Table des matières

TABLE DES MATIERES **4**

LISTE DES FIGURES **7**

LISTE DES TABLES **10**

INTRODUCTION GENERALE **11**

INTRODUCTION 11

MOTIVATIONS 13

CONTRIBUTIONS 17

ORGANISATION DE LA THESE 18

CHAPITRE 1 : LE MULTIMEDIA EN E-LEARNING **20**

1.1 INTRODUCTION 20

1.2 E-LEARNING 20

1.3 LE MULTIMEDIA 22

1.3.1 Contenu Pédagogique Multimédia –CPM- *23*

1.3.2 Effets du multimédia dans l'apprentissage *25*

1.4 PROCESSUS D'ELABORATION DU MULTIMEDIA PEDAGOGIQUE 27

1.4.1 Analyse *27*

1.4.2 Conception *27*

1.4.3 Développement *30*

1.4.4 Test *30*

1.4.5 Diffusion *30*

1.5 MODELISATION D'UN DOCUMENT MULTIMEDIA 30

1.5.1 Dimension logique *31*

1.5.2 Dimension spatiale *31*

1.5.3 Dimension temporelle *32*

1.5.4 Dimension hypermédia temporel *33*

1.6 LANGAGES DE SPECIFICATION 35

1.6.1 Le standard SMIL *36*

1.7 CONCLUSION 36

CHAPITRE 2 : APPROCHES DE CONSTRUCTION ET D'INDEXATION DE DOCUMENTS MULTIMEDIAS 38

2.1 INTRODUCTION 38

2.2 APPROCHES DE CONSTRUCTION 38

2.2.1 Système auteur (définition, critères et fonctionnalités) 38

2.2.2 Difficultés et contraintes dans l'édition de documents temporisés 42

2.2.3 Approches d'élaboration des systèmes auteur 43

2.2.4 Enseignement de la phonétique 45

2.2.5 Systèmes existant pour l'enseignement de la phonétique 47

2.2.6 Discussion 51

2.3 APPROCHES D'INDEXATION 51

2.3.1 Indexation classique 52

2.3.2 Vers l'indexation sémantique du contenu multimédia 55

2.3.3 Indexation à base d'ontologies 62

2.3.3.1. Les ontologies OWL 62

2.3.3.2. Utiliser l'ontologie dans les SRI 66

2.3.4 Travaux existants : 67

2.3.5 Discussion 73

2.4 CONCLUSION 75

CHAPITRE 3 : SACOPH, UN SYSTEME AUTEUR POUR L'ENSEIGNEMENT MEDIATIQUE DE LA PHONETIQUE 76

3.1 OBJECTIFS 76

3.2 MODELISATION DE COURS DE PHONETIQUE 76

3.2.1 Dimension structurelle 76

3.2.2 Dimension spatiale 77

3.2.3 Dimension temporelle 77

3.2.4 Dimension hypermédia temporelle 78

3.3 MODELISATION UML DU SYSTEME SACOPH 78

3.3.1 Cas d'utilisation du système SACoPh 79

3.3.2 Architecture du système SACoPh 79

3.3.3 Aspect statique du système : diagramme de classes 84

3.3.4 Aspect dynamique : diagrammes de séquence 84

3.3.5 Interface de SACoPh *86*

3.3.6 Mise en œuvre *86*

3.4 CONCLUSION 89

CHAPITRE 4 : INTEGRATION DES ONTOLOGIES DANS LA CONSTRUCTION ET L'INDEXATION DE CONTENU PEDAGOGIQUE MULTIMEDIA 90

4.1 INTRODUCTION 90

4.2 CONSTRUCTION DES ONTOLOGIES 91

4.2.1 Ontologie de domaine d'enseignement *91*

4.2.2 Ontologie pédagogique de cours vidéo *93*

4.3 PROCESSUS D'ANNOTATION 94

4.4 INDEXATION CONCEPTUELLE 97

4.4.1 Structure de la table d'index *97*

4.4.2 Pondération des concepts dans les segments temporels *98*

4.5 RECHERCHE CONCEPTUELLE 99

4.6 RAISONNEMENT 100

4.7 PROTOTYPE ET EXPERIMENTATION 103

4.7.1 Implémentation *103*

4.7.2 Expérimentation et évaluation *104*

4.8 CONCLUSION 109

CONCLUSION GENERALE ET PERSPECTIVES 110

BILAN DES TRAVAUX ET APPORTS DE LA THESE 110

PERSPECTIVES 111

BIBLIOGRAPHIE 113

LISTE DES PUBLICATIONS PERSONNELLES 120

PUBLICATIONS DANS DES REVUES INTERNATIONALES 120

COMMUNICATIONS DANS DES CONFERENCES INTERNATIONALES 120

COMMUNICATIONS DANS DES CONGRES NATIONAUX (AVEC COMITE DE LECTURE) 121

ANNEXES 122

ANNEXE 1 : **PARTIE DU CODE SOURCE DU SYSTEME SACOPH** 122

ANNEXE 2 : **PARTIE DU CODE SOURCE DU SYSTEME IRSECOV** 123

ANNEXE 3 : **CODE SMIL D'UNE LEÇON GENERE PAR SACOPH** 124

ANNEXE 4 : CODE OWL DES ONTOLOGIES ELABOREES 132

Liste des figures

FIGURE 1. L'ELEARNING (MIELNIKOFF, 2005) 21

FIGURE 2. CYCLE DE PROCESSUS E-LEARNING 22

FIGURE 3. PROCESSUS DE CREATION DE DOCUMENT MULTIMEDIA 27

FIGURE 4. RELATIONS SPATIALES ENTRE OBJETS MEDIA O_I 31

FIGURE 5. SCENARIO TEMPOREL D'UNE PRESENTATION 32

FIGURE 6. EXEMPLE DE LIENS TEMPOREL 34

FIGURE 7. AFFICHAGE DE LA COURBE MELODIQUE AVEC SURLIGNAGE EN ROUGE DES SEGMENTS CORRESPONDANT AUX SYLLABES ACCENTUEES DE L'EXEMPLE « I WONDER IF YOU COULD HELP ME ». 47

FIGURE 8. LE SYSTEME AUTEUR SOUNDS RIGHT. LES FLECHES EXTENSIBLES SONT TRAINEES EN PLACE A PARTIR DE LA COLONNE A GAUCHE ET ENSUITE MODIFIEES POUR OBTENIR UNE TAILLE APPROPRIEE 48

FIGURE 9. INTERFACE GENEREE PAR SWANS 49

FIGURE 10. VUE GLOBALE DE LIMSEE3 AVEC LA PREMIERE ANNOTATION 50

FIGURE 11. INTERFACE DE L'EDITEUR ECOMAS EN MODE EDITION ET EN MODE PRESENTATION 50

FIGURE 12. ARCHITECTURE GENERALE D'UN SRI 52

FIGURE 13. REPRESENTATION VECTORIELLE DE DEUX DOCUMENTS (D1 ET D2) ET D'UNE REQUETE (Q) DANS UN ESPACE COMPOSE DE TROIS TERMES (T1, T2 ET T3). 54

FIGURE 14. LES QUATRE MODES DE RATTACHEMENT DES METADONNEES AU DOCUMENT 58

FIGURE 15. QUELQUES ELEMENTS DE L'ONTOLOGIE DU DOMAINE DE FORMATION (A GAUCHE), ET UN EXTRAIT DE LA SOUS-ONTOLOGIE DU LA FORMATION STATISTIQUE (A DROITE) 68

FIGURE 16. ONTOLOGIE DE L'AUDIOVISUEL : NOUVELLE HIERARCHIE DES TYPES DE SEGMENTS EN MPEG-7. 69

FIGURE 17. EXEMPLE DE DESCRIPTION D'UN SEGMENT VIDEO (ISAAC, 2004) 70

FIGURE 18. PROCESSUS DE SEGMENTATION DES PRESENTATIONS VIDEO (DONG, 2010) 71

FIGURE 19. LA TAXONOMIE DE L'ONTOLOGIE MULTIMEDIA (DONG, 2010) 72

FIGURE 20. INTEGRATION DES ONTOLOGIES DE DOMAINE DANS CELLE DU MULTIMEDIA, (DONG, 2010). 73

FIGURE 21. DIMENSION STRUCTURELLE DE LA LEÇON 77

FIGURE 22. DIMENSION SPATIALE DE LA LEÇON 77

FIGURE 23. DIMENSION TEMPOREL DE LA LEÇON 77

FIGURE 24. SCHEMA MONTRE LE DEROULEMENT DU SCENARIO AVANT ET APRES L'ACTIVATION D'UN LIEN TEMPOREL. 78

FIGURE 25. CAS D'UTILISATION DE L'ENSEIGNANT 79

FIGURE 26. ARCHITECTURE GENERALE DE SACOPH 80

FIGURE 27. PRESENTATION DU COURS GENERE EN SMIL 3.0 AVEC LE LECTEUR AMBULANT 83

FIGURE 28. DIAGRAMME DE CLASSES 84

FIGURE 29. DIAGRAMME DE SEQUENCE DU CAS 'OUVRIR LEÇON' 85

FIGURE 30. DIAGRAMME DE SEQUENCE DU CAS 'GENERER SMIL' 85

FIGURE 31. INTERFACE DE SACOPH 88

FIGURE 32. CAS D'ENREGISTREMENT DE LA PRONONCIATION DE L'EXEMPLE 5 88

FIGURE 33. PARAMETRES DE CONFIGURATION : PERIPHERIQUE AUDIO, LECTEUR DE SMIL ET FONT DE TEXTE GLOBALE. 89

FIGURE 34. ONTOLOGIE DE DOMAINE D'ENSEIGNEMENT. 92

FIGURE 35. ONTOLOGIE PEDAGOGIQUE DE COURS VIDEO. 94

FIGURE 36. INTERFACE PRINCIPALE DE L'OUTIL ONTOCOV. 96

FIGURE 37. DES ASSERTIONS EXPLICITES COMPLETEES PAR DES CONNAISSANCES INFEREES 100

FIGURE 38. ARCHITECTURE GENERALE DU SYSTEME IRSECOV 102

FIGURE 39. STRUCTURE INTERNE DU SYSTEME IRSECOV (DIAGRAMME DE CLASSES) 102

FIGURE 40. INTERFACE DU SYSTEME IRSECOV. 103

FIGURE 41. VERIFCATION DE L'ONTOLOGIE 'DOMAINE D'ENSEIGNEMENT' 106

FIGURE 42. GRAPHE DE L'ONTOLOGIE TOTALE AVEC LE PLUGIN JAMBALYA. 106

FIGURE 43. EDITION DES REGLES SWRL. 107

FIGURE 44. REQUETE EN SPARQL ET SON RESULTAT. 107

FIGURE 45. RESULTATS INFFERES PAR LE MOTEUR D'INFERENCE JESS.....108

FIGURE 46. DOMAINES FONCTIONNELS DE SMIL 2.0.....126

Liste des tables

TABLE 1. ALLOCATION DES MEDIA 29

TABLE 2. CLASSIFICATION DES SYSTEMES AUTEURS FONDATEURS 39

TABLE 3. STRUCTURE DE LA TABLE D'INDEX 97

TABLE 4. LISTE DES CONCEPTS AVEC LES SEGMENTS ASSOCIES. 104

Introduction générale

Introduction

Depuis quelques années les capacités multimédias des ordinateurs ont permis de faire d'énormes progrès tant au niveau matériel (carte son, vidéo, etc.) qu'au niveau logiciel (différents algorithmes de codage, de compression, etc.). De plus, les ordinateurs peuvent être facilement interconnectés grâce à l'évolution du réseau mondial Internet. Cela donne lieu de produire des logiciels éducatifs plus démonstratifs, plus vivants et pouvant être utilisés à distance.

Il y a eu un accroissement considérable des besoins de matériel didactique multimédia dans l'e-Learning, car le contenu de ce type de matériel a suscité récemment beaucoup d'intérêts pour attirer l'attention de l'apprenant et aussi pour augmenter sa compréhension. D'importants résultats venants de recherches en psychologie prouvent que l'enseignement par les multimédia surmonte énormément les obstacles actuels de l'apprentissage en réduisant les coûts en termes de temps de production et de diffusion du contenu éducatif. Une étude menée dans ce domaine a enregistré un gain de 71% en terme de temps d'apprentissage avec le multimédia par rapport à l'enseignement en classe ; ceci encourage le développement de la suite multimédia à base de logiciels éducatifs. La raison derrière ces statistiques, c'est que d'un côté, l'enseignement basé sur le multimédia force le concepteur du didacticiel à mieux organiser le matériel d'apprentissage (contenu pédagogique multimédia) par rapport au discours prononcé dans une classe traditionnelle. D'un autre côté, il permet aux apprenants de maîtriser le rythme d'apprentissage et d'interagir avec le contenu pédagogique multimédia.

La performance d'apprentissage, une meilleure productivité, la taille croissante et la diversité de la communauté éducative on line sont également liés au multimédia redondant par rapport aux caractéristiques du mono-média du matériel d'apprentissage informatisé. Les médias doubles ou multiples, dont le contenu est étroitement lié et qui

sont combinés dans un canal d'apprentissage, ont le potentiel de fournir un apprentissage efficace lorsque les médias s'étayent clairement entre eux et sont présentés aux apprenants ayant de faibles connaissances préalables dans le domaine en apprentissage (Najjar, 1996).

Prenons l'exemple d'un document pédagogique dont le contenu est une suite de diapositives. Ces dernières portent généralement sur des concepts ou des idées très courtes en termes d'expression (pas de détails). Dans ce cas, l'apprenant a besoin de plus d'explications pour mieux comprendre. La synchronisation de la voix de l'enseignant, enregistrée sous format numérique audio, avec chaque diapositive rend plus claire la compréhension de ce contenu pédagogique.

Un autre exemple (Atif, 2003) d'un matériel pédagogique sous forme d'un document multimédia temporisé qui intègre en synchronisant simultanément un script de la vidéo (montrant le visage parlant de l'enseignant ou de tout autre matériel vidéo lié à l'objet du cours) avec une transcription textuelle. Ceci est fait pour aider les apprenants, qui ont une déficience auditive ou des difficultés à comprendre l'accent de l'enseignant, à lire le discours de ce dernier.

Les documents multimédia dont on parle dans ce cas, sont abordés non seulement sous l'angle de leur structure logique, spatiale et navigationnelle mais aussi temporelle. La structure temporelle décrit l'enchaînement des éléments médias dans le temps. Prenons le premier exemple ; cette structure permet de présenter chaque diapositive simultanément avec son explication auditive et les diapositives (de type image) se succèdent dans le temps séquentiellement. Ce type de support construisent un contenu pédagogique multimédia sous forme une séquences vidéos.

Le traitement de la dimension temporelle dans les documents, ainsi que leurs éléments de base qui ont eux-mêmes une telle dimension (audio, vidéo) constitue l'objet de cette thèse. L'introduction de cette nouvelle dimension génère nécessairement des besoins en termes de langage d'expression. Des standards ont été développés répondant à ce besoin, parmi eux, citons le format SMIL. Le consortium W3C a étudié ce nouveau format qui est l'acronyme de « Synchronized Multimedia Integration Language » tout en permettant un échange et une évolution interopérable du Web. La plus grande partie de documents ayant ce format sont destinés à l'e-learning.

MCours.com
www.mcours.com
Cours, exercices & projets de fin d'études

Le développement d'un contenu pédagogique multimédia sous forme d'une séquence vidéo est une étape importante du processus de conception de l'enseignement. Construire un matériel ou une ressource pédagogique multimédia temporisée avec ces nouvelles technologies et qui soient de qualité ne se résume pas en la mise en ligne d'un polycopié en y ajoutant quelques animations. Cela nécessite un scénario très charpenté et une présentation structurée. Une fois que l'information est accessible sur Internet, on n'obtient pas nécessairement un système d'apprentissage si on n'a pas réfléchit à la manière dont les gens vont prendre l'information et ce qu'ils vont en faire. Il est important de représenter clairement les informations et les connaissances sur le plan structurel et temporel et de se demander quelle stratégie ou quel moyen de diffusion doit-on utiliser.

L'importance de la qualité des interactions dans une séquence vidéo pédagogique, impose des scénarios de communication riches et actifs pour l'apprenant. Cette exigence de qualité impose parfois aux enseignants une complète refonte de leurs cours et de leur manière de les concevoir, d'où l'intérêt de proposer aux enseignants des systèmes auteurs qui soient adaptés à leurs besoins et qui leurs permettront d'imprégner leur pédagogie dans des contenus multimédias interactifs et standards, répondants ainsi aux besoins des apprenants en termes d'assimilation, de compréhension et de mémorisation. C'est dans ce contexte que se situe la première partie de notre travail.

La gestion des séquences vidéo pédagogiques suppose la structuration et la mise en place d'une banque de données. L'accès automatique et rapide aux segments temporels pertinents de cette banque est une tâche fondamentale mais complexe qui passe nécessairement par une étape d'indexation des séquences vidéo. Ceci représente la deuxième partie de notre travail.

Motivations

1- Le premier problème auquel nous nous intéressons correspond à la médiatisation de cours (et en particulier) de la phonétique d'une langue (et plus précisément l'anglais) en ligne. Les langues anglaise et française partagent un important lexique et beaucoup de formes orthographiques d'un même mot sont proches dans les deux langues. Pourtant, les systèmes accentuels mis en place par les deux langues pour ces mots les rendent

opaques à l'oral pour les apprenants. On observe que certaines syllabes sont plus aisément audibles que d'autres. On parle dans ces cas-là de syllabes accentuées.

L'apprenant francophone se trouve ainsi confronté à deux difficultés : percevoir, lors de la phase d'écoute, les syllabes accentuées et les syllabes non accentuées, et reproduire, lors de la phase de production, un contraste suffisant entre les deux types de syllabes. Les étudiants français, lors de la réalisation d'exposés oraux peuvent, malgré une langue assez correcte sur le plan lexical et syntaxique, faire preuves de sérieuses lacunes tant au niveau phonétique que dans celui prosodique. L'absence de la discrimination voyelles/diphtongue et le déplacement de l'accent tonique rendent certains mots méconnaissables.

Des études empiriques contrôlées confirment ce que les enseignants observent de façon quotidienne. Ces études montrent que des canadiens anglophones reconnaissaient moins bien des mots isolés prononcés par un canadien francophone que par un canadien anglophone. Les auteurs attribuent cette différence à un défaut d'accentuation. Mais dans (Stenton, 2005), l'auteur confirme que ce problème n'est pas sensoriel, mais se situe plutôt au niveau de la mémoire de travail. Il s'agit d'une négligence lors de l'encodage de l'information. Les étudiants français qui apprennent l'anglais négligent de traiter l'accent tonique, car il a peu de valeur dans leur langue maternelle, et par conséquent ils ne stockent pas cette information. Lors de leurs exposés, ils placeront donc l'accent tonique de façon aléatoire sur l'une des syllabes du mot anglais, indiquant ainsi non pas une surdité ou un problème de production mais une négligence au niveau de l'encodage et une absence de stockage de la place de l'accent tonique. Ce qui a un effet négatif sur la compréhension de leur discours par des interlocuteurs anglophones.

Beck et .al ont constaté que certains étudiants après 10 ans d'anglais, en maitrise langues étrangères appliquées, ne maitrisent pas encore la prononciation de mots qui paraissent élémentaires (comme : who, women, chocolate, village, low, allow, sun, son). Les séries telles que (there're, aren're, weren't, were, where) ou les mots graphiquement proches tel que (tough, trough, though, through, thought) posent d'énormes problèmes de mémorisation à l'oral (Beck, 2005).

(Beck, 2005) a fait l'hypothèse que le processus attentionnel joue un rôle fondamental dans ce cas. Les solutions visuelles informatiques semblent constituer une bonne solution. Le repérage visuel aide à la discrimination des parties du discours où des problèmes de perception et de compréhension se posent. Le son prend alors une forme visuelle observable dans le temps à l'inverse de sa matérialité première constituée par d'éphémère vibration de l'air. La prononciation sera plus facile si l'étudiant lit et entend simultanément le mot 'de*velo*pment', où la syllabe accentuée est mise en exergue visuellement (lui associer un style, un font et une couleur qui soient différents du reste du texte). Traiter un tel mot consiste en **un encodage auditif de l'information linguistique et un encodage visuel des deux informations linguistique et paralinguistique**, d'où la nécessité d'outil de création de documents supportant une tel présentation.

