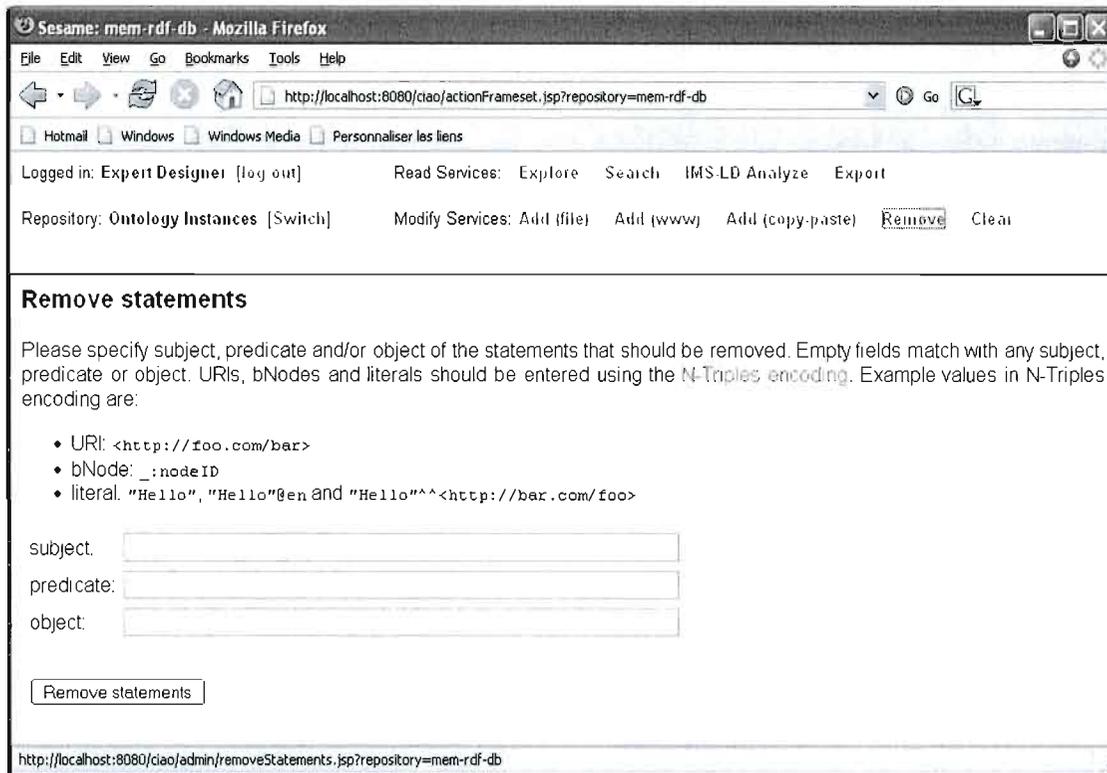


Figure 4.31: Suppression d'éléments dans l'ontologie des théories.

La commande *Clear* permet de vider le contenu de l'ontologie hors du répertoire dans lequel se trouve l'utilisateur. Lorsque l'utilisateur sélectionne cette commande, une fenêtre de confirmation apparaît et permet à l'utilisateur de valider son choix. Une fois cette opération effectuée, l'ontologie est complètement vidée de son contenu. La figure 4.32 suivante présente la fenêtre de confirmation pour le choix de la commande *Clear* effectuée sur le répertoire des instances.

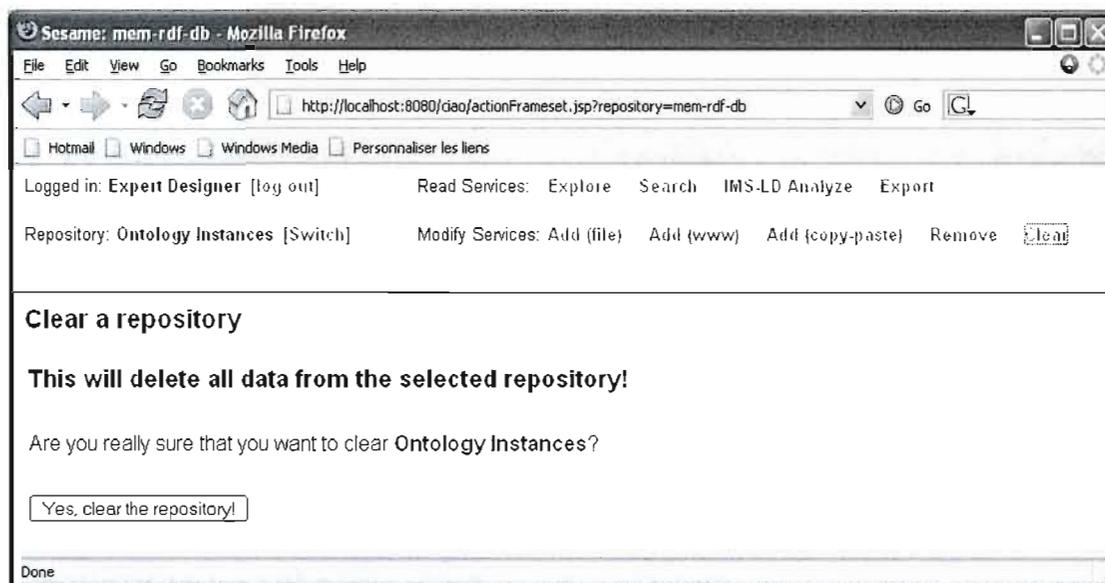


Figure 4.32: Confirmation de la suppression du contenu entier de l'ontologie.