2- Le deuxième problème à surmonter concerne l'accès rapide et automatique aux segments temporels pertinents d'une banque de séquences vidéo multimédia pédagogiques. Le e-Learning repose largement sur les documents multimédia et en particulier sur les documents vidéo. Beaucoup d'instituts, d'écoles et d'associations diffusent sur le web des enregistrements vidéo sur des exposés scientifiques lors des conférences, séminaires ou soutenances de thèses ou d'habilitation (exemple INRIA, ENS, Aristote). Des universités (ou campus virtuels) diffusent sur le Net des cours magistraux dispensés dans leurs locaux sous forme audio ou vidéo et on cite comme exemple les universités : MIT[1], Berkeley[2], Strasbourg[3], MedNet[4], Lausanne[5]. Des exposés magistraux universitaires sont regroupés sur des portails thématiques comme le WebTV pédagogique de Lyon3 ou SciVee (fondé par P.E. Bourne de UC San Diego et L.M. Chalupa de UC Davis et qui est un des nombreux exemples de sites pour vidéos

[1] MIT : http://ocw.mit.edu/courses/audio-video-courses/

[2] Berkeley : http://webcast.berkeley.edu/

[3] Strasbourg : http://audiovideocours.u-strasbg.fr/avc/home

[4] MedNet : http://www.consorzionettuno.it/mednetu/e/skins/mednetu/home/index_1024_fr.asp

[5] Lausanne : http://itunes.unil.ch/

scientifiques). Ces vidéos sont enregistrées sous différents formats : à savoir vidéo streamé (ou podcast), ou documents multimédia structurés (où la vidéo et la voix du présentateur sont synchronisées avec des diapositives) pour une diffusion en direct ou en différé.

Alors que ces documents vidéo sont de plus en plus accessibles, vue leur richesse et leur expressivité sémantique, et que leur nombre est de plus en plus croissant, leur traitement pose toujours problème. En particulier, la recherche des séquences vidéo pertinentes en suivant des critères liés au contenu sémantique n'est pas triviale. Ceci peut nuire à l'apprenant pendant sa révision ou au chercheur (ou enseignant) qui veut réutiliser une partie d'une vidéo pour son propre compte. Il est souvent plus pratique pour un utilisateur (apprenant ou enseignant) d'utiliser une information sémantique dans sa requête (concepts scientifiques) pour obtenir les réponses les plus pertinentes. Par conséquent un processus d'indexation et de recherche par la sémantique de ce type de vidéo doit être mis en place.

Avant d'atteindre ce stade, on doit noter qu'il est pratiquement impossible d'atteindre le niveau sémantique en partant d'une analyse bas niveau du contenu vidéo. Les interprétations du contenu d'un document vidéo, qui sont sémantiquement plus riche, rend la tâche de l'indexeur plus compliquée que lorsqu'il s'agit d'une simple indexation par mots clés. Ceci est dû au fait qu'il doit choisir les meilleurs index pour décrire un contenu très riche en informations. On rencontre la même difficulté lors du processus de recherche. Donc il faut d'abord mettre en place des modèles capables de décrire et de modéliser le contenu sémantique de ces vidéos afin de faciliter l'accès, la réutilisation et la navigation par la sémantique.

Dans cette optique, le traitement du contenu vidéo à l'aide de techniques à base de connaissances constitue une piste intéressante. Dans la perspective du web sémantique, qui est en voie de devenir une assise pour les environnements de formation à distance, les ontologies, mieux que toute autre méthode de représentation des connaissances, offrent de façon spécifique une sémantique riche (Psyché, 2003). Dans une plateforme d'enseignement, la précision d'une recherche d'un contenu pédagogique peut être améliorée si elle repose sur le vocabulaire conceptuel définit dans une ontologie et ceci

tout en évitant les ambiguïtés au niveau terminologique et en autorisant des inférences qui diminuent le bruit et augmentent la pertinence.

Contributions

Les objectifs de nos travaux sont de répondre aux différents éléments de problématique dégagés plus haut. Nous proposons dans ce travail un modèle d'un contenu pédagogique multimédia et deux systèmes qui gèrent la création des documents de ce modèle et leurs indexations à base d'ontologie, tout en répondant aux besoins identifiés des deux acteurs principaux, à savoir enseignant et apprenant. De façon plus détaillé, on doit :

- Améliorer la perception et la mémorisation des concepts de phonétique (tel que l'accent tonique) en utilisant l'approche de double codage (visuel et auditif). Pour ce faire, on doit concevoir un modèle de document multimédia temporisé qui concrétise cette approche.

- Proposer aux enseignants de langue un système auteur permettant la préparation des cours multimédia publiables sur le web (standard SMIL3.0), selon le modèle précité, qui s'adapte à l'enseignement des concepts de la phonétique. Ce système doit avoir une souplesse d'édition via une interface qui soit la plus conviviale possible pour que l'on s'approche le plus possible du principe de WYSIWIG (What You See is What You Get).

- Concevoir des modèles ontologiques afin de former un vocabulaire conceptuel partageable entre la communauté des enseignants et des apprenants.

- Utiliser ce vocabulaire dans l'annotation des séquences vidéo de type cours magistraux universitaires.

- Développer un système d'indexation et de recherche par le contenu sémantique des segments vidéo, à base de leurs annotations ontologiques, pour pallier au manque observé de tels outils actuellement.

Organisation de la thèse

Afin d'aboutir aux objectifs fixés, la suite de ce document est organisée de la façon suivante :

- Le premier chapitre s'intéresse à la discussion autour des multimédias, afin de dégager leurs caractéristiques en tant que contenu pédagogique et leurs apports bénéfiques en e-learning. Ce chapitre présente le processus d'élaboration de ce type de contenu en précisant la position de la phase de médiatisation du contenu dans ce processus. Une étude des différentes dimensions d'un document multimédia est présentée tout en prenant en considération la structure temporelle. Les différents langages permettant d'exprimer cette nouvelle structure sont discutés, en mettant l'accent sur le standard SMIL qui représente le langage pivot des documents multimédia dans cette thèse.

- Le deuxième chapitre traite deux axes différents : à savoir les approches de création de documents multimédia pédagogique et les approches de leurs indexations par la sémantique. De ce fait, le chapitre présente dans sa première partie les critères et les fonctionnalités qui dépendent des outils de création appelés aussi systèmes auteurs. Suit une étude synthétique et critique des systèmes auteurs existants qui aident à la création des présentations multimédia ayant le format SMIL, et surtout ceux dédiés à l'enseignement de la phonétique. La deuxième partie s'intéresse à l'indexation par la sémantique de ce type de documents. Donc, cette partie illustre les différents composants d'un système de recherche d'information et explique le principe d'une indexation sémantique et comment elle est appliquée aux documents multimédia temporisés (séquence vidéo). Ensuite, elle montre une étude analytique et critique de quelque systèmes de recherche par la sémantique, qui utilisent les ontologies dans des corpus de type vidéo et en particulier ceux dédiés à l'e-learning.

- Le chapitre trois est réservé à la présentation de notre première contribution. Pour cela, il illustre, en premier lieu, le modèle de cours multimédia temporisé qui concrétise l'approche de double codage. En deuxième lieu, il présente l'architecture de notre système auteur appelé SACoPh qui est le sigle de (Système

Auteur de Cours de Phonétique), qui génère des cours de phonétique publiables via le web, selon le modèle conçu, tout en exploitant les nouvelles fonctionnalités de la version 3.0 du standard SMIL. Ce système est destiné aux enseignants de langue qui ont peu de connaissances des outils informatiques et c'est pour cela que son interface est conçue de telle façon qu'elle soit la plus simple, la plus ergonomique et la plus conviviale possible.

- Le dernier chapitre est consacré à notre deuxième contribution. Au niveau théorique, cette contribution consiste, en premier, à la proposition et la construction de deux types d'ontologies : une pour la structuration pédagogique d'un cours vidéo et l'autre pour la description du contenu sémantique de ses différents granules. Ces deux ontologies seront utilisées dans la phase d'annotation conceptuelle. L'annotation conceptuelle d'un corpus des cours vidéo, basé sur ces deux ontologies, est élaborée en utilisant le système OntoCov (Ontologie des Cours Vidéo), que nous avons développé à cette fin. Ensuite notre contribution consiste à l'implémentation du prototype IRSeCoV : un système d'Indexation et de Recherche Sémantique des Cours Vidéo pédagogiques via les annotations conceptuelles associées au corpus en question. Le présent chapitre se termine par une expérimentation faite dans le but d'évaluer notre approche.

Nous clôturons cette thèse par une conclusion et quelques perspectives.

Chapitre 1 : Le Multimédia en e-Learning

1.1 Introduction

Les progrès techniques récents dans le multimédia ont rendu l'Internet de plus en plus interactif et dynamique. Le secteur de l'enseignement et de la formation ne pourrait ignorer cette mondialisation de l'information et il devra en profiter et assister l'apprentissage structuré et traditionnel par des applications apportant une plus-value à l'apprenant et une meilleure qualité à l'enseignement.

Par conséquent, nous avons dédié l'ensemble de ce chapitre à la discussion autour des multimédias, afin de dégager leurs caractéristiques en tant que contenu pédagogique et leurs apports bénéfiques en e-learning. Nous présentons par la suite le processus d'élaboration du matériel didactique ayant le caractère multimédia en précisant la position de la phase de médiatisation des cours dans ce processus. Nous étudions les différentes dimensions d'un document multimédia tout en prenant en considération la structure ou dimension temporelle. De même, nous exploitons les différents langages permettant d'exprimer cette nouvelle structure. Enfin nous présentons les principes du standard SMIL qui représente le langage pivot des documents multimédia à traiter dans cette thèse. Dans la section qui suit, nous définissons le concept du e-learning pour mieux déterminer sa relation avec le multimédia.

1.2 E-Learning

L'E-learning est le terme utilisé par les anglo-saxons pour désigner les formations à distance accessibles via une connexion internet. Michel (Mielnikoff, 2005) a précisé que le principe du e-learning était de pouvoir accéder à ses cours depuis un poste distant (chez soi, depuis son entreprise). Ainsi les lieux habituels de suivi d'une formation (établissements, classes, bibliothèques) n'existent plus physiquement et ils sont substitués par le Système de Gestion des Cours ou **LMS** (Learning Management

System). C'est le LMS qui fait le lien entre les apprenants, les cursus, les tuteurs, les ressources et les contenus présents dans le système. L'apprenant, via cette plate-forme, se verra attribuer un certain nombre de modules de cours, d'exercices, d'évaluations qu'il devra effectuer en tenant compte d'une planification établie. Le tuteur (ou formateur) se charge de gérer les apprenants qu'il doit suivre. Ainsi il pourra leurs affecter des ressources à consulter, des cours à étudier ou des évaluations afin de se rendre compte de la bonne assimilation des contenus proposés.

La communication entre tous ces acteurs se fait via Internet. Il y a quelques années encore, il était très difficile de concilier des contenus de qualité avec un confort d'utilisation maximum. En effet, les contraintes liées au bas débit faisaient que les seuls contenus réellement exploitables étaient constitués de pages au format HTML ou de documents textes.

Le haut débit, accessible aujourd'hui à une très large population et à un prix faible, fait que le principe du e-learning est amené plus que jamais à se développer de manière exponentielle. Ainsi un point négatif majeur que l'on pouvait noter, lié aux débits, n'est donc plus d'actualité. Aujourd'hui, tous les formats de contenu peuvent être utilisés, même les plus lourds, tels que les formats audio et vidéo.

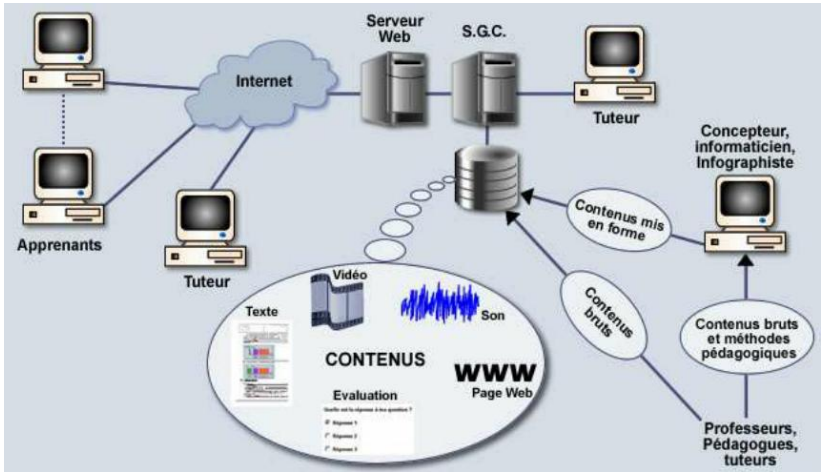


Figure 1. *L'e-learning (Mielnikoff, 2005)*

Aujourd'hui, de nombreux exemples d'e-learning utilisent l'Internet pour fournir une formation sur demande supportant un contenu interactif et riche en médias.

Le cycle du e-learning peut être décrit à travers cinq phases capitales (figure 2) : phase de création, phase d'orientation, phase de formation, phase de suivi et d'évaluation et phase de gestion.

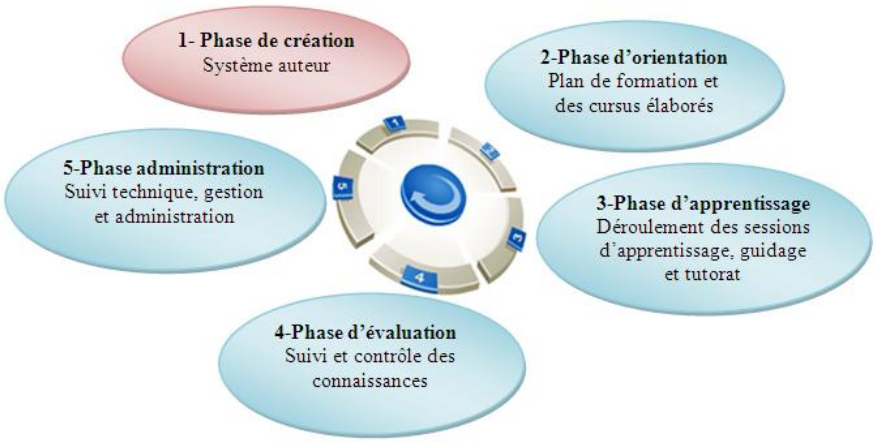


Figure 2. *Cycle de processus e-learning*

Dans la littérature, des études comparatives établies sur plusieurs plates-formes (LMS), en prenant pour guide le cycle du processus du e-learning (figure 2), mènent à constater qu'aucune plate-forme ne couvre complètement ce cycle. Chacune d'elles apporte les fonctionnalités correspondantes aux besoins pour lesquels elle a été conçue. Les deux phases les mieux couvertes sont la phase d'apprentissage et la phase d'administration. La phase création n'est couverte que par les systèmes qui offrent des outils auteurs. Ceci nous amène à nous intéresser d'avantage à cette phase, dont nous allons examiner les outils offerts dans le chapitre suivant. Puisque en premier lieu, nous nous intéressons beaucoup plus à l'élaboration et à la création du contenu multimédia, nous devons mettre en lumière le concept multimédia.

1.3 Le Multimédia

Le terme **Multimédia** peut être décrit comme une combinaison de textes, d'images, de sons, de commentaires, d'animations et de séquences vidéo à l'intérieur d'une application sur ordinateur.

Document multimédia : c'est donc un document comportant plusieurs médias où les textes, les images, les animations, le son et la vidéo sont des exemples classiques. (Chisogne, 1999), a mentionné qu'il en existe d'autres, plus exotiques, comme les pages

web ou les applets. Ce sont, avec les interactions utilisateurs, les briques de base qui vont servir à élaborer un véritable document interactif : un document multimédia.

Une définition plus rigoureuse donné par Layaïda (Layaïda, 1997) précise qu'un système ou une application est qualifiée de **multimédia** s'il (ou elle) supporte le traitement intégré de plusieurs médias dont au moins un est de nature temporisée.

Une classification des différents types de médias selon leurs natures, appelés aussi objets multimédias, a été fournie par Cécile (Cécile, 1999). Elle propose de classer ces objets en deux catégories de par leur mode de présentation dans le temps :

- Les objets discrets : leur contenu est délivré de façon instantanée, comme le texte et les images statiques.
- Les objets continus : leur contenu est délivré de façon progressive comme les vidéos, les sons ou les animations.

Comme ils peuvent aussi être classifiés par leur mode de perception :

- Les objets visibles, c'est-à-dire que l'on peut afficher : texte, image, vidéo ou animation.
- Les objets audibles comme les sons.

Pour chaque catégorie d'objets, différents formats de codage existent (Ascii pour les textes, jpeg, GIF, mp3, Tiff, png, pour les images, au, wav, ra, rm pour les audio, avi, mpeg, rv, rm, ... pour les vidéos, etc.).

Puisque on s'intéresse à l'intégration du multimédia dans le e-learning, il est important avant toute chose de définir ce qu'est « *un contenu pédagogique multimédia interactif* » pour pouvoir cadrer au mieux notre travail et en particulier ce que les systèmes, que nous allons développer, visent à produire.

1.3.1 Contenu Pédagogique Multimédia –CPM-

Afin de faire le point sur les divers sens que peut couvrir ce mot, nous allons présenter une analyse faite par (Bousbia, 2005) de chacun de ses composants, à savoir contenu, pédagogie, multimédia et interactivité, afin d'en dégager les principales caractéristiques et d'en souligner leurs impacts.

- **Contenu** : Le choix du mot contenu permet d'éviter des confonds possibles entre les différents sens du mot « cours ». Il s'agit ici de développer un support

d'informations, des idées, de connaissances ou d'un savoir d'une manière structuré selon un contexte déterminé et en rapport avec un domaine spécifique.

- **Pédagogie** : Le contenu à produire sera destiné à une formation, ce qui le différencie d'un document d'information. De ce fait, il se rédige selon une démarche pédagogique précise, dans le but de faire acquérir un savoir ou de le faire construire. Cela dépend de l'objectif pédagogique, le domaine à enseigner, la population ciblée (niveau d'étude et de difficulté), ainsi que des compétences de l'enseignant. Ce dernier doit donc organiser le contenu sous forme de programmes et créer une technique qui permet à l'apprenant de parcourir le contenu selon un certain ordre et à son rythme propre afin de s'approprier ce contenu. Disons enfin que ce contenu peut prendre plusieurs formes : un cours, une partie de cours (un chapitre, un module), une évaluation, des travaux dirigés, des travaux pratiques, …etc. Dans notre travail, nous nous intéressons à l'étude de la pédagogie de développement d'un contenu ayant la forme cours.

- **Multimédia** : Un contenu pédagogique est clairement de type multimédia puisqu'il est composé de différents types de médias (texte, image, etc.). La question qui nous concerne est comment arranger et agencer ces médias en vue d'un meilleur apprentissage. Une image ou un schéma est parfois plus significatif qu'une description textuelle. Il faut donc prendre des décisions sur le choix d'un média par rapport à un autre. La structuration de ces médias dans une interface harmonieuse (conception et présentation de l'interface homme-machine) sera une des objectifs de notre travail, ainsi que sur la manipulation de ces médias pour l'acquisition de connaissances en tenant compte des contraintes et des moyens disponibles.

- **Interactivité** : Dans un contexte de formation à distance, l'apprenant construit son propre apprentissage au fur et à mesure qu'il avance dans son programme. Le contenu pédagogique doit donc gérer l'apprentissage par une technique d'interaction efficace permettant à l'apprenant d'interagir avec le contenu et de gérer ses actions et ses retours en entrainant des réponses aux comportements.

A ce stade, Bousbia (Bousbia, 2005) a résumé qu'un contenu pédagogique multimédia interactif joue le rôle de savoir structuré par l'enseignant sous forme d'un ensemble d'unités de connaissances liées entre elles, matérialisées en médias, dans le but de présenter l'information (connaissances déclaratives) et de fournir un espace d'exploration (activités pédagogiques) ou d'échange (travail collaboratif).

Pour proposer des cours avec un CPM à distance de qualité, on ne saurait se satisfaire de "balancer" aux étudiants le cours brut de l'enseignant sous forme d'un fichier doc ou

.pdf. En effet, un tel cours, destiné à être transmis en présentiel, contient des contenus qui sont censés être explicités aux étudiants en présence. Mais dans une formation à distance, le cours brut manquera de clarté. Il faut donc **recourir à une stratégie de médiatisation des connaissances** du cours en question. Celle-ci implique une démarche rigoureuse et se fixe les objectifs suivants :

- respecter la philosophie de l'enseignement et la démarche d'apprentissage voulu par l'enseignant,
- faciliter l'acquisition des connaissances à distance par l'apprenant,
- exploiter les atouts d'une communication multimédia,
- prendre en compte les conditions difficiles de lecture à l'écran et adapter la typographie et la charte graphique,
- proposer un design web attirant adapté à la nature des contenus transmis et au public, tout en apportant une identité propre.

1.3.2 Effets du multimédia dans l'apprentissage

Du côté de l'apprenant, on fait rappel aux concepts clés de la « théorie cognitive d'apprentissage multimédia », développée par Richard Mayer (Mayer, 2001). Ce dernier fonde sa théorie sur trois hypothèses : premièrement les humains assimilent l'information selon deux canaux, l'un visuel et l'autre verbal ; deuxièmement chacun de ces canaux contient une capacité limitée d'assimilation ; et troisièmement un apprentissage actif admet la construction dynamique d'une série de processus cognitifs. Mayer essaye de déterminer dans quelles situations les apprenants peuvent profiter des matériels didactiques multimédia pour améliorer leur assimilation et leur compréhension. Dans ce sens, il propose sept principes pour la conception d'une présentation multimédia de matériels didactiques :

- *multimédia* : les étudiants apprennent mieux à partir d'images et de mots mis ensembles que de mots tous seuls ;
- *rapprochement spatial* : les étudiants apprennent mieux lorsque des mots et les images correspondantes sont présentés le plus près possible les uns des autres ;
- *rapprochement temporel* : les étudiants apprennent mieux lorsque l'audio (qui représente la voix) et le texte correspondant sont présentés de manière simultanée, plutôt que de manière successive ;
- *cohérence* : les étudiants apprennent mieux lorsque des mots, des images ou des sons qui ne sont pas nécessaires à la présentation sont exclus ;

- *modalité* : les étudiants apprennent mieux à partir d'animations et de narrations auditives qu'à partir de texte à l'écran ;
- *redondance* : les étudiants apprennent mieux à partir d'animations et de narrations auditives réunies qu'à partir d'animations, de narrations auditives et du texte à l'écran (l'ajout du texte est perçu comme de la surcharge cognitive) ;
- *différence individuelles* : le design affecte plus profondément les apprenants qui ont un bas niveau de connaissances que ceux qui ont un haut niveau ; et ceux qui préfèrent un environnement plus spatial que ceux qui préfèrent un environnement moins spatial.

Bien que dans la pratique, plusieurs de ces principes peuvent varier (et que bien d'autres peuvent s'ajouter) selon les contextes d'application, il est important de noter qu'il apparaît vraiment qu'il a mis une articulation entre les divers médias et les manières dont ils sont présentés. Si l'on accepte que cette articulation se crée conformément aux canaux de perception de l'individu, il est nécessaire d'approfondir la question des passages d'un canal à l'autre dans la pensée de l'individu où ils demeureront au niveau de la mémoire à long terme (connaissances).

Pour Mayer, lorsqu'un étudiant lit une image, le canal reste visuel tout au long d'un processus qui va de la perception à l'apprentissage. Au contraire, lorsqu'il lit un texte, il se produit dans le processus un passage du canal visuel au canal verbal.

Enfin, deux autres principes sont proposés par Mayer comme axes de recherche. Le premier est en rapport avec l'interactivité. Il semble que les étudiants apprennent mieux lorsqu'ils contrôlent la présentation et l'affichage. Le deuxième est en relation avec la personnalisation : il apparaît que les étudiants apprennent mieux avec un guidage personnalisé de la présentation.

On peut synthétiser que dans un tel contexte éducatif les médias peuvent jouer un rôle très important dans l'apprentissage et l'enseignement. L'Apprentissage est affecté négativement lorsque les médias ne sont pas intelligemment intégrés dans un contenu pédagogique. A l'inverse, l'apprentissage est meilleur (et optimisé) lorsque les médias ont été soigneusement sélectionnés et appliqués avec des stratégies pédagogiques pour servir les besoins spécifiques d'un apprenant dans différents domaines d'apprentissage.

Cependant, ce support de formation est le résultat d'un processus de création et d'une suite d'actions bien définies. Dans la section suivante, on essaye de mettre en lumière les différentes étapes de ce processus.

1.4 Processus d'élaboration du multimédia pédagogique

D'après les conditions que doit satisfaire un contenu pédagogique multimédia, cité ci-dessus, il est évident que sa préparation n'est pas une simple besogne mais toute une procédure à suivre. Le processus d'élaboration est composé de cinq grandes parties: analyse, conception, développement, test et diffusion. Comme toute démarche de création, le travail le plus important réside au niveau de la conception. (Bousbia, 2005) Bousbia a essayé de donner une définition aux différentes étapes comme suit :

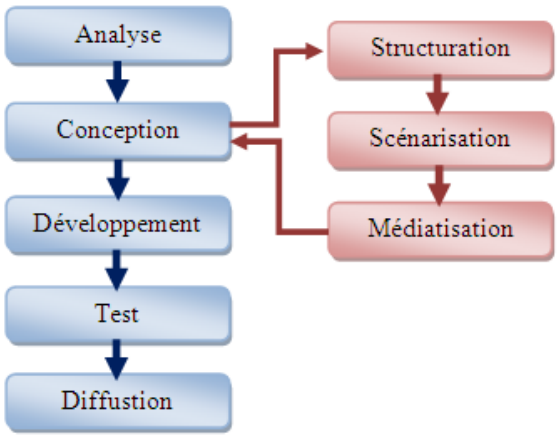


Figure 3. *Processus de création de document multimédia*

1.4.1 Analyse

Comme pour tout produit, elle consiste à la préparation d'un cahier des charges par la détermination des objectifs (contexte, domaine concerné, objectif pédagogique), des populations (public ciblé), des contraintes (niveau requis) et des ressources (éléments du contenu), et ce, en fonction des objectifs finaux de la formation et des besoins des apprenants. Elle permet aussi de synchroniser les demandes des pédagogues avec les possibilités informatiques.

1.4.2 Conception

Dans un enseignement classique, la réalisation pédagogique de contenu, conduit l'enseignant à faire des choix, à prévenir des contenus, à prévoir le déroulement de ses

actions, à organiser des situations d'apprentissage et à déterminer les temps impartis aux différentes tâches et actions. Dans une formation à distance, elle concerne la structuration, la scénarisation et la médiatisation du contenu.

1.4.2.1 Structuration

Le contenu pédagogique d'un cours contient un sous-ensemble des connaissances du domaine dont il fait partie. Sa structuration revient donc au découpage (hiérarchique) de ces connaissances en unités élémentaires de granularité très fine. Ceci est fait dans le but de : faciliter la recherche (l'indexation), filtrer (l'adaptation du contenu) et construire ou assembler le contenu (la réutilisation).

1.4.2.2 Scénarisation

Bousbia a suggéré dans son article (Bousbia, 2006) que la scénarisation est bien de donner un sens à la structure hiérarchique du contenu, par la détermination d'un parcours pédagogique et d'un scénario didactique. Ceci est fait par la détermination de l'ordonnancement des concepts que doit aborder l'apprenant dans son processus d'apprentissage.

En effet, définir un parcours pédagogique consiste à organiser et décrire les transitions entre les différentes notions à appréhender, alors que le scénario didactique concerne la planification des différentes séquences relatives à chaque notion. Le scénario didactique représente une manière d'enseigner la notion (exposé, illustration, simulation, évaluation, question/réponse, etc.). Cela revient à organiser l'évolution dans le temps du statut des connaissances relatives à la notion à enseigner en une suite d'actes pédagogiques (instructions, évènements ou actions) exécutés séquentiellement sur les médias utilisés (lire un texte, rédiger une synthèse).

Selon (Reyes, 2006), la scénarisation essaie de répondre à la problématique de la mise en scène de plusieurs médias dans un espace interactif et numérisé à partir de deux aspects qui lui sont essentiels : le scénario (en anglais *script*) et le scénarimage (en anglais *story-board*). Dans le cinéma et la télévision, le scénario expose la suite d'actions qui seront filmées. Il s'agit de planifier le tournage tout en prévoyant les effets et les significations que l'on souhaite montrer aux spectateurs. Le scénarimage, pour sa part, sert à visualiser de manière graphique (à partir de dessins dans la plupart des cas) les cadres

et leurs séquences. On peut dire que c'est une anticipation du montage avant que les scènes ne soient filmées. Les divers médias pouvant être déployés sur le Web révolutionnent la scénarisation dans la mesure où un travail de collaboration interdisciplinaire est nécessaire afin de fournir à l'apprenant un matériel didactique adapté en fonction de sa compréhension, de sa motivation et de sa mémorisation.

1.4.2.3 Médiatisation

Médiatiser veut dire: diffuser par les médias. Pour la réalisation de la médiatisation, il est important que les séquences du cours soient conçues, les scénarios d'apprentissage préalablement établis et les médias à intégrer bien définis. La médiatisation des cours permet aux enseignants d'enrichir leurs pratiques pédagogiques en utilisant les technologies multimédia tout en exerçant leur responsabilité pédagogique dans l'étape de conception. Un certain nombre d'études empiriques mené par (Najjar, 1998) suggèrent la manière de sélectionner et de combiner des médias (appelé aussi modalités d'intégration des médias) pour présenter avec succès des types spécifiques de contenus pédagogique. Ces modalités d'intégration des médias ont été résumées dans (Atif, 2003) avec le tableau suivant:

Contenu d'enseignement	Média
Instructions d'assemblage	de texte avec des images de soutien.
Les informations	texte présentant de l'animation ou vidéo.
Résolution de problèmes	d'animation avec une narration verbale
Reconnaissance	photos avec le texte ou la narration verbale.
Verbale,	sonore ou vidéo et texte.
Histoire détaillée	de la vidéo avec d'une bande son.

Table 1. *- Allocation des média*

Par exemple, dans l'apprentissage d'une langue étrangère particulière, il serait plus utile pour un apprenant d'entendre les mots. Mais certains mots dépendants du contexte peuvent être mieux compris si ces mots ont été montrés avec une vidéo. En outre, pour aider l'apprenant d'un langage, une version textuelle des mots de phonétique permettrait de consolider et renforcer le processus d'apprentissage si elle était mise avec une information auditive.

1.4.3 Développement

Il s'agit ici de numériser le contenu pédagogique par le choix et la préparation des médias et de l'outil, par le développement des modules (numériser les textes, conception et mises en œuvre des interfaces, etc.) et par l'assemblage du contenu.

1.4.4 Test

La création classique d'un contenu consiste à parcourir plusieurs fois le même cycle. En effet, après de nombreux aller-retour entre les deux étapes de conception et de réalisation, on passe à la phase de test. Cette phase conduisant très souvent à un retour qui permet d'affiner le scénario pédagogique. L'affinage successif est indispensable, car un enseignant a du mal à se focaliser sur l'élaboration d'un scénario pédagogique, et ceci vient du simple fait que cela ne fait pas partie de sa manière de penser.

1.4.5 Diffusion

C'est la dernière phase du processus d'élaboration. Elle consiste au choix du format et du support de publication : Internet, CD-ROM, etc. Et aussi du mode de diffusion qui peut-être synchrone ou asynchrone.

Bien que ce CPM a pour support un document dit document multimédia, il faut que nous mettrons en lumière ce type de document du point vue de ses quatre dimensions (nous détaillerons la dimension temporelle), ses langages (ou standards) de spécifications. On va étudier par la suite le standard de spécification SMIL. Ce dernier représente le langage de spécification à utiliser pour spécifier les cours de phonétique envisagé.

1.5 Modélisation d'un document multimédia

Un modèle de document multimédia doit permettre de représenter toutes les relations qui peuvent exister entre les différents objets. Ces relations peuvent décrire l'organisation logique du document, sa présentation spatiale, sa *synchronisation temporelle* ainsi que l'interconnexion entre ses différents éléments (Layïada, 1996). Ces différentes relations sont appelées les relations multimédias, les différentes structures ou bien encore les dimensions d'un document multimédia.

1.5.1 Dimension logique

La dimension logique concerne le regroupement des objets du document en entités sémantiquement liées. Par exemple, dans un document utilisé pour une représentation orale (suite de transparents), chaque diapositive est généralement composé d'un titre et d'un corps, qui à son tour peut contenir d'autres objets comme des images, de l'audio ou de la vidéo.

1.5.2 Dimension spatiale

La dimension spatiale concerne la disposition des objets selon les différents canaux (audio, fenêtre d'écran, etc.). Par exemple, le titre de la diapositive peut apparaître 2 centimètres plus haut que son corps ; il occupe une largeur de 80% en comparaissant de celle de la page et il est centré.

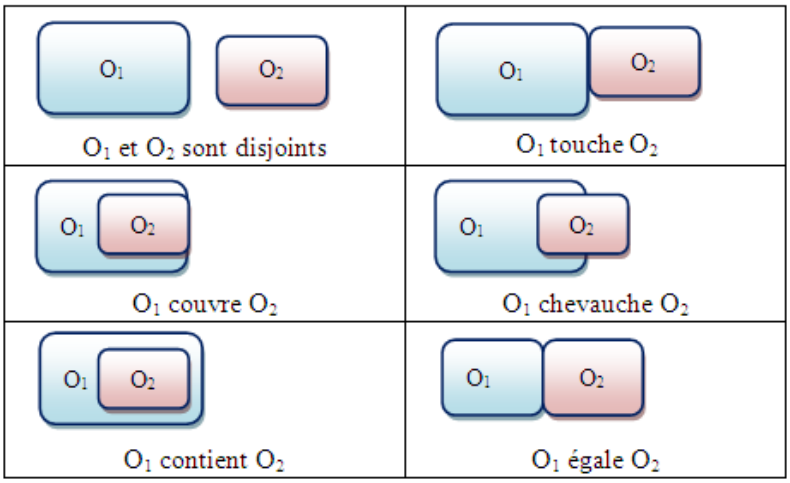


Figure 4. *Relations spatiales entre objets média O_i*

Jedidi (Jedidi, 2005) a défini les relations topologiques pouvant être établies entre les objets média et qui sont illustré dans la figure 4 par : disjoint, touche, couvre, chevauche, contient, égal.

Ainsi, les documents que nous qualifierons par la suite de cette thèse, sont non seulement caractérisés par des contenus de natures diverses : statique comme les textes, graphiques ou dynamique comme le son, les vidéo et les animations, mais aussi par l'organisation temporelle de leurs composants. Dans cette thèse, nous avons convenu que les unités d'information considérées comme atomiques sont *appelées objets*

multimédia et la description de leur enchaînement dans le temps est appelée *scénario temporel*.

1.5.3 Dimension temporelle

Cette dimension concerne la disposition des objets du document dans le temps. Par exemple, le corps de la diapositive doit apparaître 2 secondes après le début du titre. La définition de la structure temporelle d'un document consiste à spécifier le scénario temporel ou bien les schémas de synchronisation entre les objets et les éléments composites du document. Nous donnons dans ce qui suit un exemple illustrant bien cette dimension.

Un document multimédia temporisé (supportant la dimension temporelle) est présenté dans la figure 3. On y observe le déroulement de la présentation sur un axe temporel gradué. L'exemple proposé est l'introduction d'un exposé, sous la forme d'une séquence d'objets graphiques pouvant être présentés simultanément. Cette introduction présente l'auteur, le titre et la table des matières de l'exposé et chacun de ces objets est représenté par un rectangle. Les instants de début et de fin de représentation de chaque objet sont représentés respectivement par les bords gauche et droit du rectangle associé

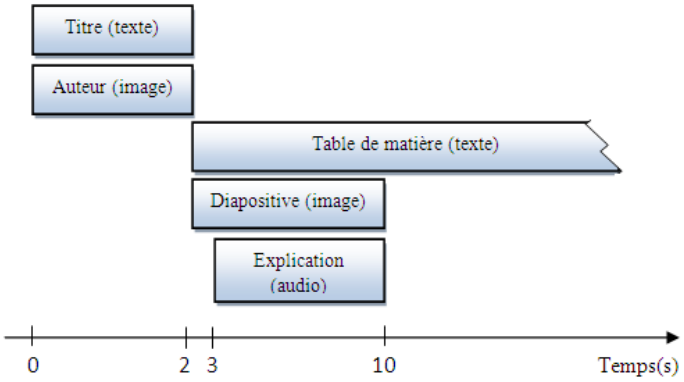


Figure 5. *Scénario temporel d'une présentation*

Les deux objets auteur et titre démarrent au début de la présentation et durent jusqu'à la seconde 2. Entre les secondes 2 et 10 s'affiche l'image d'une diapositive en parallèle avec son explication auditive. Cette dernière est en retard d'une seconde par rapport à l'image. Cette description est exacte et quantitative, car elle définit exactement les instants de début et de fin de la présentation de chaque objet multimédia du document.

Les spécifications ne sont pas forcément exprimées de manière exacte ou quantitative car il peut être utile à l'auteur de laisser la machine interpréter la spécification, dès lors qu'elle exprime suffisamment et clairement le désir de l'auteur. Cela permet aussi, comme l'a remarqué (Euzenat, 2003) de ne pas se soucier de caractériser toutes les positions mais de laisser l'interprète les déduire.

Il faut également citer la précision de (Euzenat, 2003), que ces différentes dimensions qui composent un document sont parfois liées entre elles, et nécessitent par conséquent un traitement conjoint. C'est le cas par exemple lorsque l'auteur souhaite spécifier le déplacement d'un objet sur l'écran où il est nécessaire d'exprimer une information spatiale pour la trajectoire, les positions initiale et finale correspondantes au déplacement, ainsi qu'une information temporelle tel que la durée du déplacement et son instant de début.

1.5.4 Dimension hypermédia temporel

Les liens hypermédia sont décrits par des objets particuliers du document qui permettent de relier les portions de différents documents indépendamment de leurs positions dans cette structure. Ces liens permettent de définir des relations de type sémantique entre des documents ou des parties de documents, comme les renvois et les références.

Dans le cas des documents multimédias temporisés supportant la dimension temporelle, la nature dynamique de l'information rend nécessaire la navigation. Ainsi, des hyperliens classiques doivent être étendus pour prendre en compte la dimension temporelle. Ainsi nous obtenons un nouveau type de navigation : **La navigation temporelle**.

Cette nouvelle forme de navigation est conçue par l'auteur du document de façon à aider les lecteurs dans leur parcours du document temporel. Elle est définie par des objets activables (ou boutons d'interaction). Ces derniers sont des objets sur lesquels une action prédéfinie dans le document (exemple : activation d'un lien temporel) est déclenchée lorsqu'ils sont activés, et donc nous obtenons la structure hypermédia temporelle du document (Cécile, 1999).

Ce type de lien est défini par un ancre de départ (objet localisé spatio-temporellement), un ancre d'arrivée (autre objet) et le lien qui porte une certaine sémantique (renvoi, référence, annotation, etc.). Au niveau de l'interface utilisateur, un lien nécessite d'être activé explicitement par une interaction de l'utilisateur.

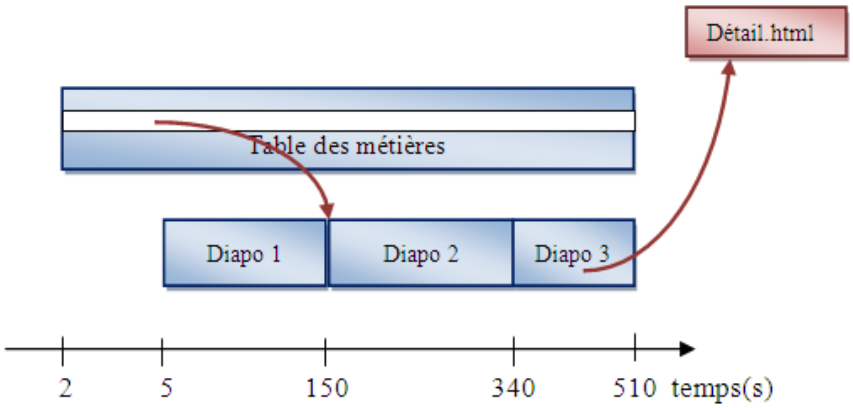


Figure 6. *Exemple de liens temporel*

Cet aspect est illustré sur la figure 6, le rectangle blanc représente une région spatiale de l'objet « Table des matières » et sa langueur représente l'intervalle de temps durant lequel le lien associé à cette région est actif sur l'objet dans lequel il se trouve. Dans l'exemple, on constate que ce lien (une entrée de la table) permet d'avancer la présentation à l'instant du début de l'objet diapo 2. L'objet diapo 3 est un lien vers une page web contenant un peu de détail. L'activation du lien est basée sur une interaction explicite de l'utilisateur.

La seule nuance est que d'une part, l'activation de l'ancre de départ est restreinte au laps de temps couvert par sa présentation et que d'autre part, l'ancre d'arrivée correspond non seulement à un autre objet ou document mais aussi à un instant précis de sa présentation (Euzenat, 2003).

Notre travail traite des documents multimédias ayant un caractère particulier : ils sont *structurés*. Les documents structurés peuvent être représentés (dans un langage de balisage) sous forme d'arbre hiérarchique. Ceci permet de faire référence à leurs éléments constitutifs. Une autre caractéristique souvent associée est la notion de lien hypertexte. Ces liens référencent des éléments (images ou autres documents) comme s'ils faisaient partie du document initial. Ainsi, nous pouvons réutiliser certaines parties de documents dans plusieurs autres. Cette décomposition de l'information supportée

par ces documents a pour premier objectif de faciliter leurs portabilités ainsi qu'un traitement automatisé, de grandes masses de documents, par des applications variées.

Nous exploitons ainsi une autre caractéristique de ces documents, à savoir leur description dans un langage de spécification permettant d'exprimer la position des objets médias suivant les différentes dimensions. Ces langages peuvent être standardisés (HTML, SMIL, XML) ou pas (Madeus, LaTeX, ou autres).

Nous nous intéressons particulièrement dans cette thèse à la prise en compte de la dimension temporelle lors du traitement de ce type de documents. Nous allons, par la suite examiné les différentes catégories de langages de spécification permettant d'exprimer cette dimension, de même que les fonctionnalités supportées par les systèmes d'édition de ce type de documents.