4.4 Évaluation de CIAO par des concepteurs

Suite aux différents travaux effectués dans le cadre du projet CIAO, nous avons fait évaluer notre système par des experts du centre LICEF en ce qui a trait aux activités de DP. Ces évaluations ont permis de recueillir des commentaires d'amélioration le système. Elles ont été réalisées du 21 septembre 2005 au 23 septembre 2005 dans les locaux du LICEF. Pour cette activité, cinq experts ont été interrogés sur les différentes fonctionnalités du système CIAO. Voici les noms des chercheurs : Michel Léonard, Karin Lundgren, Marcelo Maïna, Diane Ruelland et Isabelle Savard.

L'évaluation a d'abord porté sur l'interface utilisateur de l'application, les commandes et les barres de menus, l'interopérabilité du système avec ceux utilisés pour la construction des ontologies, l'exportation de données et la vitesse d'exécution du système.

Par la suite les différents services de CIAO ont été évalués de façon individuelle. Ce sont : l'exploration, la recherche, l'analyse et l'exportation. Les critiques des chercheurs sur ces services ont permis d'améliorer des interfaces et les fonctionnalités du système.

De façon générale, les concepteurs (experts) ont noté une utilité claire pour le système CIAO pour ce qui concerne les activités qu'ils effectuent au niveau du DP. L'attention a particulièrement été portée sur le service d'analyse et validation qui fait entrevoir une conception plus "intelligente" de scénarios pédagogiques. CIAO permet ainsi d'offrir l'assistance requise aux concepteurs, ce qui leur donne une plus grande assurance durant les activités de DP, et vient ainsi palier le manque d'informations (requis pour la validation) dans les systèmes qu'ils utilisent.

CHAPITRE V

DISCUSSION ET CONCLUSION

Nous récapitulons dans la première partie les résultats de notre travail. Nous présentons dans la seconde partie les perspectives de recherches débouchant sur les améliorations à apporter à notre modèle.

5.1 Synthèse

Notre but était d'obtenir un outil d'assistance pouvant offrir une assistance aux concepteurs durant leurs tâches de DP, en se servant de l'"intelligence" offerte par une base de connaissances basées sur les ontologies des théories, de l'apprentissage et du DP. Notre travail a consisté à :

- proposer l'architecture d'un système pour l'intégration de l'assistance dans les activités de DP à travers l'utilisation de théories pertinentes ;
- explorer les possibilités de mise en œuvre du système;
- concevoir et mettre en œuvre les services associés à l'outil retenu.

Dans un premier temps, nous avons exploré les concepts de systèmes d'apprentissage en ligne, de systèmes auteurs (SA) et d'ingénierie ontologique pour situer notre projet par

rapport à ces différents éléments. Cette étude a fait l'objet du chapitre 2 où sont aussi présentées les limites actuelles des systèmes auteurs en ce qui concerne l'assistance offerte durant les activités de DP. Nous avons également traité du rôle que peut jouer l'ingénierie ontologique dans le téléapprentissage. Ce chapitre a permis de relever différents aspects de la problématique associée à l'intégration de l'assistance tout au long des activités effectuées par les concepteurs durant le DP.

Dans le chapitre 3, nous avons examiné les ontologies des théories qui servent de bases de connaissance pour la production de notre système, de même que les scénarios produits par les SA durant les activités de DP. On y trouve également le lien entre les ontologies et les SA à travers leurs scénarios en vue de palier le manque d'assistance. Ce chapitre présente notre proposition de système pour l'intégration de cette assistance durant les activités de DP. Le système proposé repose sur deux éléments : (1) une base de connaissances regroupant les théories du DP et l'intelligence qu'elles contiennent, et (2) des informations contenues dans les fichiers de scénarios pédagogiques obtenus suite à l'utilisation des SA par les concepteurs.

Il est possible d'apporter des améliorations pour ajouter à la fiabilité et à la robustesse du système. Elles permettront aux concepteurs de procéder à leurs activités de Design Pédagogique avec plus d'assurance. En effet, en l'état actuel des développements, l'ensemble des fonctionnalités des différents services est fonctionnel. L'analyse et la validation des scénarios est également fonctionnelle et offre une validation effectuée sur le poste client. Dans la section qui suit, nous exposons les possibles perspectives de recherche et les développements à faire pour que le système soit à 100% utilisable.

5.2 Perspectives

La finalité de ce travail comme tout travail effectué dans un contexte de recherche-développement est l'utilisation des résultats produits. Pour y arriver, des améliorations sont à apporter. Nous avons dans la dernière version de l'outil, considéré essentiellement deux types d'utilisateurs : les utilisateurs "anonymes" qui ne se font pas authentifier par le système et qui n'ont pas besoin d'effectuer des modifications sur le contenu de nos bases de connaissances et un utilisateur expert disposant de ce droit, afin de modifier

l'ontologie. La conséquence est que nos ontologies ont gérées de façon centrale par un administrateur et ne peuvent donc en aucun cas être mises à jour par un usager particulier. Nous devons donc apporter des améliorations à ce niveau, une fois qu'une version très avancée de l'ontologie des théories du DP aura été élaborée. Nous envisageons ainsi de permettre à chaque utilisateur de gérer de façon indépendante une ontologie sur son poste en y ajout les éléments qu'il juge utile dans ses activités. Toutefois l'usager aurait toujours la possibilité de se servir de l'ontologie de base qui sera toujours gérée par un administrateur.

Nous envisageons d'étudier la présentation des résultats fournis suite à une analyse sémantique de façon à ce que l'utilisateur puisse s'en servir pour effectuer une nouvelle exploration de l'ontologie. Ainsi, lorsque le système recommandera des théories particulières à utiliser dans une conception donnée, l'usager aura la possibilité d'explorer chacune de ces théories en fonction de sa documentation dans la base de connaissances. En intégrant une telle fonctionnalité, le service de validation sera grandement amélioré et la pertinence des théories développées dans notre ontologie sera mise de l'avant. Ces théories augmenteraient ainsi le support dont a besoin l'usager.

Nous avons abordé dans ce travail le problème de l'assistance fournie durant les activités de DP dans l'optique de mettre en place un outil permettant de mener ces activités à bien. Nous avons également profité de la présence d'experts dans les locaux du LICEF pour bénéficier de commentaires et d'améliorations pouvant être apportées, en faisant des séances d'évaluation du système. Nous n'avons pas pensé à la possibilité de permettre aux experts et aux utilisateurs en général de pouvoir nous fournir de façon permanente leurs avis suite à l'utilisation de l'outil. Etant donné que nous rendons CIAO disponible à partir des serveurs Web du centre LICEF, nous pourrions offrir une interface conviviale à partir de laquelle les concepteurs pourraient passer des commentaires en tout temps. Cela permettra également de créer une sorte de forum de discussion autour des fonctionnalités de l'outil afin que tout concepteur puisse faire valoir ses idées et recueillir également celles des autres.

Dans la suite du développement du système CIAO, nous allons travailler à l'intégration des fonctionnalités de CIAO à celles d'un système auteur. Dans un premier temps, nous

tenterons d'associer CIAO à l'éditeur de scénarios MOT³⁴. MOT est un logiciel permettant de représenter graphiquement des connaissances dans divers champs de savoir et de mettre en évidence les relations existant entre elles. Le logiciel possède des capacités graphiques sophistiquées. Le lien qui sera fait avec notre système CIAO permettra essentiellement à un concepteur de soumettre un scénario pédagogique construit avec MOT directement à CIAO afin qu'une validation y soit effectuée. Ce service constituera ainsi une fonctionnalité additionnelle de notre service d'analyse et de validation offert par CIAO.

³⁴ http://www.cogigraph.com:90/cogigraph/article.php3?id_article=16

ANNEXE A

CAS D'UTILISATIONS ASSOCIÉS AU SERVICE D'ANALYSE SYNTAXIQUE ET SÉMANTIQUE.

Dans cette annexe, nous présentons les cas d'utilisation traitant du service d'analyse syntaxique et du service d'analyse sémantique offert par le système CIAO.

A.1 Cas d'utilisation "Effectuer une validation syntaxique sur un fichier de scénario pédagogique"

Tableau A.1: Cas d'utilisation de l'analyse syntaxique

| | |
|---|--|
| Acteur principal : L'utilisateur du système CIAO (qui est un concepteur d'activités pédagogiques) | |
| Parties prenantes et intérêts : | |
| L'utilisateur du système : il veut savoir si son scénario est syntaxiquement correct, et le cas échéant, obtenir la liste des erreurs qu'il renferme. | |
| Le système CIAO : il veut offrir à l'utilisateur le service d'analyse syntaxique. | |
| Préconditions : Aucune. | |
| Garanties de succès (postconditions) : L'analyse est effectuée. Le concepteur dispose de l'assistance requise pour modifier à souhait son scénario. | |
| Scénario principal (succès) : | |
| Action Acteur (ou intention) | Responsabilités du système |
| 1. L'utilisateur identifie le fichier à analyser. | |
| 2. L'utilisateur active l'option d'analyse syntaxique. | 3. Le Système valide les données présentes à l'intérieur du fichier. |
| | 4. Le Système dresse la structure du scénario et identifie le nombre d'occurrences pour chacun des éléments du scénario. |
| | 5. Le Système affiche le résultat de l'analyse. |

A.2 Cas d'utilisation "Effectuer une validation sémantique sur un fichier de scénario pédagogique"