1.6 Langages de spécification

Les travaux actuels sur les documents multimédia portent essentiellement sur la spécification des scénarios temporels afin de prendre en considération cette nouvelle caractéristique. Nous citons trois modes de spécification temporelle qui ont été expérimentés, à savoir : la spécification par placement absolu ou timeline (spécification des instants de début et de fin de vie des objets média), l'utilisation des langages de programmation ou d'opérateurs de spécification ou l'utilisation d'algèbre de relations.

Victor (Euzenat, 2003) a classé ces différentes techniques de spécification en deux classes : quantitative (exacte) ou qualitative (l'exemple de la figure 5 illustre la première approche). Cependant, un travail de synthèse a été fait sur ces différents modes par Jourdain (Jourdain, 1999), qui a permis de proposer une autre classification de ces techniques en trois approches :

- Les techniques **opérationnelles ou impératives** : nous citons celles qui se basent sur l'axe de temps absolu comme Director Macromedia (Macromedia, 1998), et le langage HyTime (HyTime, 1997) et celles qui se basent sur la programmation à base de scripts comme Lingo dans Director et le standard MHEG (MHEG, 1995).
- Les techniques **déclaratives par opérateurs temporels** qui correspondent à la spécification fondées sur les graphes plats tel que les langages Firefly (Buchanan, 1992) et HTSPN (Sénac, 1996), ou les graphes hiérarchisés (arborescents) tels que le langage CMIFed (Van, 1993) et le standard SMIL (SMIL, 1998) (SMIL, 2001). La

dernière approche consiste à exploiter l'organisation logique du document pour décrire et mettre en œuvre sa synchronisation temporelle. Le principal avantage de cette approche est la possibilité d'organiser le document en modules indépendants sur lesquels on peut appliquer des primitives globales de synchronisation. Ces primitives s'appliquent sur des intervalles et permettent la mise en parallèle ou en séquences des éléments appartenant à une entité logique.

- Les techniques **déclaratives par relations temporelles**, Madeus (Layaïda, 1997), dans lesquelles l'auteur déclare les placements temporels souhaités sans donner toutes les informations temporelles attachées aux objets. La plupart de ces techniques s'appuient sur l'algèbre d'Allen (Allen, 1983) pour la spécification des contraintes temporelles.

1.6.1 Le standard SMIL

SMIL (Synchronized Multimedia Integration Language) est un langage de spécification de documents multimédia basé sur XML. Il permet de caractériser les exigences spatiales, structurelles ainsi que temporelles d'une présentation multimédia. SMIL est un langage déclaratif recommandé par le W3C (World Wide Web Consortium) permettant la spécification des présentations multimédia interactives par l'intégration et la synchronisation de contenu multimédia diversifiés (image, son, vidéo, animation, flux de texte (streaming)).

La dernière version du SMIL, la version 3.0, publié en 2008 a apporté des modifications sur les anciennes versions et a intégré de nouveaux modules permettant aux auteurs une caractérisation basée sur le profil de plus en plus flexible. La structure d'un document SMIL et la syntaxe de ce langage sont données d'une manière détaillé dans l'annexe C.

1.7 Conclusion

Cette partie nous a permis de cadrer notre travail et de présenter les possibilités offertes par le standard SMIL avec lequel on peut créer un contenu pédagogique multimédia temporisé et interactif qui répond assez bien aux problématiques soulevées dans l'introduction.

Notre challenge principal est d'intégrer les pédagogues, les psychologues ainsi que les standards et techniques de représentation des connaissances (plus précisément les ontologies) au sein du processus de modélisation afin de favoriser la création,

l'interopérabilité et l'indexation sémantique des contenus multimédia pédagogiques. Cela implique d'une part, des outils de création plus abordables pour les enseignants, qui ont peu de connaissances des outils informatique, et d'autre part des outils d'indexation et de recherche par la sémantique du contenu pour les apprenants.

Pour apporter une solution à ces problèmes, nous verrons en détail dans le chapitre suivant les différents systèmes auteurs qui permettent la création de contenu pédagogique ayant le format SMIL. La deuxième partie sera consacrée à étudier les différentes approches d'indexation, par la sémantique, des documents multimédia dans le contexte du e-learning et en particulier des cours vidéo. Nous nous attachons à travers cette étude synthétique à répondre aux questions suivantes: quel sont les modèles de CPM ayant le format SMIL (appliqué en e-learning) existants ? Est-ce qu'elles concrétisent l'approche double codage de l'information ? Quels sont les systèmes auteurs qui offrent la possibilité de créer un cours de phonétique? Quels sont les approches utilisées pour l'indexation sémantique des cours vidéo ? Quels sont leurs avantages ? Leurs inconvénients ? Et quels problèmes restent à résoudre?

Chapitre 2 : Approches de construction et d'indexation de documents multimédias

2.1 Introduction

Le présent chapitre traite deux axes différents à savoir : les approches de création de documents multimédia pédagogiques et les approches de leurs indexations par la sémantique. Dans sa première partie, le chapitre présente les critères, les fonctionnalités et les contraintes qui dépendent des outils de création appelés aussi systèmes auteurs. Dans le cadre du e-learning, plusieurs systèmes ont été conçus afin de faciliter l'édition et création de contenus en ligne. Donc cette partie présente une étude synthétique et critique des systèmes auteurs existants qui aident à la création des présentations multimédia ayant le format SMIL dédiées à l'enseignement de la phonétique.

La deuxième partie s'intéresse à l'indexation par la sémantique de ce type de documents. Donc, cette partie illustre l'indexation classique dans un SRI, tout en détaillant le principe d'un modèle classique utilisé qui est le modèle vectoriel. Puis elle explique les techniques d'intégration des ontologies dans ce système pour avoir une indexation et une recherche par la sémantique. Ensuite, elle montre une étude analytique et critique des systèmes de recherche par la sémantique dans des corpus de type vidéo et en particulier ceux dédié aux e-learning.

2.2 Approches de construction

2.2.1 Système auteur (définition, critères et fonctionnalités)

Un système-auteur peut être défini comme un « environnement de développement logiciel de haut niveau », entendant par-là qu'il permet en théorie, grâce à une interface graphique, de réaliser l'essentiel ou la totalité d'une application multimédia sans

utiliser un langage de programmation. Nous Excluons le cas où l'auteur veut aboutir à une interaction un peu plus complexe (Bousbia, 2005).

Avec un système auteur, en principe, le temps d'apprentissage et de développement d'un logiciel multimédia est inférieur de beaucoup en temps requis comparativement à un autre système de programmation.

Un système auteur, selon (Psyché, 2007), offre au concepteur pédagogique le moyen de concevoir un système d'apprentissage en fonction d'une expertise pédagogique. Ces systèmes s'élaborent généralement en tenant compte d'une « théorie du design pédagogique ».

Une des classifications possibles de systèmes auteurs revient à (Murray, 2003). Le tableau 2, résume les sept catégories de la classification de Murray.

No	Catégories des systèmes	Exemples de systèmes faisant référence
1	Systèmes de séquencement et de planification du curriculum	DOCENT, IDE, ISD Expert, Expert CML ;
2	Systèmes à stratégies pédagogiques	Eon, GTE, REDEEM (et COCA), SmartTrainer
3	Systèmes de simulation et d'entraînement	DIAG, RIDES, MITT-Writer, ICAT, SIMQUEST, XAIDA
4	Systèmes experts et tuteurs cognitifs	Demontr8, D3 Trainer, Training Express
5	Systèmes à connaissances multiples	CREAM-Tools, DNA, ID-Expert, IRIS, XAIDA
6	Systèmes à usages spécifiques	IDLE-Tools/IMap, LAT
7	Systèmes hypermédia intelligents / adaptatifs	CALAT, GETMAS, Interbook, MetaLinks

Table 2. *Classification des systèmes auteurs fondateurs (Murray 2003)*

Une autre classification présentée dans (LCM 2003) classifie ces systèmes en deux catégories à savoir : les outils spécifiques au e-learning, tels qu'Authorware ou Director, et les outils plus généralistes comme les éditeurs HTML, tels que FrontPage ou Dreamweaver, ou les éditeurs de simulation tels que Flash. Face à une évolution technologique que l'on sait de plus en plus rapide, le développement des systèmes auteurs dans une optique éducative est devenu un domaine de recherche à part entière. En effet, au cours des dernières années il y a eu des progrès significatifs dans le développement de tels systèmes de façon à concevoir des outils permettant aux utilisateurs de créer des contenus pédagogiques multimédias utilisables sur CD-ROM

et/ou en ligne. Parmi ces systèmes citons Toolbook et Serpolet Auteur. Dans ce qui suit, nous présentons les critères et les fonctionnalités qu'un système auteur doit avoir et offrir.

2.2.1.1 Critères

Quel que soit le type et le niveau de l'utilisateur du système auteur, plusieurs attributs doivent être étudiés soigneusement et aussi considérés par le concepteur d'un système auteur. Nous allons présenter ci-dessous les critères considérés comme principaux, d'après (Blandine, 2000) :

- **La convivialité** : La première approche avec un logiciel a un rôle psychologique fondamental pour tout utilisateur, même chez des informaticiens. En effet, si l'écran ou la présentation du logiciel n'est pas soigné, l'utilisateur peut être plus ou moins bloqué pour la suite. L'importance de l'interface homme-machine justifie la généralisation des icônes et l'intégration du multimédia dans les systèmes auteurs pour pouvoir projeter son idée, faire comprendre, expliquer, fabriquer ou modéliser. Le concepteur doit par conséquent attacher une attention particulière à la présentation de son produit.

- **La transparence** : la gestion des données, et le fonctionnement interne du point de vue relationnel entre les différents éléments (variables, objets, etc.) doivent être complètement transparents à l'utilisateur. L'utilisateur ne doit pas se soucier ni du moment ni de l'endroit, ni du type de fonctions et librairies à inclure et à lancer pour l'exécution de ces diverses tâches. En effet, si l'utilisateur doit comprendre le fonctionnement des registres de données, la notion de système auteur n'a plus lieu d'être !

- **L'assistance** : Comme un système auteur est supposé, entre autres, être un outil de programmation pour les non-informaticiens, il est fondamental d'assurer un minimum d'aide en ligne ou au moins une partie explicative des principales tâches ou commandes ou icônes du système. Même pour des tâches devenues banales comme la saisie de données textuelles, le système doit au moins indiquer à l'utilisateur dans quelle fenêtre et quand cette peut se faire.

- **L'interactivité** : la communication entre le système auteur et son utilisateur, souvent sous la forme d'une assistance de test et de contrôle automatique des tâches de l'utilisateur, doit être particulièrement soignée car elle est vitale pour l'utilisateur et pour la réussite du produit sur le marché. Un système auteur doit intéresser son utilisateur et non l'ennuyer.

- **La fiabilité** : le système doit être le moins bloquant possible quelle que soit l'action menée par l'utilisateur sur le logiciel. Ce dernier doit pouvoir revenir en arrière, modifier ou supprimer sans difficulté tout composant de son logiciel. De même, il doit pouvoir assembler les pièces qui constituent son application dans un ordre qui ne soit pas trop contraignant, indépendamment des uns et des autres et ce à n'importe quel moment du développement.

2.2.1.2 Fonctionnalités

Il existe plusieurs fonctionnalités des systèmes auteurs; on présentera celles qui sont communes (Murray, 1999):

- **Fonctionnalités basiques** : Les fonctionnalités telles que copier/coller, trouver et annuler sont basiques mais doivent être implémentées dans un système auteur. La majorité des systèmes les possède même si annuler une action peut être parfois un problème complexe.

- **Utiliser des paradigmes familiers** : le fait d'utiliser des paradigmes familiers aux utilisateurs de logiciels est un principe plus qu'une fonctionnalité mais néanmoins un principe essentiel. En effet, il s'agit d'avoir un outil **ergonomique**. Il faut que les boutons et/ou menus soient aisément identifiables et pour cela autant reprendre les paradigmes classiques. Il est aussi important de garder des similarités avec les outils de création de contenus pédagogiques traditionnels.

- **WYSIWYG** (What You See Is What You Get) ou un prototyping rapide : Comme dans les environnements de développement rapide (Delphi, Visual C++...), l'utilisateur doit pouvoir rapidement créer et tester son système. Un rapide aller-retour édition-test est souhaitable.

- **Conception graphique** : L'utilisateur des systèmes auteurs, est censé être un enseignant ou un formateur sans connaissances informatiques préalables. La conception graphique ou visuelle est donc incontournable. Le formalisme est représenté par des icônes ou dessins et il doit être aisément compréhensible et mémorisable. Une vision claire et intuitive de son travail doit être offerte à l'utilisateur.

Puisque les documents temporisés nécessitent un traitement particulier pendant son édition. Dans ce qui suit, nous présentons les difficultés rencontrées dans leur édition et les fonctionnalités et contraintes que doivent être supportés par des systèmes auteur qui traite ce type documents en particulier.

2.2.2 Difficultés et contraintes dans l'édition de documents temporisés

Le processus de définition du scénario temporel d'un document multimédia est complexe car l'objet à construire est un objet dynamique dont le comportement varie d'une présentation à une autre, du fait des réactions aux interactions du lecteur. L'environnement utilisé par l'auteur doit l'aider à maîtriser cette complexité et lui faciliter autant que possible la tâche.

La nature dynamique des objets manipulés, tels que la vidéo et l'audio, ainsi que la définition de leur ordonnancement temporel (le scénario) rendent plus complexe la réalisation d'outils auteur. Le principe statique du WYSIWIG, dans lequel l'information présentée à tout instant du processus d'édition correspond à l'information finale, ne peut s'appliquer à l'édition du scénario temporel des documents multimédias. Il n'est en effet pas possible de spécifier un comportement dynamique, par exemple un enchaînement entre deux vidéos, et d'en percevoir de façon immédiate et instantanée le résultat. C'est pourquoi, Cécile et al [Cécile, 1999] ont distingués deux étapes dans le processus de conception des documents multimédias temporisés. Ces deux étapes se présentent dans les deux fonctions suivantes qui doivent être supportées par un système auteur:

1. Fonction d'édition : permettre de réaliser les opérations de création, de construction et de modification du document par un auteur. L'opération de construction d'un document consiste à y inclure des objets multimédias de base et ensuite à spécifier des relations entre ces différents objets. Ces relations peuvent être liées à leur organisation logique, leur disposition spatiale sur les différents canaux à travers le temps, ou encore à leur synchronisation temporelle.
2. Fonction de présentation : Elle consiste à présenter à l'utilisateur (le lecteur) le contenu du document une fois que son édition est achevée. Cette phase consiste à lui fournir un ensemble de commandes permettant d'explorer ou de naviguer dans l'espace du document pour découvrir l'information qu'il contient à travers l'espace, le temps et l'interaction avec le document.

Il est à noter que ces deux phases ne sont pas nécessairement dissociées. Car le cycle d'édition d'un document est formé par une première phase pendant laquelle on

introduit une ou plusieurs contraintes, puis d'une seconde phase de présentation pendant laquelle on vérifie que le comportement du document correspond bien à l'effet recherché. C'est *l'approche incrémentale* de l'édition des documents temporisés.

L'environnement auteur doit permettre de modifier facilement les documents en cours de conception. La construction d'un document est un processus itératif dans lequel le document courant subit de nombreuses modifications. Pour cela, l'auteur doit pouvoir :

- Effectuer des modifications locales sans avoir à reconsidérer globalement le document. Par exemple, l'auteur doit pouvoir modifier la durée d'un objet sans pour autant mettre en cause toute le reste des synchronisations temporelles qu'il a pu spécifier auparavant. C'est la *propagation automatique des contraintes temporelles* après chaque modification.
- Passer aisément du mode édition vers le mode présentation (et vice-versa), pour d'une part aller voir le résultat de ces modifications (du mode édition vers le mode présentation), et d'autre part retrouver facilement les informations qu'il souhaite modifier (du mode présentation vers le mode édition). Ceci est pour objectif de concrétisé *l'approche incrémentale.*
- L'environnement auteur doit offrir un support visuel permettant de percevoir le scénario en cours de spécification. L'auteur a besoin d'avoir une vue globale du scénario qu'il est en train de construire.

2.2.3 Approches d'élaboration des systèmes auteur

Un système auteur qui permet l'édition et la création de documents temporisés est celui qui permet de développer une structure narrative basée sur une collection de ressources médias et qui gère les flux visuel et temporel de la présentation. Du point de vue technique, il est nécessaire que le système sous-jacent soit capable d'assurer la synchronisation entre plusieurs médias, tout en tenant compte de leurs caractéristiques respectives.

Le projet OPERA[1] [Opéra2001] de l'INRIA[2] est entièrement dédié au thème des documents électroniques structurés, hypertexte et multimédia en général. Il s'intéresse à la spécification des différentes dimensions du document et particulièrement à la dimension temporelle.

Un travail a été effectué par de nombreux collaborateurs au projet (stagiaires, ingénieurs, doctorants et chercheurs) sur le thème de conception d'environnements auteurs qui répondent aux différents critères cités précédemment. Cécile, qui est un membre, a résumé les axes de travail sur ce thème de recherche ; nous en citons les principaux:

- La conception d'un ***environnement multivues*** pour l'édition de documents multimédias. L'idée de base est que les multiples facettes d'un document multimédia sont plus facilement accessibles à l'auteur au travers de différentes vues, qui sont autant de filtres sur le document. Ainsi, outre la vue de présentation qui permet de jouer le document (scénario), on peut définir une vue des objets organisée sous forme d'une hiérarchie de composants ainsi qu'une vue du scénario qui visualise l'enchaînement temporel des objets du document. Ces vues peuvent être synchronisées entre elles (voir SMIL-Editor, LimSee).
- La conception d'***une boîte à outils*** appelée 'Kaomi' qui permet le développement d'environnements auteur/lecteur de documents adaptés à différents langages avec un minimum d'efforts concernant la conception et la réalisation. Elle offre un ensemble de services qui mettent en œuvre les principes d'édition (environnement multivues, édition directe, etc.) issus des travaux du projet Opéra. Cette Boîte à outils a servie à la création de différents environnements auteurs : Madeus-Editeur [Layaïda1997], SMIL-Editeur, MHML-Editeur.

[1] OPERA : Outils Pour les documents Electroniques, Recherche et Applications

[2] INRIA : Institut National de Recherche en informatique et Automatique

Au sujet de la conception et de la mise en œuvre de nouveaux systèmes auteurs, deux tendances se dessinent selon Reyes (Reyes, 2006):

- les systèmes pourraient *raffiner les interfaces graphiques* des auteurs afin de rendre le processus d'élaboration plus facile,
- les systèmes pourraient s'automatiser à un tel point que les auteurs ne définissent plus que le contenu et que l'utilisateur final affiche la présentation à la demande, c'est-à-dire en *automatisant la génération des présentations.* Bulterman et Hardman affirment que suivre une ligne comme cette dernière implique de renoncer à produire des *présentations exceptionnelles* et plutôt de se contenter du transfert de l'information nécessaire.

Parmi les objectifs de notre travail, il y a celui qui consiste en la réalisation d'un système auteur destiné aux enseignants des langues afin de créer un cours de phonétique. De ce fait une étude synthétique de quelques systèmes auteur dans ce domaine est obligatoire afin de profiter de leurs avantages et compléter ce qui manque. Mais avant d'étudier ces systèmes, nous devons mettre en lumière les concepts les plus essentiels utilisés dans l'enseignement de la phonétique.

2.2.4 Enseignement de la phonétique

2.2.4.1 Que ce que la phonétique ?

La phonétique (du grec « phônêtikos », où « phônê » qui signifie la « voix », le « son ») est une branche de linguistique qui étudie les sons utilisés dans la communication verbale. Elle concerne les sons eux-mêmes (les « phones »), leur production, leur variation plutôt que leur contexte. La phonétique se divise en trois branches :

- la phonétique **articulatoire**, qui étudie les positions et les mouvements des organes utilisés pour la parole par son émetteur.
- la phonétique **acoustique**, qui étudie la transmission de l'onde sonore entre son émetteur et son récepteur.
- la phonétique **auditive**, qui se préoccupe de la façon dont les sons sont perçus et décodés par son récepteur. Nous nous intéressons dans notre étude du troisième type.

Quand on représente les sons d'une langue, on se sert de l'Alphabet Phonétique International (API), un système partagé par la plupart des linguistes. Dans cet alphabet, il existe un symbole pour chaque son. Quand on se sert de cet alphabet pour représenter

les prononciations, on entoure la représentation par des crochets. Ainsi, pour écrire le mot *chat*on mettrait [ʃa]. Voir le manuel de (Pullum, 1986) pour le tableau complet des sons du API. C'est la transcription phonétique.

Soit l'exemple suivant : Voyez et lisez ces quatre mots : flood, foot, soon, door. Ils se prononcent différemment et pourtant ils contiennent exactement les mêmes voyelles. Si vous cherchez le sens des quatre mots cités plus haut, vous trouverez tout d'abord leur transcription phonétique.

Flood	Foot	Soon	Door
[flʌd]	[fʊt]	[su:n]	[dɔ:]

Ci-après d'autres notions à appréhender dans l'enseignement de la phonétique d'une langue :

Un phonème est la plus petite unité discrète ou distinctive que l'on puisse isoler par segmentation dans la chaîne parlée. C'est une entité abstraite, qui peut correspondre à plusieurs sons. Il est en effet susceptible d'être prononcé de façon différente selon les locuteurs ou selon sa position et son environnement au sein du mot.

Une diphtongue est une voyelle dont le point d'articulation varie entre deux sons de base pendant sa réalisation. Ce mot vient du latin d'origine grecque « diphthongus » (de « diphthoggos », double son). Dans certaines langues, dont l'anglais, on peut trouver également des triphtongues dont la qualité varie deux fois.

Une syllabe : est une unité ininterrompue du langage oral (Tracy, 2006). Son noyau, autour duquel elle se construit, est généralement une voyelle. Une syllabe peut également avoir des extrémités précédant ou suivant la voyelle, qui, lorsqu'elles existent, sont toujours constituées de consonnes. Notons que la syllabe constitue l'objet d'étude de la phonétique

L'accent tonique met en relief l'émission d'une syllabe dans un mot en augmentant l'intensité de la voix ainsi qu'en élevant, généralement, le ton. La syllabe frappée de l'accent est dite tonique, et les autres atones. On distingue fréquemment les langues à accent tonique des langues à tons et à accent de hauteur.

Dans ce qui suit, nous présentons l'étude de quelques systèmes existant tout en déterminant s'ils offrent la possibilité à l'enseignant de langue de préparer un cours de phonétique en concrétisant l'approche double codage.

2.2.5 Systèmes existant pour l'enseignement de la phonétique

2.2.5.1 Outils informatique de prononciation

En ce qui concerne la phonétique acoustique, les progrès de représentation graphique sur écran sont spectaculaires depuis quelques années. Les logiciels tel que (speaker, tell me more, English plus, ou Voice book) utilisent des *oscillogrammes* pour présenter la voix mais ils ont un apport limité.

Figure 7. *Affichage de la courbe mélodique avec surlignage en rouge des segments correspondant aux syllabes accentuées de l'exemple « I wonder if you could help me ».*

Tandis que WinPitch (Martin, 2005) ou Speech analyser, qui utilisent des *courbes de fréquences* fondamentales, ils ont une utilisation trop restreinte en raison de leur complexité ou de leurs erreurs d'ergonomie.

Le système auteur Sound Right se base sur les courbes fondamentales pour dessiner des *courbes intonatives* simplifiées au moyen de *flèches extensibles* qui s'affichent en dessous le texte. Péchou (Péchou, 2002) a mentionné que la difficulté d'interprétation des courbes complexes expliquent leur utilisation limitée en cours de langue dans les établissements scolaire et à l'université.

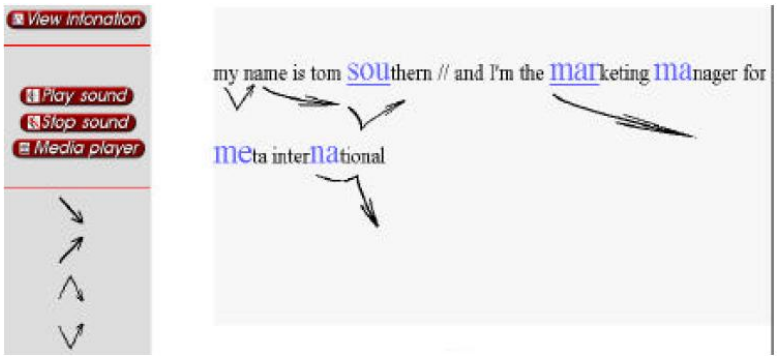


Figure 8. *Le système auteur Sounds Right. Les flèches extensibles sont traînées en place à partir de la colonne à gauche et ensuite modifiées pour obtenir une taille appropriée.*

Prosodic Font est un système développé au Media Lab MIT (Rosenberger, 1999) dans le but de générer automatiquement des polices de caractères dynamique (à partir d'un discours oral en entrée) qui varient avec le temps et la variation de sonorité d'un discours parlé. Le but est de générer des textes animés en fonction de l'intonation et de la prosodie du discours. Une telle solution n'est pas admise comme meilleur solution didactique pour l'enseignement de la prononciation.

2.2.5.2 Systèmes auteur pour un contenu pédagogique en SMIL

Dans cette section, nous allons étudier brièvement trois éditeurs de documents SMIL :

SWANS (Beck, 2005) est un système auteur qui permet à tout enseignant de générer semi automatiquement des documents multimédia où l'accent tonique est marqué visuellement (par des marqueurs typographique tel que la couleur, le style,…) et auditivement. Le document généré est une page web où l'apprenant a la possibilité de lire et/ou écouter un discours de manière synchronisé avec le texte annoté. Le scénario d'édition par ce système, nécessite au départ l'importation des média (texte, audio, vidéo) dans l'environnement de travail. Ensuite, la synchronisation du texte (qui est segmenté en unités de souffle) avec sa prononciation audio ou vidéo. Enfin, l'enseignant peut annoter le texte par des marqueurs typographiques.

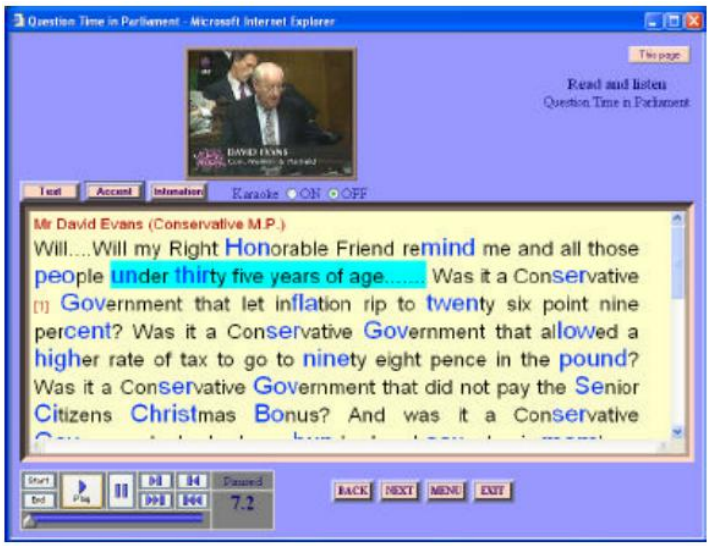


Figure 9. *Interface générée par SWANS*

Le système LimSee3 (Roisin, 2006) (Roisin, 2008) est un éditeur multimédia de nouvelle génération, qui utilise des modèles de documents pour simplifier l'édition et alléger les tâches répétitives. Il permet en plus de générer les documents selon différents formats de sortie (SMIL, XHTML+javascript et time-sheets). Actuellement, il existe trois modèles de document qui sont articulés autour de la construction d'un cours multimédia.

Le premier modèle permet de construire un diaporama (ensemble de transparents) destiné à préparer un support de cours. Chaque diapositive peut contenir un ou plusieurs média. Ces médias sont insérés ou importés de l'extérieur par de simples gestes (copier-coller ou glisser-déposer). Le deuxième modèle de document permet de construire un cours enregistré. Dans ce cas, l'édition nécessite d'abord, l'importation des transparents utilisés pendant le cours, de l'image de l'enseignant et de la piste audio (sa voix enregistrée), ensuite, la synchronisation entre eux (ajuster les transparents par rapport à la bande audio ou vidéo) pour que les transparents défilent à l'écran au bon rythme. L'outil de synchronisation permet de réécouter la bande audio et indique par des clicks les moments où il faut changer le transparent. Le troisième modèle permet d'annoter en temps réel des interrogations des élèves.

Mcours.com
www.mcours.com
Cours, exercices & projets de fin d'études

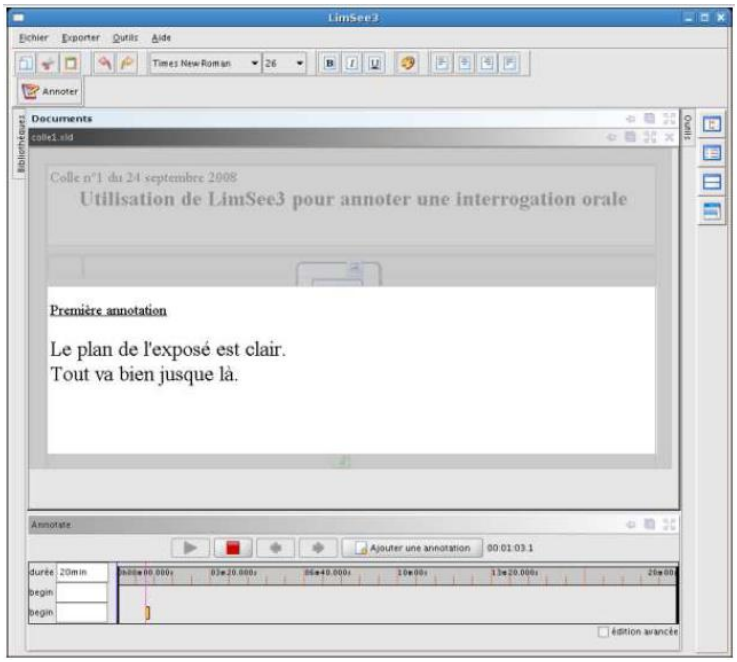


Figure 10. *Vue globale de LimSee3 avec la première annotation*

L'éditeur ECoMaS (Merzougui, 2004) est lui aussi est un éditeur de cours médiatisé qui est basé sur un modèle de document. La présentation finale des cours générés est la même que celle du deuxième modèle de LimSee3, mais le scénario d'édition est différent. L'édition par ECoMaS nécessite au départ l'importation des transparents qui sont des images, puis, l'enseignant peut enregistrer ses explications orales concernant chacune d'elles (voir Figure 11 à gauche). Ensuite, le système génère une présentation publiable sur le web (SMIL2.0), où les transparents sont synchronisés avec leur explication auditive et la table d'index qui offre la navigation temporelle durant la présentation du cours (Figure 11 à droite).

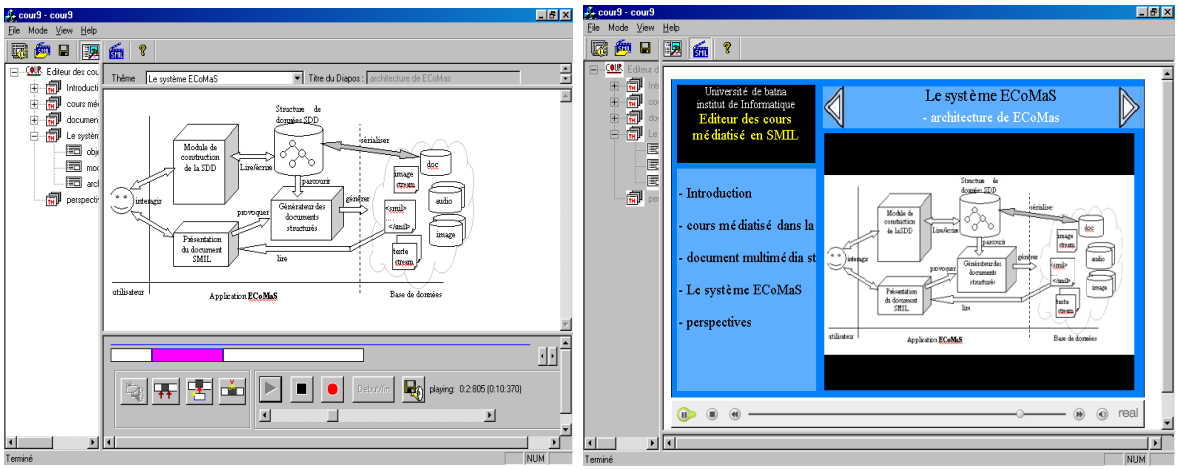


Figure 11. *Interface de l'éditeur ECoMaS en mode édition et en mode présentation*

2.2.6 Discussion

Les trois derniers éditeurs utilisent des modèles de document (ou gabarits), organisés de manière hiérarchique. Chaque modèle est vu comme un document à trous, où l'utilisateur ne fait que remplir les trous par des médias (texte, image, audio ou vidéo). On voit bien que ces médias sont importés de l'extérieur et donc l'enseignant doit les préparer à l'avance et chacun avec l'outil qui lui correspond. Ceci est fastidieux pour l'enseignant et surtout s'il veut éditer un texte formaté (avec des couleurs et des styles…) et ensuite lui associer sa prononciation car il devra alors utiliser deux outils différents (un pour le traitement de texte et l'autre pour le son), les importer au système et ensuite les synchroniser.

Une limitation gênante de ces outils est l'absence d'un service graphique qui permet d'éditer du texte formaté. Les versions de SMIL 1.0 et 2.0 utilisés par ces outils ne supporte pas de balises pour le formatage de texte (couleur, style, font,…), ce qui est très important pour concrétiser l'approche de double codage cité dans notre problématique. Le système SWANS utilise le standard XHTML+SMIL. L'éditeur ECoMaS utilise le langage RealText juste pour présenter le titre d'une diapositive et le contenu de la table d'index. La dernière version de SMIL est SMIL3.0 (SMIL 3.0) et elle supporte des fonctionnalités de formatage de texte intra-document (.smil), mais jusqu'à présent, il n'y a pas d'éditeur graphique pour cette version. La première partie de notre travail est une contribution pour à ce manquement et donc au développement d'un éditeur de SMIL3.0. La phonétique essaie de représenter les sons d'une façon plus précise en utilisant l'alphabet phonétique international (API). L'enseignant ne peut pas utiliser ce type de caractères avec les éditeurs existants.

2.3 Approches d'indexation

S'il est important de savoir modéliser et créer des documents multimédia, il est également important de permettre aux apprenants de localiser rapidement une information recherchée dans ce type de documents. Ceci ne se fait qu'à travers des systèmes de recherche d'information, qui nécessite dans une première phase l'indexation de la base documentaires. Nous présentons en premier lieu l'indexation classique et sa position dans un système de recherche d'information (SRI).

2.3.1 Indexation classique

La Recherche d'Information (RI) est un domaine de recherche en informatique qui s'attache à définir des modèles et des systèmes dont le but est de faciliter l'accès à un ensemble de documents sous forme électronique (corpus), afin de permettre à un utilisateur de retrouver ceux qui sont pertinents pour lui, c'est-à-dire ceux dont le contenu correspond le mieux à son besoin d'information à un moment donné. La Figure 12 décrit l'architecture générale d'un SRI; elle présente les éléments principaux mis en jeu dans ce système.

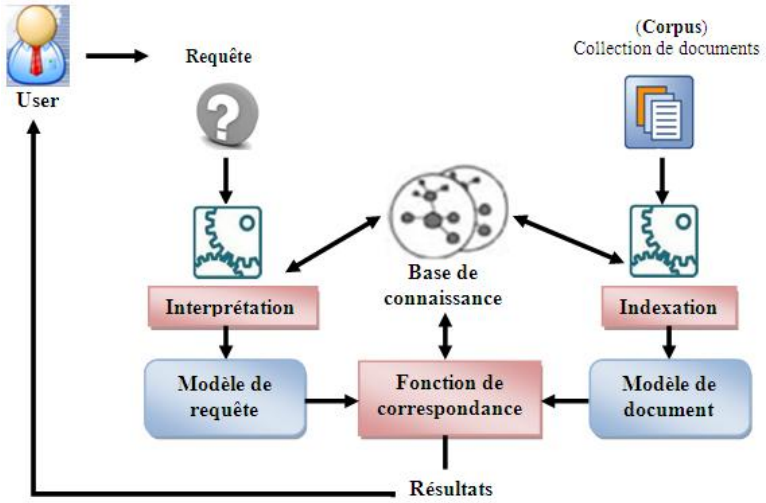


Figure 12. *Architecture générale d'un SRI*

Martin (Martin, 2004) a défini les différents composants d'un SRI comme suit :

1. **L'indexation** consiste à extraire et à représenter sous la forme d'un index électronique directement manipulable par le système le contenu des documents. La norme AFNOR NF Z 47-102 1996 définit l'indexation comme une opération qui consiste à décrire et à caractériser un document à l'aide de représentations des concepts contenus dans ce document, c'est-à-dire à transcrire en langage documentaire les concepts descripteurs après les avoir extraits du document par une analyse. Ainsi, le choix des descripteurs de documents détermine la qualité de la représentation. Les approches classiques de RI textuelle sont basées sur les mots clés comme descripteurs du contenu des documents; le contenu d'un document textuel est ainsi exprimé sous la forme d'un ensemble de mots clés jugés représentatifs de ce contenu. Ces mots clés, qui constituent le vocabulaire d'indexation, sont un sous-ensemble des mots apparaissant dans les documents. En effet, seuls les mots porteurs de sens sont habituellement retenus dans le vocabulaire d'indexation : les mots outils de la langue comme "de", "le", "par", qui sont recensés dans une base appelée anti-dictionnaire, ne sont pas conservés

dans le vocabulaire d'indexation. Par ailleurs, il est courant de "raciniser" (ou lemmatiser) les mots, c'est-à-dire de détecter les variantes orthographiques, comme les pluriels, ou les conjugaisons des verbes, et de ramener les variantes orthographiques d'un même mot à une racine commune.

2. **L'interrogation** est l'interaction d'un utilisateur final avec le SRI, une fois le contenu des documents représenté de manière interne sous forme d'index. L'utilisateur exprime son besoin d'information sous la forme d'une requête, qui est interprétée selon le modèle de requête, et le système évalue la pertinence des documents par rapport à cette requête par l'intermédiaire de la fonction de correspondance (et éventuellement d'une base de connaissances). La réponse à ce besoin est la liste des documents qui obtiennent une valeur de correspondance élevée. Cette liste est généralement triée par ordre de valeur de correspondance décroissante – c'est-à-dire du plus pertinent au moins pertinent – et ensuite présentée à l'utilisateur.

3. **Le modèle de documents** (aussi appelé langage d'indexation) exprime le contenu sémantique des documents dans un formalisme de représentation des connaissances (qui peut être très simple comme celui des mots clés, ou beaucoup plus complexe à partir de formalismes de représentation de connaissances 'ontologie'). Le choix de ce formalisme est fondamental car il détermine la qualité de la représentation interne des documents, et donc la qualité de la recherche et des résultats. Cette modélisation doit offrir un compromis entre deux besoins non compatibles, qui sont la capacité de la représentation pour qu'elle puisse être traitée efficacement par un système informatique, et l'expressivité du formalisme pour qu'elle soit précise et exhaustive, afin d'exprimer aussi fidèlement que possible le contenu des documents.

4. **Le modèle des requêtes** exprime le contenu sémantique du besoin d'information de l'utilisateur. Ce formalisme détermine la précision de la définition du besoin. Dans la plupart des approches existantes pour le texte, le modèle de requêtes utilise le même formalisme de représentation des connaissances que le modèle de documents pour des raisons de simplicité et de cohérence.

5. **Le modèle de correspondance** entre une requête et un document formalise le degré de similarité entre les modèles de la requête et du document. Comme dit plus haut, la fonction de correspondance évalue cette similarité afin de déterminer la pertinence des documents pour cette requête.

6. **Une représentation des connaissances du domaine** considéré, afin de satisfaire aux contraintes des modèles de documents et de requêtes, et de prendre en compte des connaissances externes. Cette représentation des connaissances peut

inclure un thésaurus composé des termes apparaissant dans l'ensemble des documents, reliés entre eux par des liens de généricité/spécificité ou de synonymie. Comme, elle peut inclure une ou plusieurs ontologies qu'on va les voir en détail dans les partie suivantes.

Parmi les différents modèles de recherche d'informations dits classiques, figurent les modèles booléen, vectoriel, logique et probabiliste. Nous présentons ci-dessous le modèle vectoriel, qui est à la base de nos travaux, en le décrivant selon les modèles de documents, de requêtes et de correspondance.

2.3.1.1 Modèle vectoriel

Le fondateur de ce modèle est (Salton, 1971). Le nom de ce modèle est inspiré de son principe qui représente les documents et les requêtes par des vecteurs d'un espace à n dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation. L'index d'un document dj est le vecteur $\vec{dj} = (w_{1,j}, w_{2,j}, \ldots\ldots, w_{n,j})$, où $w_{i,j} \in [0,1]$ dénote le poids du terme ti dans le document dj (voir dans ce qui suit sa formule de calcul). Le poids d'un terme dénote son intérêt dans le document. Une requête est également représentée par un vecteur $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$, où $w_{i,j} \in [0,1]$ est le poids du terme ti dans la requête.

La figure 13 montre un exemple d'espace vectoriel composé des trois termes t_1, t_2 et t_3 ainsi que les index de deux documents D_1 et D_2 et une requête Q.