Tableau A.2: Cas d'utilisation de l'analyse sémantique

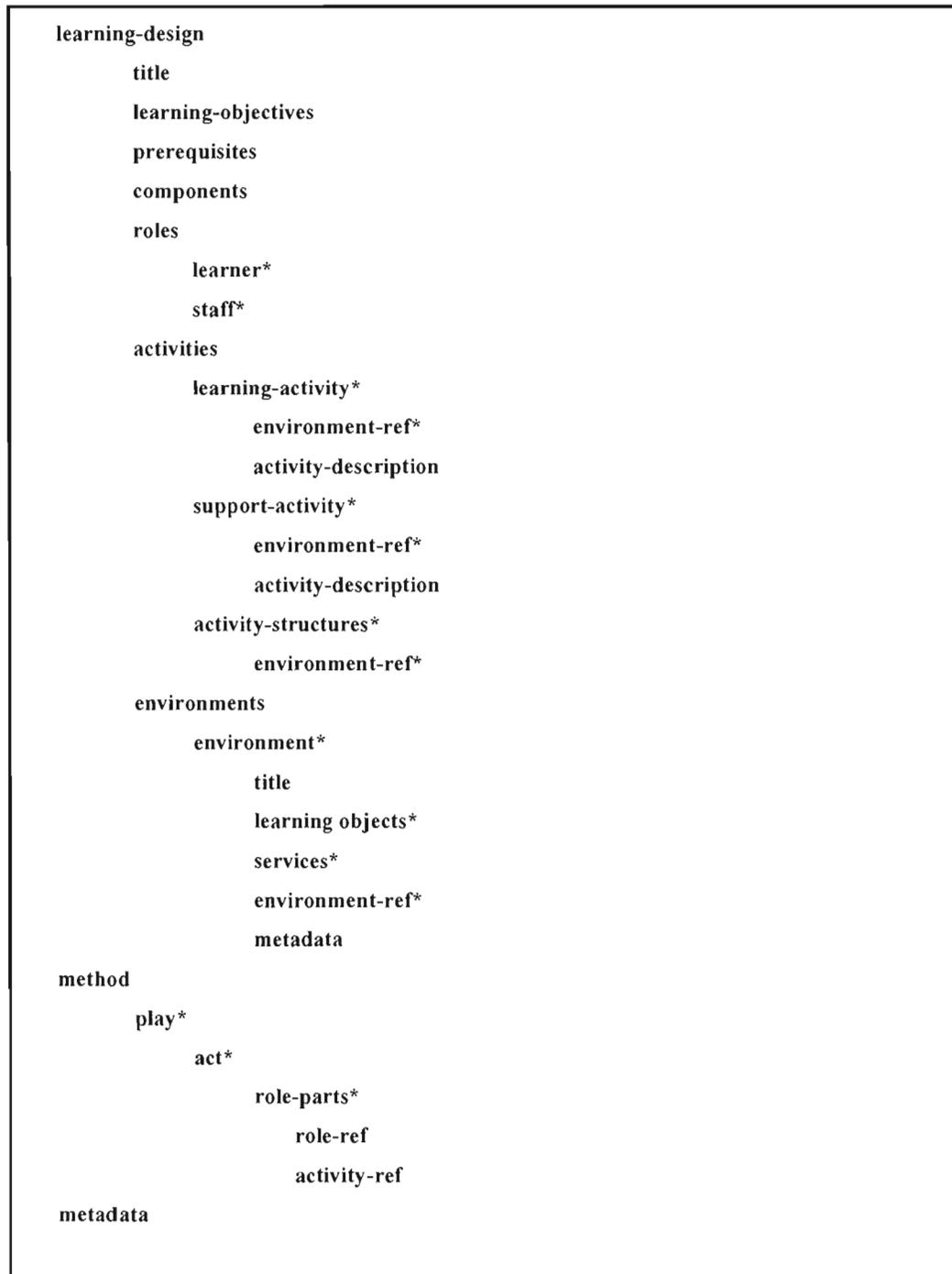
| | | |
|---|--|---|
| Acteur principal : L'utilisateur du système CIAO (qui est un concepteur d'activités pédagogiques) | | |
| Parties prenantes et intérêts : L'utilisateur du système : il veut obtenir des recommandations du Système pour la conception des activités mentionnées par son scénario pédagogique. Le système CIAO : il veut offrir à l'utilisateur le service d'analyse sémantique. | | |
| Préconditions : Aucune Garanties de succès (postconditions) : L'analyse est effectuée. Le concepteur obtient les recommandations du Système.. | | |
| Scénario principal (succès) : | | |
| Action Acteur (ou intention) | | Responsabilités du système |
| 1. | L'usager identifie le fichier à analyser. | |
| 2. | L'usager active l'option d'analyse sémantique. | |
| | | 3. Le Système valide les données présentes à l'intérieur du fichier. 4. Le Système dresse la structure du scénario et identifie le nombre d'occurrences pour chacun des éléments du scénario. 5. Le Système affiche les recommandations effectuées à partir du contenu du scénario pédagogique. |
| Extensions (ou scénarios alternatifs) : | | |
| 5a. | Le fichier de scénario contient des erreurs dans sa construction syntaxique. | 1. Le Système dresse la liste des erreurs contenues dans le fichier. 2. Le Système indique à l'usager que son scénario n'est pas syntaxiquement correct et que l'analyse sémantique ne peut pas se poursuivre. |

ANNEXE B

STRUCTURE HIÉRARCHIQUE GLOBALE D'UN FICHIER DE SCÉNARIO PÉDAGOGIQUE ET RÈGLES DE CARDINALITÉ

Dans cette annexe, nous présentons dans un premier temps la structure hiérarchique globale d'un fichier de scénario pédagogique. Par la suite, nous présentons les règles de cardinalité qui sont vérifiées durant le service d'analyse de CIAO. Ces règles sont utilisées autant pour l'analyse syntaxique que pour l'analyse sémantique. Nous terminons en présentant les autres types d'erreurs vérifiées durant les différentes analyses et les règles de formulation des recommandations utilisées pour l'analyse sémantique.

B.1 Structure hiérarchique globale d'un fichier de scénario pédagogique



B.2 Règles de cardinalité pour l'analyse syntaxique

Éléments obligatoires

Tableau B.1: Éléments obligatoires validés dans le scénario pédagogique

| Mandatory elements | Occurrence | Excerpt of the XML manifest |
|--------------------|------------|--|
| Act | 1..n | <imsld:act identifier="Domain1Node2"> <imsld:title>Act #1</imsld:title> |
| Learner role | 1..n | |
| Method | 1 | <imsld:method> |
| Play | 1..n | <imsld:play identifier="Domain1Node1"> <imsld:title>Play #1</imsld:title> |
| Role-part | 1..n | - <imsld:role-part> <imsld:role-ref ref="Domain1Node7" /> <imsld:learning-activity-ref ref="Domain1Node5" /> </imsld:role-part> |

Éléments recommandés et éléments optionnels

Tableau B.2: Éléments optionnels validés dans le scénario pédagogique

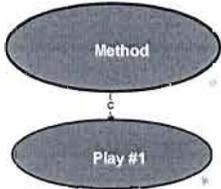
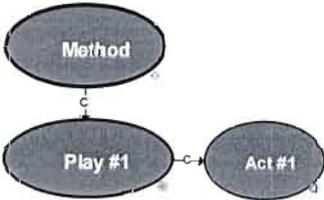
| Recommended elements | Occurrence | Excerpt of the XML manifest |
|----------------------|------------|---|
| Learning Activity | 0..n | <imsld:learning-activity identifier="Domain1Node5"> <imsld:title>Learning Activity #1</imsld:title> <imsld:environment-ref ref="Domain1Node9" /> <imsld:activity-description> <imsld:item /> </imsld:activity-description> </imsld:learning-activity> |
| Activity Structure | 0..n | <imsld:activity-structure identifier="Domain1Node4" structure-type="selection" number-to-select="2"> <imsld:title>Activity Structure #1</imsld:title> <imsld:learning-activity-ref ref="Domain1Node5" /> <imsld:support-activity-ref ref="Domain1Node6" /> </imsld:activity-structure> |
| Environment | 0..n | <imsld:environment identifier="Domain1Node9"> <imsld:title>Environment #1</imsld:title> |

| Recommended elements | Occurrence | Excerpt of the XML manifest |
|--------------------------|------------|--|
| Learning Object | 0..n | <pre> <imsl:learning-object identifier="Domain1Node12"> <imsl:title>Learning Object #1</imsl:title> <imsl:item identifier="Domain1Node12-Domain1Node14" identifierref="RES-Domain1Node14" /> </imsl:learning-object> </pre> |
| Learning objective | 0..n | <pre> <imsl:learning-objectives> <imsl:title>Learning objectives title</imsl:title> <imsl:item identifier="LOB-learning-objectives" identifierref="RES-learning-objectives"> <imsl:title>Learning objective title</imsl:title> </imsl:item> </imsl:learning-objectives> </pre> |
| Learning outcome | 0..n | <pre> <imsl:learning-object identifier="Domain1Node11"> <imsl:title>LO Outcome #2</imsl:title> <imsl:item /> </imsl:learning-object> </pre> |
| Optional elements | No | Tag |
| Prerequisite | 0..n | <pre> <imsl:prerequisites> <imsl:title>Prerequisites title</imsl:title> <imsl:item identifier="PREQ-prerequisites" identifierref="RES-prerequisites"> <imsl:title>Prerequisite title</imsl:title> </imsl:item> </imsl:prerequisites> </pre> |

| recommended elements (according to LICEF's LD experts) | Occurrence | Excerpt of the XML manifest |
|--|------------|--|
| Resource | 0..n | <imscp:resource identifier="RES-Domain1Node14" type="imsldcontent" href="Item #1"> |
| Service (Conference or send-mail or search-index) | 0..n | <pre> <!-- (SEND-MAIL) --> <imsld:environment identifier="Results_to_Recorder"> <imsld:title>Send Results to Recorder</imsld:title> <imsld:service identifier="mail_recorder"> <imsld:send-mail select="all-persons-in-role"> etc </imsld:send-mail> </imsld:service> </imsld:environment> <!-- (CONFERENCE) --> <imsld:environment identifier="Main_Negotiation_Confer"> <imsld:title>Main Negotiation Forum</imsld:title> <imsld:service identifier="Main_Negotiate_SO"> <imsld:conference conference-type="asynchronous"> <imsld:participant role-ref="Learner"/> <imsld:participant role-ref="Teacher"/> etc <imsld:item identifier="I-Main_Negotiate_SO"/> </imsld:conference> </imsld:service> </imsld:environment> </pre> |
| Staff role | 0..n | |
| Support Activity | 0..n | <pre> <imsld:support-activity identifier="Domain1Node6"> <imsld:title>Support Activity #1</imsld:title> <imsld:environment-ref ref="Domain1Node10" /> <imsld:activity-description> <imsld:item /> </imsld:activity-description> </imsld:support-activity> </pre> |

B.3 Autres erreurs vérifiées durant l'analyse syntaxique

Tableau B.3: Erreurs vérifiées durant l'analyse syntaxique

| Combination of elements | Error number | Error messages |
|--|--------------|--|
| <ul style="list-style-type: none"> - 1 method <ul style="list-style-type: none"> o 0 Play  | Er1 | <p>The scenario is NOT syntactically CORRECT.</p> <p>For each method, there must have AT LEAST:</p> <ul style="list-style-type: none"> - 1 play |
| <ul style="list-style-type: none"> - 1 method <ul style="list-style-type: none"> o 1 Play <ul style="list-style-type: none"> ▪ 0 Act  | Er2 | <p>The scenario is NOT syntactically CORRECT. For each play there must have AT LEAST:</p> <ul style="list-style-type: none"> - 1 act |
| <ul style="list-style-type: none"> - 1 act <ul style="list-style-type: none"> - 0 role-part  | Er3 | <p>The scenario is NOT syntactically CORRECT.</p> <p>For each act, there must have AT LEAST:</p> <ul style="list-style-type: none"> - 1 role-part |
| <ul style="list-style-type: none"> - 1 method <ul style="list-style-type: none"> - 0 "learner" role | Er4 | <p>The scenario is NOT syntactically CORRECT.</p> <p>There must be AT LEAST :</p> <ul style="list-style-type: none"> - 1 learner role |

B.4 Règles de formulation des recommandations pour l'analyse sémantique

Tableau B.4: Règles utilisées pour l'analyse sémantique

| Combination of elements | Error number | Recommendation messages |
|---|--------------|---|
| <p>Activities:</p> <ul style="list-style-type: none"> - n support-activity - n+3 learning-activity <p><i>Nous devons avoir un nombre d'activités axées sur l'apprenant qui soit supérieur d'au moins trois à celui des activités axées sur le formateur.</i></p> | Rc6 | <p>Your scenario seems to more learner-centered.</p> <p>Here are a non exhaustive list of educational theory which integrate this concept:</p> <ul style="list-style-type: none"> - Piaget Theory, - Collins Theory, - Bruner Theory. |
| <p>Activities:</p> <ul style="list-style-type: none"> - n+3 support-activity - n learning-activity <p><i>Nous devons avoir un nombre d'activités axées sur le formateur qui soit supérieur d'au moins trois à celui des activités axées sur l'apprenant.</i></p> | Rc7 | <p>Your scenario seems to more teacher-centered.</p> <p>Here are a non exhaustive list of educational theory which integrate this concept:</p> <ul style="list-style-type: none"> - Gardner Theory - Gagné Theoy - Merrill Theory, |
| <p>Activities:</p> <ul style="list-style-type: none"> - n+/-2 support-activity - n +/-2 learning-activity <p><i>Nous devons avoir un écart de plus ou moins 2 entre le nombre d'activités axées sur le formateur et le nombre d'activités axées sur l'apprenant.</i></p> | Rc8 | <p>Your scenario seems to be a combination of learner-centered and teacher-centered.</p> <p>Here are a non exhaustive list of educational theory which integrate this concept:</p> <ul style="list-style-type: none"> - ... |
| <p>Activities:</p> <ul style="list-style-type: none"> - n support-activity | Rc9 | <p>Your scenario seems to more team-based.</p> |