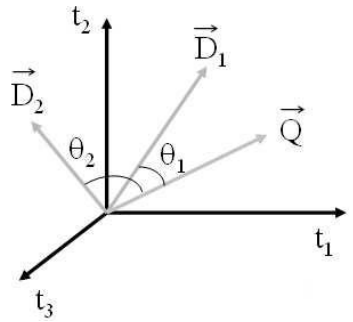


Figure 13. *Représentation vectorielle de deux documents (D1 et D2) et d'une requête (Q) dans un espace composé de trois termes (t1, t2 et t3).*

La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Une mesure classique est le cosinus de l'angle formé des vecteurs:

$$\text{Correspondance}(dj, q) = \cos(\vec{dj}, \vec{q}) \qquad (1)$$

Où cos ($\vec{d}$j,$\vec{q}$) est le cosinus de l'angle formé par les vecteurs $\vec{d}$j et $\vec{q}$:

$$\cos(\vec{d}j,\vec{q}) = \frac{\vec{d}j \cdot \vec{q}}{||\vec{d}j|| \cdot ||\vec{q}||} = \frac{\sum_{i=1}^{n} wi,j \cdot wi,q}{\sqrt{\sum_{i=1}^{n} w^2 i,j} \cdot \sqrt{\sum_{i=n}^{n} w^2 i,q}}$$

Plus deux vecteurs sont similaires, plus l'angle formé est petit, et plus le cosinus de cet angle est grand.

Le poids des termes Wi,j : Le poids d'un terme dénote son intérêt dans l'index, qui dépend du fait que le terme est important dans le document, et du fait que le terme permet de distinguer un document des autres dans la base documentaire. Les méthodes de pondération les plus largement utilisées pour le texte sont basées sur des variantes de la formule ***tf.idf*** (ou ***term frequency. inverted document frequency***).

$$W_{i,j} = tf.idf = tf.\ Log\ (\frac{N}{ni})$$

La valeur ***tf*** est définie en fonction du nombre d'occurrences du terme dans le document, N est le nombre total de documents dans le corpus et n_i est le nombre de documents dans lesquels le terme t_i apparaît.

La valeur ***tf*** mesure l'importance d'un terme dans un document. La valeur ***idf*** est une mesure de la discrimination du terme. Plus précisément, la notion de discriminant renvoie à la qualité d'un terme qui distingue bien un document des autres. Un terme qui apparaît seulement dans un petit nombre de documents permet bien de discriminer les documents. Au contraire, un terme qui apparaît dans un grand nombre, voire dans tous les documents d'une collection n'est pas discriminant en raison de sa distribution uniforme.

Notons que le contenu que nous devons indexer sémantiquement est multimédia et ainsi il se différencie d'un simple texte. La section suivante présentera les techniques utilisées pour indexer un tel type de contenu.

2.3.2 Vers l'indexation sémantique du contenu multimédia

L'indexation des documents multimédia nécessite l'intégration d'une chaîne plus complète de traitement du contenu dans le processus de production d'un document multimédia. Jedidi (Jedidi, 2005) a précisé les trois étapes d'une chaîne complète d'une application multimédia à savoir : analyse, description et application/traitement. En

entrée de cette chaîne, les médias (image, son, vidéo …) sont analysés pour pouvoir extraire automatiquement et/ou manuellement des informations pertinentes (ou caractéristiques), puis ces informations sont représentées sous un format prédéfini pour pouvoir être largement et efficacement utilisées dans des applications et des traitements de média. Dans le cas d'une application de type SRI Multimédia, on peut constater que la phase d'analyse précède l'étape d'indexation proprement dite. Les méthodes d'analyse sont classifiées en 3 catégories selon le niveau :

- Un niveau bas qui décrit les caractéristiques simples des éléments d'un document comme les couleurs, la texture d'une image ou l'enveloppe d'un son;
- Un niveau sémantique qui fournit une description de haut niveau de ce que contient la vidéo, qu'il s'agisse de personnages, d'objets, de lieux ou de leurs interactions;
- Un niveau structurel qui met en évidence une organisation du document.

Les techniques et approches d'analyse de bas niveau dépassent le sujet de cette thèse. Donc, on s'intéresse à l'analyse sémantique du contenu multimédia. Rappelons que les documents traités dans cette thèse, sont des documents multimédia supportant la dimension temporelle. Donc, on peut les classifiés comme des séquences vidéo multimédia. De ce fait, on va spécifier ci-après les études sur l'analyse et l'indexation par la sémantique de ce type de documents.

On doit noter qu'il est pratiquement impossible d'atteindre le niveau sémantique en partant d'une analyse bas niveau du contenu vidéo. Les interprétations du contenu d'un document vidéo, qui sont sémantiquement plus riche, rend la tâche de l'indexeur plus compliquée que lorsqu'il s'agit d'une simple indexation par mots clés. Ceci est dû au fait qu'il doit choisir les meilleurs index pour décrire un contenu très riche en informations. On rencontre la même difficulté lors du processus de recherche. C'est pour cela que des modèles capables de décrire et de modéliser le contenu sémantique de séquences vidéo sont mis en place, afin de faciliter l'accès, la réutilisation et la navigation par la sémantique. Ces modèles utilisent des techniques à base de connaissances. Le web étant constitué de ressources multimédia, la technique de leur indexation généralement employée consiste à annoter des portions de ces ressources

avec des informations exploitables par des agents logiciels. Nous allons donc définir dans la section suivante ce que recouvre la notion d'annotation.

2.3.2.1 Annotation sémantique

Une annotation, en sens générique, regroupe aussi bien l'ajout d'informations sans contraintes particulières, comme un échange de courriel à propos d'une vidéo, ou l'ajout d'informations (caractérisant le contenu) qui doit respecter un format bien défini. Dans ce deuxième cas, le terme **métadonnées** est plus approprié car il est plus spécifique. C'est ce dernier type qui nous intéresse dans ce travail. Plusieurs standards permettent de décrire et d'annoter des contenus multimédias en utilisant une liste définie d'attributs comme la date de création, les auteurs, l'encodage utilisé par la vidéo, la résolution de l'image ou la bande passante exigée pour sa transmission par réseaux.

Comme les métadonnées sont des informations à propos des données qui composent le document, il est important de pouvoir les rattacher au document ou de créer un lien entre eux. Jedidi (Jedidi, 2005) a spécifié quatre façons d'effectuer un rattachement entre un document et les métadonnées qui y sont associées (voir figure 14):

1. par insertion direct dans le document : les métadonnées sont contenues dans le document,
2. par accompagnement : les métadonnées sont externes, mais véhiculées avec le document et référencées avec leur adresse URL,
3. par lien Href au document : les métadonnées sont externes et pouvant être véhiculées séparément du document, et ils sont référencées avec un URL spécifique,
4. par enveloppement : l'ensemble des métadonnées couvre la ressource.

Cette classification de ces différents modes de rattachement de métadonnées aux documents amène Jedidi à distinguer deux types majeurs de métadonnées : les métadonnées externes et les métadonnées internes. Pour plus de détail voir (Jedidi, 2005).

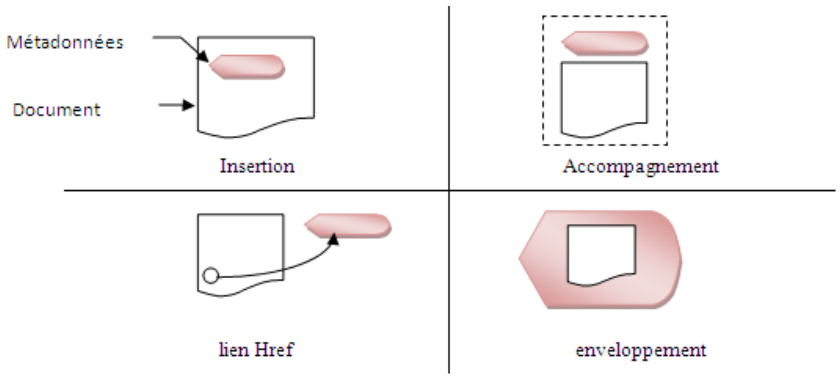


Figure 14. *Les quatre modes de rattachement des métadonnées au document*

L'annotation vidéo est le processus par lequel des informations textuelles ou autres sont associées ou jointes à des segments déterminés dans le document vidéo pour enrichir le contenu. Ces informations ne modifient pas le document mais elles sont simplement mises en correspondance avec celui-ci.

Dans le cas où l'annotation se fait sur des documents audiovisuel, Troncy le considère comme un processus qui se compose de quatre étapes (Troncy, 2004):

1- Identification : Il s'agit de donner des informations de catalogage sur le document (titre, auteur, format, droit,…), utilisant un ensemble de descripteurs normalisés comme celles du Dublin Core par exemple.
2- Segmentation : il s'agit de repérer dans le document des entités spatio-temporelles d'**intérêt** qui seront annotées utilisant des coordonnées cartésiennes et des dates pour localiser dans l'espace et dans le temps des segments pertinents.
3- Caractérisation ou annotation: il s'agit de donner une signification formelle aux segments préalablement repérés utilisant des descripteurs contrôlés et ayant une sémantique formelle pour la machine.

2.3.2.2 Standards et outils d'annotation

Dublin Core (DCMI, 2010) pour l'audiovisuel : le Dublin Core permet une description du document à un niveau bibliographique. Il ne spécifie pas de syntaxe particulière, mais il fournit un ensemble d'éléments terminologiques pour la description. Il peut être exprimé en XML ou en RDF. Quinze descripteurs de haut niveau sont utilisables pour décrire des informations qui sont de nature bibliographique (Nom, identifiant, version, langage, contributeurs, type, définition, commentaires…). Les quinze éléments et qualificateurs de Dublin Core peuvent être utilisés pour décrire à la fois la structure et le détail des vidéos. Ceci vient du fait qu'il peut être utilisé conjointement à un autre

formalisme de description, comme MPEG-7 (voir juste après), afin de fournir un ensemble efficace de métadonnées. Cette combinaison a l'avantage de permettre l'association de descripteurs standards aux différents niveaux de décomposition du flux audiovisuel, qu'ils portent sur l'ensemble du document, sur une séquence, une scène, un plan, une image ou une partie d'une image. Dans cette approche, le document audiovisuel est décomposé de façon statique selon un arbre annoté par des métadonnées.

Le standard MPEG-7 (Motion Picture Engineer Group) MPEG-7 supporte un ensemble de descriptions allant des caractéristiques de bas niveau (forme, taille, texture, couleur, mouvement, position…) à des caractéristiques de plus haut niveau (auteur, date de création, format, objets et personnages ainsi que leurs relations, contraintes spatiales et temporelles…) (MPEG-7, 2002). Les métadonnées qui apparaîtront dans un document MPEG-7 sont de plusieurs natures différentes, résumées dans les points suivants :

- création et production : des métadonnées qui décrivent la création et la production du contenu. Elles décrivent le titre, l'auteur, le but de la création.
- utilisation : des métadonnées reliées à l'utilisation du contenu. Elles comportent les droits d'accès, les informations financières et les droits de publication.
- média : ces métadonnées décrivent les caractéristiques de stockage telles que le format, le type de compression, etc.
- aspects structurels : des descriptions d'un point de vue contenu. Ces métadonnées décrivent les segments qui peuvent représenter des composantes spatiales, temporelles ou spatio-temporelles du contenu audiovisuel. Chaque segment peut être décrit par les caractéristiques suivantes (la couleur, la texture, la forme, la motion, durée, taux d'échantillonnage, taille...) et quelques informations sémantiques élémentaires.

RDF et Métadata dans les documents SMIL : le domaine ou module *Méta Information* de la version 2.9 de SMIL contient les éléments et les attributs qui permettent la description (globale et détaillée) de documents SMIL. Ce module gère pleinement l'utilisation de l'élément <meta> de SMIL 1.0, mais il introduit aussi de nouvelles fonctionnalités pour décrire des objets média en utilisant la syntaxe et le modèle du cadre de description de ressources (Ressource Description Framework RDF). Ce dernier est un langage puissant de méta-information pour fournir des informations sur des ressources, voir (RDF, 1999).

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
……
  <metadata id="meta-rdf">
   <rdf:RDF
     xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
     xmlns:dc = "http://purl.org/metadata/dublin_core#"
     xmlns:smilmetadata = "http://www.example.org/AudioVideo/.../smil-
ns#" >
<!—description  de la présentation SMIL en entier -->
   <rdf:Description about="http://localhost/cours-system.smil"
     dc:Title=" Cours du système d'exploitation"
     dc:Description="contient les chapitres suivant : l'exclusion
mutuelle, l'inter blocage, systèmes réparties"
     dc:Date="2000-10-12"
     dc:Format="text/smil" >
     dc:Creator= "Creer par Dr. Ben Abbass"
     <smilmetadata:ListOfAudioUsed>
   <rdf:Seq ID="les mécanismes de l'exclusion mutuel">
      <rdf:li Resource=" http://localhost/voix1.rm "/>
      <rdf:li Resource=" http://localhost/voix2.rm "/>
   </rdf:Seq>
     </smilmetadata:ListOfVideoUsed>
   </rdf:Description>
<!—Description de l'objet audio voix1.rm -->
   <rdf:Description about=" http://localhost/voix1.rm"
dc:Description=" les verrous, test and set, les sémaphores, les
estampilles " dc:Format="audio/rm"  smilmetadata:Duration="50 mn">
<smilmetadata:ContainsSequences>
  <rdf:Seq ID="différents mécanismes">
     <rdf:li Resource=" http://localhost/ voix1.rm#sémaphore/>
     <rdf:li Resource=" http://localhost/ voix1.rm#verrous/>
  </rdf:Seq>
</smilmetadata:ContainsSequences>
   </rdf:Description>
<!—Description d'un segment temporel de l'objet audio voix1.rm-->
   <rdf:Description about="#sémaphore" dc:Title="sémaphores"
     dc:Description="définitions des fonctions ininterruptibles
     P(s) et V(s)+ algorithmes"  smilmetadata:Duration="10 mn"
  </rdf:Description>
  </rdf:RDF>
 </metadata>
```

Le code illustré ci-dessus montre l'utilisation de l'élément **métadata** ainsi que la norme RDF et les attributs de Dublin Core dans la description du document smil en entier. Ce code décrit aussi un objet audio contenu dans ce document. Ensuite il décrit un segment temporel dans ce dernier objet média en déterminant leur durée.

Plusieurs outils existent permettant l'annotation des vidéos. Ces outils permettent le découpage semi-automatique de la vidéo en plans et l'affectation manuelle des mots clés aux plans ainsi délimités. Il s'agit d'interpréter des caractéristiques textuelles (mots clés) des documents audiovisuels et de les organiser dans des bases de données dans lesquelles la recherche va s'effectuer. Charhad a présenté quelques outils d'annotation vidéo, Nous citons quelque un dans ce qui suit :

- l'outil **Segmen-Tool** pour découper temporellement les vidéos et produire un début de description MPEG-7.

- l'outil **Video-Annex** (Lin, 2003): l'annotation avec cet outil peut s'appliquer à la fois sur le document complet, et sur des parties du document (segment vidéo ou image clé d'un plan). L'annotation, qui porte sur le document en entier ou sur des segments, est effectuée à l'aide de possibilité de champs d'annotation libre. Ce champ permet aussi d'utiliser d'autres concepts qui ne figurent pas forcément dans la liste.

- **Smart VideoText** est un système d'annotation vidéo basé sur le formalisme des graphes. Dans ce système, les portions vidéo représentent les nœuds du graphe. Ces portions sont identifiées par des références liées à la structure physique du document (identifiant du plan, numéro de l'image dans le plan, etc.) et aussi par les annotations libres.

- **COALA – Log Creator –EPFL** : l'outil COALA (content Oriented Audiovisuel Library) conduit par l'EPFL en Suisse a débouché sur la réalisation d'une plate-forme prototype d'indexation et d'annotations des journaux télévisés de la TSR (Télévision Suisse Romande). Contrairement aux autres outils, il se présente comme une application du Web spécialisé dans l'annotation d'un genre particulier de document vidéo.

La recherche de segments vidéo particulières s'avère parfois difficile, notamment si elle est effectuée par des personnes non-documentalistes. Le raisonnement est typiquement une manipulation qui permet, par exemple, de mieux satisfaire les requêtes lors de l'interrogation de la base des descriptions. Cependant, le type de langage utilisé (documentaire), qui restreint les inférences à la seule validation de structure, et l'emploi du texte libre ou de thésaurus pour décrire le contenu, qui empêche de véritablement contrôler la sémantique des descriptions, limitent sérieusement les possibilités de raisonnement.

Dans le domaine du Web sémantique, où des annotations sont principalement définies pour être interprétées par des machines, des annotations sémantiques sont décrites comme des annotations qui se rapportent à une **ontologie**. Cette dernière identifie formellement des concepts et des relations entre les concepts dans les documents. IL faut noter que les annotations sémantiques utilisant une ontologie ont comme objectif majeur de désambiguïser le document pour un traitement automatique. Puisque on s'intéresse, dans notre travail, à indexer des séquences vidéo pédagogiques en utilisant

des ontologies codé en OWL, on doit d'abord mettre en lumière cette technique ainsi que les notions qui l'entoure.

2.3.3 Indexation à base d'ontologies

Dans cette section, nous décrivons, en premier lieu, les principes de base des ontologies OWL (OWL, 2004), du raisonnement avec OWL et des règles SWRL (SWRL, 2004), ainsi que les possibilités qu'ils offrent. En deuxième lieu nous présentons la manière d'intégration et d'utilisation des ontologies dans les systèmes de recherche d'information et plus précisément dans la phase d'indexation.

2.3.3.1. Les ontologies OWL

Les ontologies sont un moyen de représenter la connaissance. Ces représentations de connaissances correspondent à « *une spécification formelle explicite d'une conceptualisation partagée*». C'est la définition la plus simple et fait le plus autorité et elle est due à Thomas Gruber (Gruber, 1993). Les termes "formelle" et "explicite" signifient qu'une ontologie permet une interprétation automatisée de la conceptualisation par la machine. Autrement dit, une ontologie définit un vocabulaire commun pour les chercheurs qui ont besoin de partager l'information dans un domaine. Les ontologies peuvent être utilisées par des personnes, des bases de données et des applications qui ont besoin de partager des informations sur un domaine.

Nous présentons la définition des éléments principaux que l'on trouve dans une ontologie OWL :

- (**Individu/instance**) Les individus ou instances sont les objets du domaine de discours que l'on représente dans une ontologie.
- (**Concept/classe, sous-classe, super-classe, taxonomie**) Un concept ou classe regroupe un ensemble d'individus qui ont des caractéristiques communes. Une classe peut être sous-classe d'une autre, appelée super-classe : dans ce cas, tout individu appartenant à la sous-classe appartient aussi à la super-classe. Une taxonomie est une hiérarchie de classes qui ont des relations sous-classe (relation de spécialisation)/super-classe (généralisation du sens) entre elles.
- (**Relation ou propriété, domaine, portée, sous-relation, super-relation**) Une relation ou propriété modélise le lien qui existe entre deux classes (appelé aussi relation objet) ou entre une classe et un type de données (relation type de données). Le domaine d'une relation représente l'origine de ceci. La portée de la relation représente sa destination. Une relation peut être une sous-relation d'une

autre, appelée super-relation. Dans ce cas, le domaine et la portée de la sous-relation sont contenus respectivement dans le domaine et dans la portée de la super-relation.

- (**Instance d'une relation**) Une instance d'une relation relie un individu qui appartient au domaine de la relation à un individu ou à un type de données qui appartient à la portée de la relation.

Les définitions suivantes détaillent les attributs des propriétés.

- (**Propriété inverse**) On peut définir une propriété comme l'inverse d'une propriété donnée. Cela veut dire que, si une instance de cette dernière propriété relie l'individu (a) à l'individu (b), alors on peut déduire qu'une instance de la propriété inverse relie (b) à (a).
- (**Propriété fonctionnelle, inverse fonctionnel**) Une propriété est fonctionnelle quand elle ne peut avoir qu'une seule instance pour un individu donné. L'inverse d'une propriété fonctionnelle est son inverse fonctionnel.
- (**Propriété transitive**) Une propriété est transitive lorsque, si l'individu (a) est relié à (b) par une instance de cette propriété et que (b) est relié à (c) par une autre instance de la propriété, alors on peut déduire que (a) est relié à (c) par une instance de la propriété.
- (**Propriété symétrique**) Une propriété est symétrique quand pour tout (a) relié à (b) par une instance de cette propriété, on peut déduire que b est relié à a par une autre instance de la propriété.

Pour définir une classe dans OWL, on fournit un ensemble de conditions logiques. Ces conditions peuvent être nécessaires ou nécessaires et suffisantes. Elles sont construites à partir d'autres classes, par union, par intersection ou par héritage. On peut également imposer des restrictions aux propriétés de la classe.

- (**Restriction, restriction existentielle, universelle, de cardinalité, de valeur**) Une restriction consiste à limiter le nombre ou la nature des valeurs que peuvent avoir les propriétés des individus d'une classe. Une restriction peut être existentielle (si elle oblige à avoir au moins une valeur de la propriété dans un ensemble donné), universelle (si elle oblige à avoir toutes les valeurs d'une propriété dans un ensemble donné), de cardinalité (si elle oblige à avoir un nombre de valeurs minimal, maximal ou exact pour une propriété) ou de valeur (si elle oblige à avoir une valeur donnée pour la propriété).
- (**Classes disjointes**) Deux classes sont dites disjointes quand il ne peut exister des individus qui appartiennent à la fois aux deux classes. Dans OWL, les classes ne sont pas disjointes par défaut; il faut le déclarer explicitement.

Les ontologies OWL sont extensibles grâce au mécanisme d'importation :

- (**Importation d'ontologie**) Une ontologie peut importer une autre ontologie pour avoir une visibilité sur ses éléments. L'ontologie qui importe a un accès en lecture seule à tous les éléments contenus dans l'ontologie importée. Elle peut ensuite ajouter de nouveaux éléments, qui ne seront visibles que dans l'ontologie qui importe. L'importation est transitive : une ontologie qui importe une deuxième importe indirectement toutes les ontologies importées dans cette deuxième ontologie. En général, l'importation se fait en indiquant, dans l'ontologie qui importe, l'URL où se trouve l'ontologie importée. Ce mécanisme permet une grande flexibilité, car on peut réutiliser des ontologies existantes juste en les référençant dans un nouveau fichier OWL et en ajoutant de nouvelles classes, propriétés, individus, règles, etc. Souvent, on trouve des ontologies qui suivent ce schéma : une première ontologie de haut niveau qui contient des éléments génériques est importée par une deuxième ontologie de domaine qui les spécialise pour un domaine concret. Dans ce cas, on dit que la deuxième ontologie étend la première.

Raisonnement : Une ontologie qui possède une base théorique formelle (la logique de description) permet l'implantation de logiciels appelés moteurs d'inférence ou raisonneurs, qui sont capables de traiter une ontologie pour déduire des faits qui ne sont pas explicitement déclarés C'est-à-dire, ils peuvent trouver des informations qui sont implicitement contenues dans l'ontologie pour les rendre explicites. Ce processus s'appelle inférence ou raisonnement.

La principale tâche du raisonnement qu'un moteur d'inférence peut effectuer est connue comme la subsomption. La subsomption permet de savoir si une classe est une sous-classe d'une autre ou non. En effectuant cette tache sur toutes les classes d'une ontologie, le moteur d'inférence peut construire une hiérarchie de classes inférée (en opposition à la hiérarchie de classes déclarée) dans laquelle toutes les relations super-classe/sous-classe sont explicitées.

D'autres tâches de raisonnement permettent de déduire des faits tels que l'appartenance d'un individu à une classe en utilisant les liens d'héritage, l'existence de propriétés liant deux individus à partir de la transitivité, de la symétrie et des propriétés inverses, etc.

Un raisonnement peut utiliser **des règles**. Il existe des conditions que les ontologies OWL ne peuvent pas exprimer, comme les Règles de Horn (Horn, 1951), d'où le W3C a proposé un langage appelé **SWRL** (SWRL, 2004) qui combine OWL-DL avec le Rule

Markup Langage (RuleML). SWRL étend OWL-DL en ajoutant des clauses de Horn. Une règle SWRL présente la forme suivante : $b_1 \wedge \cdots \wedge b_n \rightarrow a_1 \wedge \cdots \wedge a_n$.

Où $b_1 \wedge \cdots \wedge b_n$ est le corps ou antécédent de la règle et $a_1 \wedge \cdots \wedge a_n$ est l'entête ou conséquent. Les termes $a_1,...,a_n,b_1,...,b_n$ sont des atomes SWRL. Un atome peut représenter une relation (prédicat binaire), un concept (prédicat unaire) ou un built-in (prédicats n-aires). Il existe d'autres syntaxes, notamment une syntaxe XML (voir annexe 4) basée sur celle d'OWL, ce qui permet d'inclure les règles SWRL associées à une ontologie dans le même fichier XML que l'ontologie. L'utilité de ces règles est d'exprimer des relations qui seraient trop compliquées, voire impossibles, à exprimer avec OWL-DL seulement. Par exemple, dans notre ontologie on peut représenter la relation entre un oncle et son neveu à partir des relations père–fils et frère–frère. La règle SWRL qui exprime cette relation est représentée dans l'exemple suivant :

Personne(?x) ^ Personne(?y) ^ Personne(?z)^ père(?x,?y) ^ frère(?x,?z) → *oncle(?z,?y)*

Les éléments x, y et z, qui sont précédés d'un point d'interrogation, sont des variables qui représentent des individus. Cette règle veut dire que, si l'on a trois individus appartenant à la classe Personne appelés x, y et z, que x est le père de y et que x a un frère z, alors z est l'oncle de y.

Une ontologie peut être construite d'une façon manuelle, automatique ou mixte. Le développement général d'une ontologie se déroule en trois principales étapes qui sont : la conceptualisation, la formalisation et l'implémentation ou opérationnalisation exploitable par des agents logiciels (spécifier les concepts et relation dans un langage formelle). Ces étapes sont généralement précédées par une étape d'évaluation des besoins et de délimitation du domaine de connaissances à modéliser.

Il existe plusieurs langages de spécification d'ontologie Web tel que RDFS, DAML-ONT, OIL, DAML+OIL, **OWL** +RDFS, OWL 2, etc. Comme il existe des outils qui aident à la construction des ontologies (on peut citer l'exemple de TERMINAE). En revanche d'autres outils d'aide à la construction d'ontologies sont de type éditeurs comme DOE (Differential Ontology Editor), ODE (Ontology Design Environment 2002), OntoEdit (Ontology Editor) et l'éditeur **Protégé 2000**. Pour ce qui concerne le traitement des règles SWRL, la plupart des moteurs d'inférence sont capables de traiter les règles

SWRL ajoutées à une ontologie. Par exemple, Pellet implante de façon native un algorithme spécifique pour des règles DL-sûres dans OWL. Afin de traiter les règles, il est également possible d'utiliser des moteurs spécifiquement dédiés aux règles, tel que le moteur Jess[7]. Ce moteur de règles possède un langage propre pour l'expression des connaissances sous forme de règles. Il peut être utilisé depuis Protégé grâce à l'existence d'un pont qui permet de traduire un modèle d'ontologie dans le langage de Jess, d'exécuter les règles dans Jess et finalement de récupérer le résultat dans Protégé.

2.3.3.2. Utiliser l'ontologie dans les SRI

Dans un SRI, l'utilisation de l'ontologie peut être introduite dans les trois phases comme illustré dans la figure 12 (ontologie est représentée par base de connaissance):

- Au niveau du processus d'indexation, l'ontologie peut aider à l'indexation des documents, alors appelée indexation sémantique.
- Les ontologies peuvent également aider à la reformulation du besoin de l'utilisateur pour améliorer les requêtes utilisateurs et à l'accès aux documents.
- Enfin l'ontologie peut être utilisée dans le modèle lui-même pour réaliser l'appariement entre le besoin et les documents. Pour plus de détail sur ces trois points voir (Hernandez, 2006).

Hernandez a défini deux étapes pour faire une indexation sémantique.

1- La première étape consiste à identifier les concepts ou instances de l'ontologie dans les granules documentaires appelées aussi ***annotation conceptuelle***.
2- La deuxième étape consiste à ***pondérer les concepts*** pour chaque document en fonction de la structure conceptuelle dont il est issu.

Les ontologies doivent avoir une large composante lexicale, pour être utilisées dans le cadre de la RI, et ceci afin que ces concepts peut recouvrir la majorité des notions qui puissent être retrouvés dans les documents. Nous énumérons deux principaux gains attendus par l'utilisation des ontologies dans un SRI qu'on peut résumer comme réduire le silence et le bruit:

- réduire le silence dans les réponses aux requêtes, en traitant le problème de la synonymie,

[7] Voir le lien http://www.jessrules.com/

- réduire le nombre de réponses bruitées, en traitant le problème d'homonymie.

La recherche d'information (principalement sur le web), peut même être considérée comme un des champs d'applications favoris des ontologies. Par exemple, elles sont souvent présentées comme une pièce maîtresse dans le domaine du web sémantique. Dans la section suivante, on va étudier quelques travaux qui utilisent les ontologies dans la recherche d'information de type pédagogique et/ou séquence vidéo.

2.3.4 Travaux existants :

L'utilisation d'ontologie dans un contexte d'indexation s'est développée ces dernières années dans différents domaines, on en cite deux qui nous intéressent : le domaine audiovisuel (Troncy, 2004), (Charhad, 2005), (Isaac, 2005), (Carbonaro, 2008) et le domaine de l'e-learning (Dong, 2010), (Desmoulins, 2000), (Benayache, 2005), (Buffa, 2005). Dans ce dernier, plusieurs travaux sont basés sur l'idée générale d'indexer des fragments de documents sur la base d'ontologies de nature différentes, ontologie structure documentaire, ontologie de domaine ou ontologie pédagogique (figure, formule, équation,…), en vue de les réutiliser pour composer plus ou moins automatiquement de nouvelles ressources.

Par exemple le projet IMAT (Desmoulins, 2000) est tourné vers l'utilisation d'ontologie pour l'indexation. La manipulation de documents nécessite l'emploi d'une ontologie dite 'document' qui vient s'ajouter aux ontologies classiques de domaine et de pédagogie. L'ontologie de cours se décompose en trois sous-parties décrivant le contenu, le contexte et la structure. Le contenu représente l'ontologie de domaine et les deux autres parties sont liées à l'aspect pédagogique (structuration au chapitres, nature des parties, etc.).

Le projet MEMORAe (Benayache, 2005) a pour objectif de rendre les ressources d'une formation plus accessibles aux apprenants via des ontologies. Ce projet propose une approche fondée sur une mémoire organisationnelle de l'ensemble des notions à appréhender d'une formation et des ressources de celle-ci. Son principe est d'indexer ces ressources sur les concepts d'ontologie. Deux ontologies ont été construites et intégrés. La première est une ontologie de domaine qui décrit les concepts de la formation : personne (étudiant, enseignant,..), documents (livre, page web, …), activité

pédagogique (cours, TP,...). La deuxième est une ontologie d'application qui spécifie l'ensemble des concepts utiles à une formation particulière tels que l'algorithmique ou la statistique (voir figure 15).

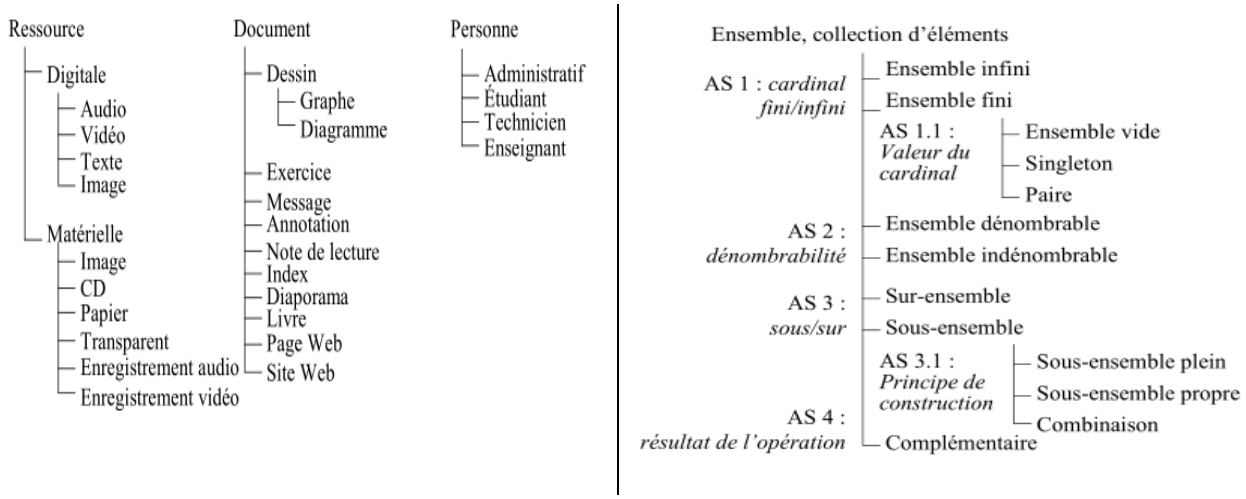


Figure 15. *Quelques éléments de l'ontologie du domaine de formation (à gauche), et un extrait de la sous-ontologie du la formation statistique (à droite).*

Le projet Trial Solution (Buffa, 2005) consiste à prendre chaque ressource pédagogique et à la décomposer en objets pédagogique 'OP'. Chaque OP est représentée par son contenu sémantique et sa relation avec les autres OP ainsi que des métadonnées qui le concernent. Un outil d'annotation a été développé par ce projet qui permet d'indexer chaque nœud par des métadonnées et par des termes d'un thesaurus. On peut citer d'autres travaux similaires dans le domaine du e-learning (Bouzeghoub, 2005), (Hammache, 2008) et (Behaz, 2009).

Dans le domaine de l'audiovisuel, on présente le travail de (Troncy, 2004) qui articule les connaissances conceptuelles propres à un domaine à travers l'ontologie dans un contexte d'indexation de documents audiovisuels ayant pour thème 'Emission sportif télévisé'. Dans ce travail, on distingue deux ontologies :

- Une ontologie audiovisuelle qui permet de normaliser le sens des termes couramment utilisés pour décrire la structure et la mise en forme des documents audiovisuels. Par exemple le code ci-dessous représente un schéma qui indique qu'un magazine sportif est de type émission sportive et qu'elle commence toujours par une séquence plateau début, suivi par un certain nombre de séquences qui sont soit une séquence plateau, soit un enchaînement plateau-lancement-reportage et se termine par une séquence plateau fin.

- L'autre ontologie est du domaine qui modélise les concepts d'un sport particulier qui est le cyclisme (tour de France, magazine sportif, etc.).