| Combination of elements | Error number | Recommendation messages |
|--|--------------|---|
| <ul style="list-style-type: none"> - n+3 learning-activity Role: <ul style="list-style-type: none"> - n Learner | | Here are a non exhaustive list of educational theory which integrate this concept: <ul style="list-style-type: none"> - Vygotsky's Theory, - Wenger Theory, ... |
| Activities: <ul style="list-style-type: none"> - Learning-activity - Support-activities Environment: <ul style="list-style-type: none"> - learning-object <p><i>Nous devons avoir, en plus des activités d'apprentissage et des activités de supports, des instances d'objet d'apprentissage (learning-object).</i></p> | Rc10 | Your scenario seems to belongs to the instructivist paradigm <p>Here are a non exhaustive list of educational theory which integrate this concept: ...</p> |
| Activities: <ul style="list-style-type: none"> - Learning-activity - Support-activities Environment: <ul style="list-style-type: none"> - n services email and conference Role: <ul style="list-style-type: none"> - n Learner <p><i>L'environnement d'apprentissage doit faire intervenir des services de email et/ou de conférence. Si tel est le cas, nous avisons le concepteur qu'il se trouve dans une situation de scénario de type socioconstructiviste.</i></p> | Rc11 | Your scenario seems to belongs to the socio-constructivist paradigm <p>Here are a non exhaustive list of educational theory which integrate this concept:</p> |

BIBLIOGRAPHIE

- (Anderson et Pelletier, 1991) Anderson, J. R. et Pelletier, R. 1991. « A development system for model-tracing tutors ». *Proc. of the International Conference on the Learning Sciences*, Evanston, IL, pp. 1-8.
- (Bosse, 1988) Boose, J. H. 1988. « A Survey of Knowledge Acquisition Techniques and Tools. » *3rd AAAI Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop*, November 1988, pg. 3.1-3.23. Banff, Canada.
- (Bourdeau et Bates, 1996) Bourdeau, J. et Bates, A. 1996. «Instructional Design for Distance Learning», in *Journal of Science Education and Technology*
- (Bourdeau et Mizoguchi, 2002) Bourdeau J. et Mizoguchi R. 2002. « Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment », In Cerri, S., Gouardères, G. & Paraguaçu, F., *Intelligent Tutoring Systems*. Springer, Heidelberg: Lecture Notes in Computer Science, 399-409.
- (Bourdeau et Mizoguchi, 2002) Bourdeau, J. et Mizoguchi, R. 2002. « Theory-aware Authoring Environment: an Ontological Engineering Approach », *Proc. of the ICCE Conference*, 3-6 Dec. 2002, Auckland, New Zealand.
- (Bourdeau, Mizoguchi, Psyché et Nkambou, 2004) Bourdeau, J., Mizoguchi, R. , Psyché, V. et Nkambou, R. 2004. « Selecting Theories in an Ontology-Based ITS Authoring Environment », *Proc. of the ITS Conference*, Brazil.
- (Breuker et de Velde, 1994) Breuker, J. and de Velde, V. 1994 *The Common KADS Library for Expertise Modelling*, IOS Press, Amsterdam.
- (Chen, Hayashi, Kin, Ikeda, et Mizoguchi, 1998) Chen, W. Hayashi, Y. Kin, L. Ikeda, M. et Mizoguchi, R. 1998. « Ontological Issues in an Intelligent Authoring Tool », in Chan T-W., Collins A. et Lin J. (Eds.), *Global Education On the Net, Proc. of ICCE'98*, vol. 1, 41-50.
- (Collins et Al., 1996) Collins, J.A., Greer, J.E., & Huang, S.H. 1996. « Adaptive assesment using granularity hierarchies and Bayesian nets ». *Proceedings of the Third International Conference: ITS '96*. Frasson, Gauthier & Lesgold (Eds). Springer, pp. 569-577.
- (Devedzic, V. Jerinic, L. et Radovic, D., 1999) Devedzic, V. Jerinic, L. et Radovic, D. 1999. « The GET-BITS Model of Intelligent Tutoring Systems », *International Journal of Continuing Engineering Education and Life-Long Learning*.
- (Gagne 1985) Gagne, R. 1985. *The Conditions of Learning and Theory of Instruction*. New York: Holt, Rinehard, and Winston.

- (Gagne, Briggs et Wager, 1992) Gagne, R. M., Briggs, L. J., et Wager, W. W. 1992. *Principles of instructional design*. 4th ed. Fort Worth, TX: Harcourt Brace Jovanovich.
- (GomezPerez, 1999) Gómez-Pérez A. 1999. *Ontological Engineering: A state of the art*. *Expert Update*, 2(3), 33-43.
- (Goodkovsky et Al., 1994) Goodkovsky, V.A., Kirjutin, E.V. et Bulekov, A.A. 1994. « Shell, tool, and technology for Pop Class ITS production ». In P. Brusilovsky, S. Dikareve, J.Greer et V. Pertrushin (Eds). Proc. of East-West International Conference on Computer Technology in Education. Part 1, pp. 87-92. Crimea, Ukraine.
- (Gros et Spector, 1994) Gros, B. et Spector, J. M. 1994. Evaluating automated instructional design systems: A complex problem. "Educational Technology," p.34(5), p. 37-46. (EJ 483719)
- (Gruber, 1993) Gruber T. 1993. « A Translation Approach to Portable Ontology Specifications ». *Knowledge Acquisition*, 5(2), p. 199-220.
- (Hartley et Sleeman, 1973) Hartley, J. et Sleeman, D. 1973. Towards more intelligent teaching systems. *International Journal of Man-Machine Studies* 2, p. 215-236.
- (Hoffman, 1987) Hoffman, R. 1987. « The Problem of Extracting the Knowledge of Experts From the Perspective of Experimental Psychology » *AI Magazine*, p. 53-67, Summer 1987.
- (Ikeda, Seta et Mizoguchi, 1997) Ikeda, M. Seta, K et Mizoguchi, R. 1997 « Task Ontology Makes It Easier To Use Authoring Tools ». *Proc. of IJCAI-97*, Nagoya, Japan, p. 342-347.
- (Inaba, Ohkubo, Ikeda et Mizoguchi, 2003) Inaba A., Ohkubo R., Ikeda M., et Mizoguchi R. 2003. *Modeling Learner-To-Learner Interaction Process in Collaborative Learning - An Ontological Approach to Interaction*. Supplementary Proceedings of the International Conference on Computer Support for Collaborative Learning (CSCL2003), p. 4-6, Bergen, Norway, June 14-18.
- (Jonassen et Reeves, 1996) Jonassen, D.H. et Reeves, T.C 1996. « Learning with Technology: Using Computers as Cognitive Tools ». In D.H. Jonassen, (Ed.) *Handbook of Research on Educational Communications and Technology*. New York: Scholastic Press, Chapter 25.
- (Koedinger et Anderson, 1995) Koedinger, K., et Anderson, J. 1995. « Intelligent tutoring goes to the big city ». *Proc. of the International Conference on Artificial Intelligence in Education*, Jim Greer (Ed). AACE: Charlottesville, VA, p. 421-428.
- (Merill, 1997) Merrill, M. D. 1997. Learning-oriented instructional development tools. "Performance Improvement," 36(3), p. 51-55. (EJ 544 730)

- (Merrill 1983) Merrill, M.D. 1983. « Component Display Theory ». In *Instructional-design theories and models: An overview of their current status*. C.M. Reigeluth. (Ed). Hillsdale, NJ: Lawrence Erlbaum, p. 279 - 333.
- (Mizoguchi et Bourdeau, 2000) Mizoguchi, R et Bourdeau, J. 2000. « Using Ontological Engineering to Overcome Common AI-ED Problems », *JAIED (2000)*, p. 107-121.
- (Mizoguchi, 1998) Mizoguchi, R. 1998 « A Step towards Ontological Engineering ». <http://www.ei.sanken.osakau.ac.jp/english/step-onteng.html>
- (Mizoguchi, 2003) Mizoguchi R. 2003. *Tutorial on Ontological Engineering. Part 1: Introduction to Ontological Engineering*. New Generation Computing, 21, p. 365-384, Ohmsha and Springer Verlag.
- (Mizoguchi, 2004) Riichiro MIZOGUCHI, Le rôle de l'ingénierie ontologique dans le domaine des EIAH, Entretien réalisé par Jacqueline BOURDEAU, Revue STICEF, Volume 11, 2004, Rubrique, ISSN : 1764-7223, mis en ligne le 30/11/2004, <http://sticef.org>
- (Mizoguchi, Sinitsa et Ikeda, 1996a) Mizoguchi, R. Sinitsa, K. et Ikeda, M. 1996a. «Task ontology design for intelligent educational/training systems », *Workshop on Architectures and methods for Designing Cost effective and Reusable ITSs, ITS96*, Montreal. <http://advlearn.lrdc.pitt.edu/itsarch/papers/mizoguchi.html>.
- (Mizoguchi, Sinitsa et Ikeda, 1996b) Mizoguchi, R. Sinitsa, K. et Ikeda, M. 1996b. « Knowledge engineering of educational systems for authoring systems design », *Proc. Euro AI-ED*, p.329-335.
- (Mizoguchi, Tijerino et Ikeda, 1992) Mizoguchi, R. Tijerino, Y. et Ikeda, M. 1992. « Task ontology and its use in a task analysis interview system -- Two-level mediating representation in MULTIS -- », *Proc. of the JKAW'92*, p.185-198.
- (Mizoguchi, Vanwelkenhuysen et Ikeda, 1995) Mizoguchi R. Vanwelkenhuysen, J. et Ikeda, M. 1995. « Task Ontology for Reuse of Problem Solving Knowledge ». *Building & Knowledge Sharing 1995 (KB&KS'95) (2nd International Conference on Very Large-Scale Knowledge Bases)*, Enschede, The Netherlands, pp.46-59.
- (Mueller, 1998) Mueller, W. 1998. «Effective design of computer-based instructional material », In *Proceedings of 11th International Conference on Technology in Collegiate Mathematics*, New Orleans, United-States.
- (Murray, 1993) Murray, T. 1993. « Formative Qualitative Evaluation for "Exploratory" ITS research ». *J. of Artificial Intelligence in Education*. 4(2/3), pp. 179-207.
- (Murray, 1996) Murray, T. (1996). « Special Purpose Ontologies and the Representation of Pedagogical Knowledge », *ICLS 96*.

- (Murray, 1997) Murray, T. 1997. « Expanding the knowledge acquisition bottleneck for intelligent tutoring systems. *International J. of Artificial Intelligence in Education*. Vol. 8, No. 3-4, pp. 222-232.
- (Murray, 1998) Murray, T. 1998. « Authoring knowledge-based tutors: Tools for content, instructional strategy, student model, and interface design ». *J. of the Learning Sciences*, 7(1) pp. 5-64.
- (Murray, 1999) Murray, T. 1999. « Authoring Intelligent Tutoring Systems: An analysis of the state of the art ». *International Journal of Artificial Intelligence in Education (1999)*, 10, 98-129
- (Murray, Blessing et Ainsworth, 2003) Murray T., Blessing S. et Ainsworth S. 2003, *Authoring Tools for Advanced Technology Learning Environnements: Toward cost-effective adaptive, interactive and intelligent educational software*, Kluwer publishers.
- (Nkambou, Frasson, et Gauthier, 2003) Nkambou, R., Frasson, C., et Gauthier, G. 2003. « Authoring Tool for Knowledge Engineering in ITS ». In *Authoring Tools for Advanced Technology Learning Env.*, Murray T. and al., Eds., 2003, pp. 93-138.
- (Ohlsson, 1987) Ohlsson, S. 1987. « Some Principles of Intelligent Tutoring ». In Lawler & Yazdani (Eds.), *Artificial Intelligence and Education, Volume 1*. Ablex: Norwood, NJ, pp. 203-238.
- (Paquette et al, 1994; Locatis & Park, 1992; Merrill, 1997) Paquette, G., Aubin, C., et Crevier, F. 1994. *An intelligent support system for course design*. "Educational Technology," 34(9), 50-57. (EJ 493 315)
- (Paquette et al. 1995) Paquette, G, Ricciardi-Rigault, C., Bourdeau, J., Paquin, C. et Liégeois, S. 1995. « Modelling the Virtual Campus for Interactive Distance Learning », *Proc. of EDMEDIA'95*. Charlottesville, VA: AACE, 523-528.
- (Paquette et Girard, 1996) Paquette, G. et Girard, J. 1996. « AGD: a course engineering support system », *Proc. of ITS-96*, Lecture Notes in Computer Science, 1086:382-391. Springer-Verlag
- (Paquette, 2002) Paquette, G. (2002). *L'ingénierie pédagogique : pour construire l'apprentissage en réseau*. Québec: Presses de l'Université du Québec.
- (Psyché, 2004) Psyché V. 2004. « Proposition d'une méthode d'ingénierie ontologique pour les EIAH : application aux systèmes auteurs », p. 51-54
- (Psyché et al., 2004) Psyché V., Bourdeau J., Mizoguchi R., Nkambou R. 2004. « Selecting Theories in an Ontology-Based ITS Authoring Environment (2004) ». Dans: J.C. Lester et al. (Eds.): *ITS 2004*, LNCS 3220, pp. 150-161, 2004. *Springer-Verlag Berlin Heidelberg 2004*
- (Psyché, Mendes et Bourdeau, 2003) Psyché V., Mendes O., et Bourdeau J., « Apport de l'ingénierie ontologique aux environnements de formation à distance ». In *STICEF*, vol. 10, Hotte R. and Leroux P., Eds.: STICEF, 2003, pp. 89-126.
- (Psyché, Mizoguchi et Bourdeau, 2003) Psyché, V., Mizoguchi, R. et Bourdeau, J. 2003. « Ontology development at the conceptual level for Theory-Aware ITS Authoring Systems », *Proc. of the AIED Conference*, Sydney.
- (Psyché, Bourdeau, Nkambou et Mizoguchi, 2005) Psyché, V., Bourdeau, J., Nkambou, R. et Mizoguchi, R. 2005. « Making learning design standards work with an ontology of educational theories », *Proc. of the AIED Conference*, Amsterdam.
- (Puerta & Musen, 1992) Puerta, A.R and Musen, M. 1992. « A multiple-method knowledge-acquisition shell for the automatic generation of knowledge-acquisition tools ». *Knowledge Acquisition*, 4, 171-196.

-
- (Redfield, 1997) Redfield, C. 1997. « An ITS authoring tool: Experimental advanced instructional advisor ». In *Papers from the 1997 AAAI fall symposium (Tech. Rep. FS-97-01, 72-78)*, Menlo Park, CA: AAAI.
- (Reigeluth 1983) Reigeluth, C. 1983. The Elaboration Theory of Instruction. In Reigeluth (Ed.), *Instructional Design Theories and Models*. Hillsdale, NJ: Lawrence Erlbaum.
- (Reigeluth 1996) Reigeluth, C. 1996. « What Is the New Paradigm of Instructional Theory ». *ITFORUM*, online. <http://itech1.coe.uga.edu/itforum/paper17/paper17.html>
- (Reigeluth, Buderson et Merrill, 1994) Reigeluth, C. Buderson, C. et Merrill, D. 1994. « Is there a Design Science of Instruction? ». In Merrill, M. D. and Twitchell D.G., ed. *Instructional Design Theory*. Englewood Cliffs, NJ: Ed. Tech. Publ.
- (Shaw et Gaines, 1986) Shaw, M. L. G. et Gaines, B. R. 1986. « Advances in Interactive Knowledge Engineering ». *Submitted to Expert Systems '86*. University of Calgary, Alberta, CANADA: Dept. of Computer Science.
- (Shute et Regian, 1990) Shute, V.J. et Regian, J.W. 1990. « Rose Garden Promises of Intelligent Tutoring Systems: Blossom or Thorn? ». *Presented at Space Operations, Automation and Robotics Conference*, June 1990, Albuquerque, NM.
- (Suppes, 1967) Suppes, P. 1967. « Some theoretical models for mathematics learning ». *Journal of Research and Development in Education*, 1, 5-22.
- (Tennyson et Barron, 1995) Tennyson, R. et Barron, A., Eds. 1995. « Automating Instructional Design: Computer-Based Development and Delivery Tools ». Springer, NATO ASI Series, Series F: Computer and Systems Sciences, vol. 140.
- (Tim Berners, Hendler et Lassila, 2001) Tim Berners-Lee, James Hendler et Ora Lassila, 2001. « The Semantic Web », *Scientific American*, May 2001.
- (Uhr, 1969) Uhr, L. 1969. « Teaching machine programs that generate problems as a function of interaction with students ». *Proceedings of the 24th National Conference*, 125-134.
- (Wenger, 1987) Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.