```
              MagazineSportif   :=   ( PlateauDebut,
                                       ( PlateauCaracteriseParSaForme |
                                         (PlateauLancement, Reportage) )+,
                                      PlateauFin )
PlateauCaracteriseParSaForme    :=   (SequenceExtrait)*
                    Reportage   :=   (Interview | SequenceExtrait)*
```

Troncy et al ont également proposés de raccorder tous ces nouveaux types aux descripteurs MPEG-7. Pour cela, ils ont utilisés l'outil **SegmentTool** qui permet de découper temporellement des émissions et produit un squelette de description MPEG-7. Par la suite, ils ont modifiés ce squelette afin de typer chacun des segments en fonction de son genre, utilisant l'ontologie de l'Audio-Visuel et de l'annoter avec les concepts de la deuxième ontologie du domaine (exemple sport de cyclisme). La figure 16 montre la nouvelle hiérarchie des types de segments (en MPEG-7 étendu) contenant les différents genres et séquences de l'ontologie de l'audiovisuel (Troncy, 2004).

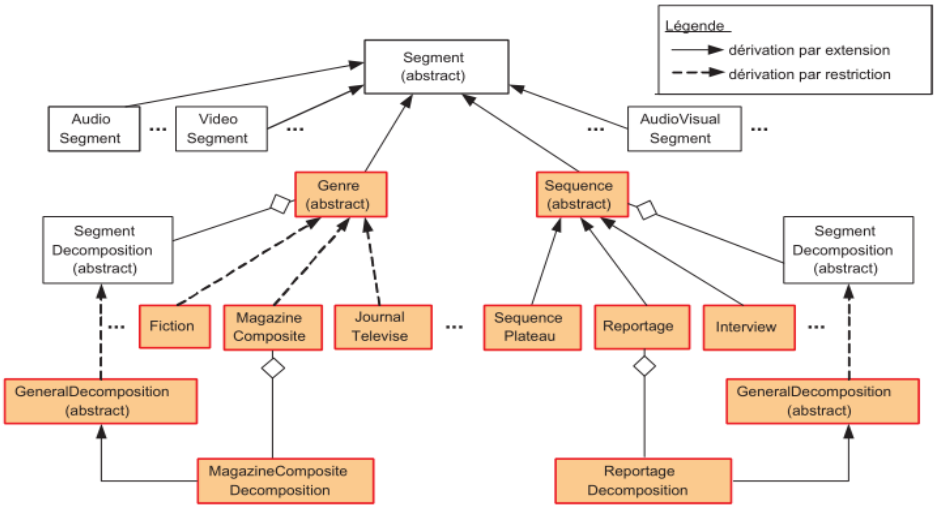


Figure 16. *Ontologie de l'audiovisuel : nouvelle hiérarchie des types de segments en MPEG-7.*

Le travail d'Isaac (Isaac, 2005) s'articule sur la description sémantique du contenu d'émissions télévisées ayant pour thème la médecine. Cette description repose sur la notion de patron d'indexation s'appuyant sur des scénarios d'utilisation existants et exploite les technologies issues du web sémantique. Il combine plusieurs ontologies à savoir : ontologie de l'AV et ontologies thématiques liées au domaine médicale (MENELA qui décrit le domaine des pathologies coronariennes et contient les concepts

liés à la chirurgie cardiaque correspondant au thème du corpus, GALEN contient les concepts liées à tous les domaines médicaux).

Isaac et Troncy ont eux aussi utilisé dans leurs travaux, l'outil '*SegmentTool*' pour produire une structure documentaire. Ensuite, ils décrivent les éléments AV en leur assignant certaines assertions de relations avec des concepts du domaine, relations de nature strictement représentationnelle ou plus interprétative. L'exemple de la figure 17 donne une idée des descriptions qu'Isaac et .al ont cherchés à obtenir. Ils ont mis en valeur la distinction entre les deux types de connaissances qui les intéressent.

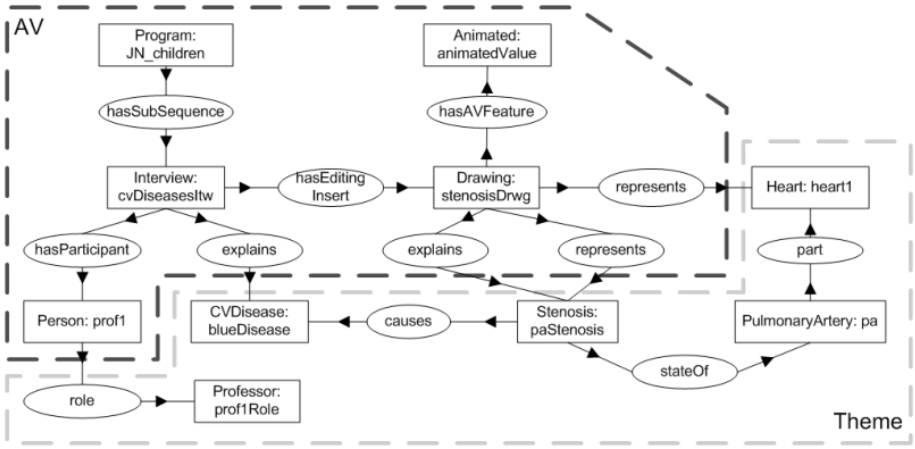


Figure 17. *Exemple de description d'un segment vidéo (Isaac, 2004)*

Ils ont utilisé le moteur d'inférence ***Sesame*** pour stocker et interroger les ontologies et les assertions. Donc leur système est implémenté dans un module d'inférence sur mesure, où les axiomes et les règles de RDFS sont complétés par ceux de l'OWL-DLP et des règles de raisonnement spécifiques aux ontologies exploitées.

(Charhad, 2005) a proposé un modèle pour la représentation du contenu sémantique des documents vidéo. Ce modèle permet la prise en compte synthétique et intégrée des éléments d'informations issus de chacune des modalités (image, texte, son). Il a développé et testé des outils sur des parties du modèle sur la collection TRECVID (2003 et 2004) : Un outil d'extraction des concepts et où la liste des concepts extraits automatiquement figure parmi une liste d'entités nommées (nom d'une personne, nom d'une place géographique, nom d'une organisation) et un outil de détection et de reconnaissance de l'identité du locuteur qui se base sur l'analyse de la transcription automatique de la parole dans un document vidéo. Les résultats de ces deux outils sont

exploités pour générer une description relationnelle où les relations conceptuelles sont inférées selon le type média. Par exemple, les relations 'parle' et 'parle de' sont extraites de la transcription et de la segmentation en locuteurs du contenu audio.

Nous allons citer également certaines études récentes et d'importance manifeste (Dong, 2010) et (Carbonaro, 2008), sur le traitement du contenu sémantique des présentations vidéo dans le domaine de e-learning. Dong et al. (Dong, 2010), offrent un modèle de multi-ontologie d'annotation de documents multimédias. Ils se concentrent dans leur document sur les présentations vidéo des conférences, séminaires et la formation en entreprise. Ces derniers se composent, en général, de nombreux thèmes ou sujets, et chaque sujet couvre plusieurs diapositives. Cette structure hiérarchique essentielle permet la segmentation, l'indexation et l'accès à ce type de vidéo.

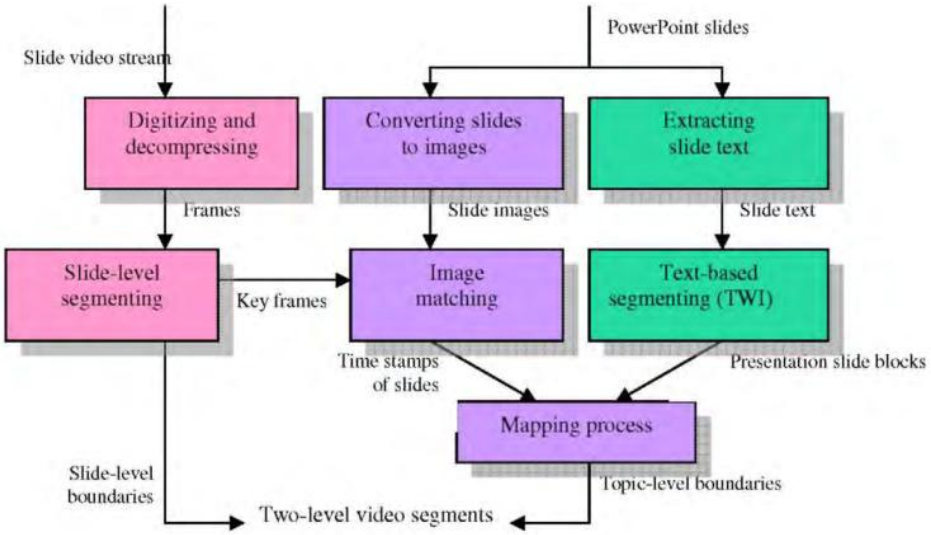


Figure 18. *Processus de segmentation des présentations vidéo (Dong, 2010)*

La segmentation vidéo est la première étape dans leur processus d'annotation. Cette tâche a été automatisé en prenant en considération les deux sources de données suivante: *Slide video stream* (c.-à-d. le flux vidéo d'une explication des diapositives capturés lors de la présentation) et *Slide PowerPoint* (c.-à-d. le fichier PowerPoint). La figure 18 explique les étapes de ce processus.

Une segmentation vidéo à deux niveaux est étudiée: niveau sujet et niveau slide. La segmentation au niveau Slide opère sur les flux vidéo de diapositives capturées par une caméra fixe, tandis que la segmentation au niveau sujet se base sur l'extraction du texte d'une diapositive via le module TWI. Ce dernier génère une séquence de blocks de

diapositives, dont chacun examine un sujet. Pour associer chaque block avec son segment vidéo qui lui correspond, une relation temporelle entre une diapositive d'un flux vidéo et l'ensemble de diapositives doit être établie. Ceci est accompli en faisant correspondre des images de diapositives (qui ont été converties à partir de fichier PowerPoint) avec des images clés extraites après la segmentation de flux vidéo. En basant sur des informations temporelles de chaque diapositive, les blocks peuvent être mappés aux segments vidéo ayant un sujet.

Une fois la segmentation achevée, chaque segment est annoté par une ontologie multimédia (OM) et plusieurs ontologies de domaine. L'ontologie «OM» est basé sur la norme MPEG7, mais se concentre sur l'aspect du contenu description. Il contient trois types de classes ou des concepts (voir figure 19): concepts multimédias (image, vidéo, audio, vidéo du segment, et etc...), concepts non-multimédia (agent, lieu, temps, concepts, etc...) et descripteur d'ontologies de domaine (par exemple comme ontologie des gènes 'Gene Ontology' 'GO'). Nous notons que l'aspect pédagogie est absent dans l'annotation dans ce travail.

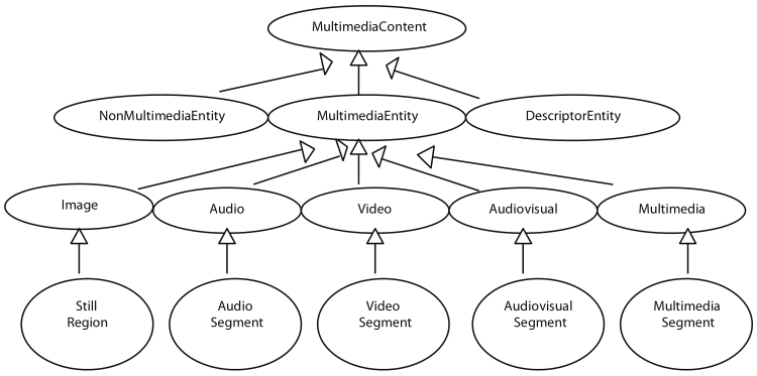


Figure 19. *La taxonomie de l'ontologie Multimédia (Dong, 2010)*

La figure ci-dessous illustre la façon d'intégrer plusieurs ontologies de domaine avec l'ontologie multimédia MO pour l'annotation d'un segment vidéo. La classe "VideoSegment" est de type 'multimedia entity'. Trois propriétés définies dans l'ontologie MO sont listées, "hasStartTime," "hasAbstract," and "hasDominantColor." L'ontologie de fouille de donnée (Data Mining Ontology ; DMO) et l'ontologie des gènes (Gene Ontology ; GO) sont deux domaines qui sont intégrées avec MO. "hasDMOAnnotation" et "hasGOAnnotation," de l'ontologie DMO et GO respectivement, sont additionnées à la classe "VideoSegment" comme deux propriétés.

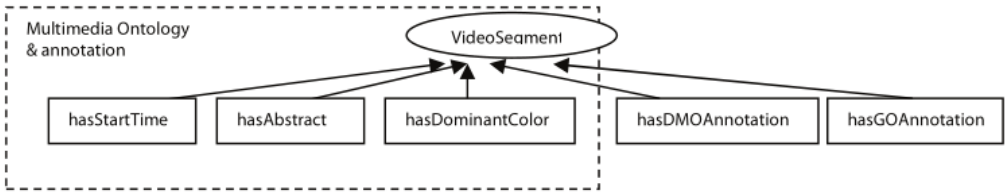


Figure 20. *Intégration des ontologies de domaine dans celle du multimédia, (Dong, 2010).*

2.3.5 Discussion

Nous constatons que l'indexation par le contenu sémantique des documents multimédia temporisés (cas des vidéos) nécessite obligatoirement une phase d'annotation. L'utilisation des ontologies pour annoter ce type de document permettre d'améliorer la performance d'un système d'indexation et de recherche en termes de rappel et de précision par la possibilité de faire des raisonnement au-dessus de ces annotations. Nous remarquons qu'il y a une combinaison de deux ou plusieurs ontologies afin d'être utilisés dans la phase d'annotation :

- une ontologie (dépend du type de corpus) est utilisé pour structurer logiquement le document vidéo en associant un type ou un genre pour ses différents segments (ontologie de l'audiovisuel : « émission, reportage,… ». ontologie multimédia : « image, audio,… » etc.) ;
- une ontologie thématique ou de domaine est utilisée pour décrire la sémantique de chaque segment (ontologie de sport cyclisme, ontologie médicale, …)

D'après les travaux étudiés ci-dessus, nous essayons d'extraire les limites et les insuffisances des systèmes étudiés et ainsi dons elles ne répondent pas à nos besoins.

En première lieu, on doit indiquer que dans le e-learning, le support de cours est diffusé sur le web selon deux catégories de documents : documents multimédias statiques (tels que : page web, PDF, .doc, etc.) et documents dynamiques ou temporels (tels que format audiovisuel, ou SMIL (SMIL, 2004)). Le nombre des documents de la deuxième catégorie ne cesse de croitre alors que les travaux d'indexation dans le domaine de l'e-learning précités ne se basent que sur les documents de la première. Notons Bien que le travail de Dong se concentre sur l'indexation par le contenu sémantique de document de type présentations vidéo des conférences et séminaires (et non pas des cours pédagogique), nous remarquons que l'aspect pédagogique est absent dans la description sémantique du contenu de ces vidéos.

En deuxième lieu, la nature temporelle de ce type de documents engendre un certain nombre de contraintes sur leur gestion documentaire. En effet, la spécificité de ces documents est d'être des objets temporels. Par essence, cette temporalité ne se présente pas et ne se stocke pas ; ceci a plusieurs conséquences. L'une d'elles est l'imposition du rythme de lecture du document; Si la vidéo est d'une heure, il faut une heure pour la visualiser, et si l'information recherchée commence à la 12ème minute et dure 10 minutes, alors il faut attendre tout ce temps ou faire défiler les 11 premières minutes pour la retrouver.

En troisième lieu, les cours vidéo diffusés sur le web sont annotés selon des métadonnées d'ordre général (format, date de création, auteur, titre, mots clés et parfois résumé). On remarque bien que l'emploi du texte libre (mots clés, résumé) pour décrire le contenu empêche le contrôle de la sémantique des descriptions et ceci limite sérieusement les possibilités de raisonnement.

L'étudiant et l'enseignant se contentent d'utiliser un langage naturel pour exprimer leurs besoins. Prenons l'exemple de requête suivant : « *je cherche des segments vidéo qui montrent un exemple sur l'utilisation d'un tableau comme paramètre d'une fonction* ?»

Les moteurs de recherches offerts actuellement ne permettent pas une recherche par le contenu sémantique d'une séquence (ou segment) dans un cours vidéo car ce dernier n'est pas structuré pédagogiquement, et leur sémantique pédagogique n'est pas spécifié formellement. Il n'y a pas d'outils qui offrent cette possibilité jusqu'à présent.

Ainsi notre contribution se situe dans le contexte d'offrir à la communauté du e-learning (apprenant ou enseignant) un système qui aide à la recherche par le contenu sémantique des segments vidéos pédagogiques en utilisant les ontologies. Notons bien que le mécanisme de d'indexation des documents à partir des ontologies, d'après l'étude faite ci-dessus, s'inscrit dans le domaine du Web Sémantique par la proposition d'une mesure visant à pondérer le lien entre les individus de l'ontologie de domaine et les segments vidéo.

2.4 Conclusion

Dans cette section nous avons étudié les approches concernant la création d'un contenu pédagogique multimédia d'un côté et les approches de son indexation par la sémantique d'un autre côté. Nous avons essayé de présenter une analyse de quelques travaux de recherche dans ces deux axes et nous avons terminé avec une discussion. Dans cette dernière, nous avons démontré l'insuffisance des systèmes existants pour répondre aux besoins et objectifs que nous avons déterminés dans l'introduction générale.

Nous appuyant sur l'analyse précédente et ses manques, notre intention était de justifier les contributions de la thèse, de construire les bases de la proposition présentée dans les chapitre suivant 3 et 4. Surtout, on donne des éléments de réponse concernant, d'une part, la concrétisation de l'approche double codage de l'information phonétique dans un cours pédagogique multimédia (présenté dans le chapitre 3). D'autre part, on présentera dans le chapitre 4 notre approche d'indexation et de recherche par le contenu sémantique (basé sur les ontologies) des segments vidéo pédagogique.

Chapitre 3 : SACoPh, un système auteur pour l'enseignement médiatique de la phonétique

3.1 Objectifs

- améliorer la perception et la mémorisation des concepts de phonétique (tel que l'accent tonique) par la conception d'un modèle de cours spécifié en SMIL3.0 qui support l'approche de double codage ;
- proposer aux enseignants de langue un éditeur permettant la préparation des cours multimédia publiables sur le web (standard SMIL3.0), selon le modèle que nous devons conçus et qui s'adapte à l'enseignement des concepts de la phonétique. Cet outil doit avoir une souplesse d'édition via une interface la plus conviviale pour que l'on s'approche le plus possible du principe de WYSIWIG.

3.2 Modélisation de cours de phonétique

Nous avons examiné un support de cours de phonétique, et à partir de ce dernier nous proposons que le cours soit un ensemble de leçons où chacune est représentée par un document multimédia. Donc la leçon sera générée par notre outil selon un modèle que nous proposons comme suit. Puisque chaque leçon est un document multimédia, donc sa description se fait selon ses différentes dimensions :

3.2.1 Dimension structurelle

Chaque leçon contient un titre et un ensemble de règles de phonétiques accompagnées par des exemples. Chaque règle aura des exemples démonstratifs, et chaque règle ou exemple sera associé à sa prononciation (fichier audio). Les parties du texte d'un exemple sur lesquelles l'enseignant veut attirer l'attention de l'apprenant sont mis en exergue visuellement (portent une couleur et un style différents).

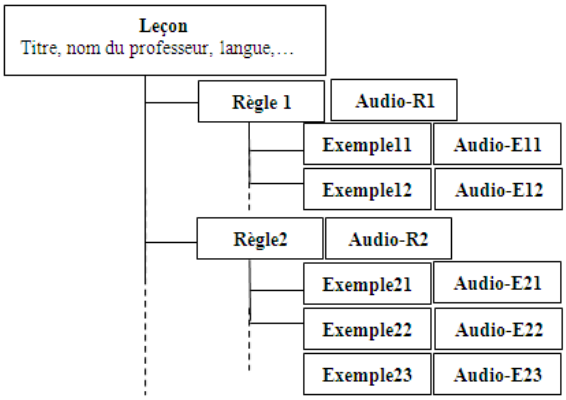


Figure 21. *Dimension structurelle de la leçon*

3.2.2 Dimension spatiale

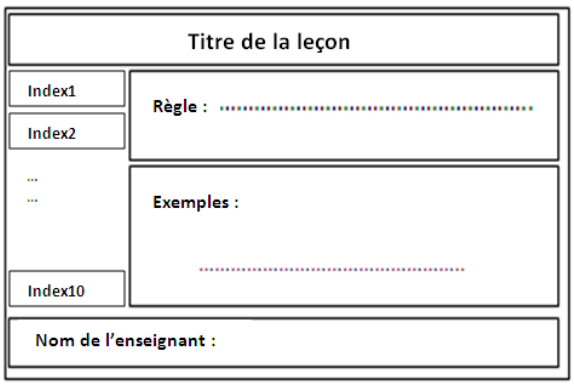


Figure 22. *Dimension spatiale de la leçon*

3.2.3 Dimension temporelle

Chaque règle ou exemple s'affiche en parallèle avec sa prononciation, et l'ensemble des règles s'affichent en séquence. Les règles se succèdent dans le temps l'une après l'autre. La figure suivante montre l'aspect temporel du document :

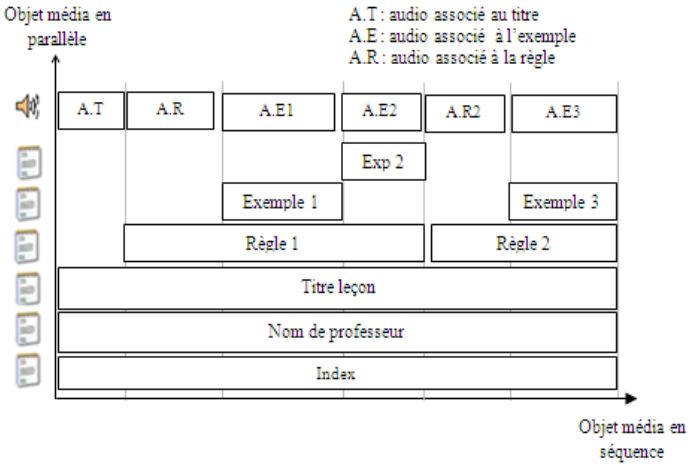


Figure 23. *Dimension temporelle de la leçon*

Les objets média sont représentés par des rectangles où la longueur reflète la durée d'affichage ou de présentation de l'objet correspondant.

3.2.4 Dimension hypermédia temporelle

L'objet index ou sommaire contient une liste des règles de la leçon. Les éléments de cette liste sont des zones cliquables ou un clic sur l'un d'eux permet de visionner la présentation de la leçon du début du segment qui lui correspond (la règle en question); c'est une navigation temporelle.

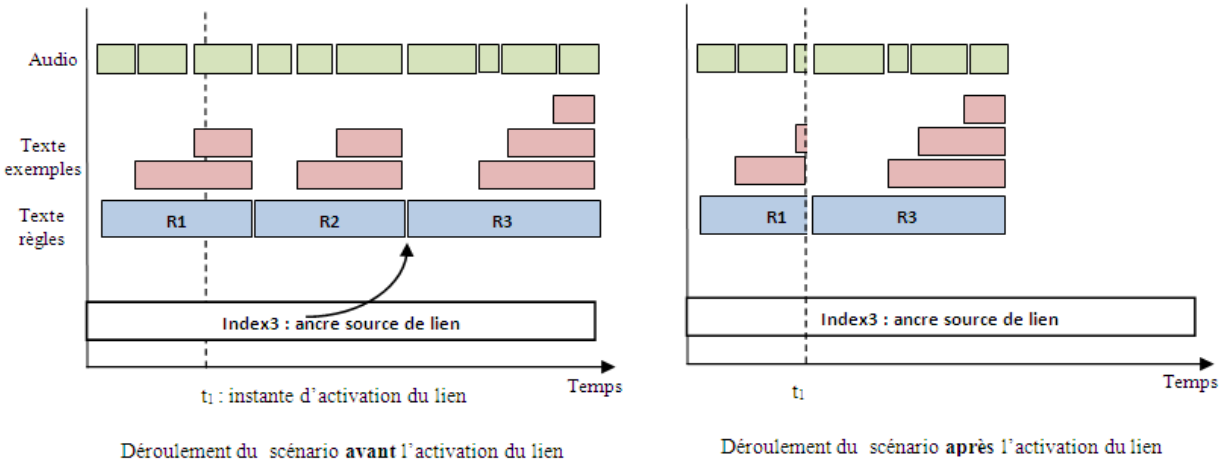


Figure 24. *Schéma montre le déroulement du scénario avant et après l'activation d'un lien temporel.*

3.3 Modélisation UML du système SACoPh

La démarche de notre système est d'associer une modalité de représentation typographique à la modalité orale de l'accentuation. L'association du style typographique à la qualité du discours parlé a été peu explorée. Il existe trois solutions pour associer ces deux types de représentation : celle qualifier par l'automatique (cas du système Prosodic Font), celle qualifier d'interactive et celle qui combine les deux solutions (cas du système SWANS).

La solution admise dans notre cas est la solution interactive car elle permet à l'auteur (enseignant) de choisir sa propre présentation d'une part et de décider où il souhaite mettre l'accent d'autre part.

Afin de cadrer les fonctionnalités attendues du système SACoPh, nous allons utiliser le langage UML (Uml, 2005) dans la modélisation de notre système. UML propose 13

types de diagrammes. UML n'étant pas une méthode puisqu'elle n'offre pas une démarche à suivre, son utilisation est laissée à l'appréciation de chacun. Dans notre cas nous avons choisi de modéliser notre système à travers 3 types de diagrammes à savoir : diagramme de cas d'utilisation, Diagramme de classes et diagramme de séquence.

3.3.1 Cas d'utilisation du système SACoPh

Les cas d'utilisation de notre système nous permettent de définir les fonctions principales dont l'acteur 'enseignant' doit disposer. Cependant, dans le cadre de ce travail les fonctions principales auxquelles on s'intéresse sont celles de créer une nouvelle leçon et d'ouvrir une leçon existe. Ces derniers sont ensuite raffinés, comme la figure 25 montre, par la fonction "inclue" pour être décomposés en plusieurs cas d'utilisation.

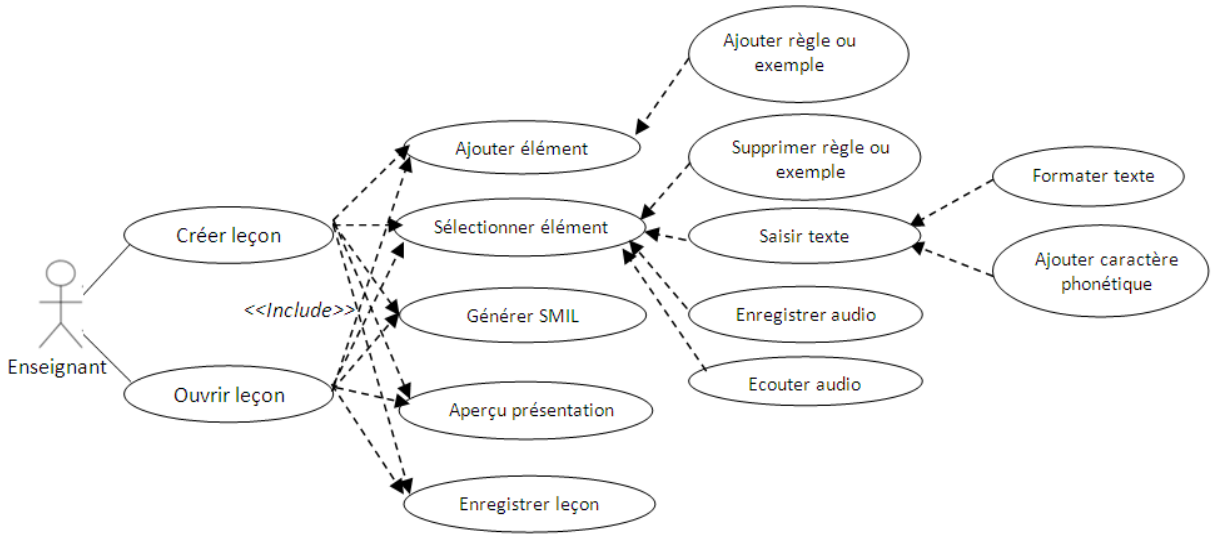


Figure 25. *cas d'utilisation par l'enseignant*

3.3.2 Architecture du système SACoPh

A partir des cas d'utilisation identifiés, nous pouvons extraire les modules permettant de les assurer. Le schéma de la figure 26 présente l'architecture générale du système. Dans ce qui suit, nous allons d'abord développer le processus de fonctionnement général du système, suivi par une explication détaillée de ses différents modules.

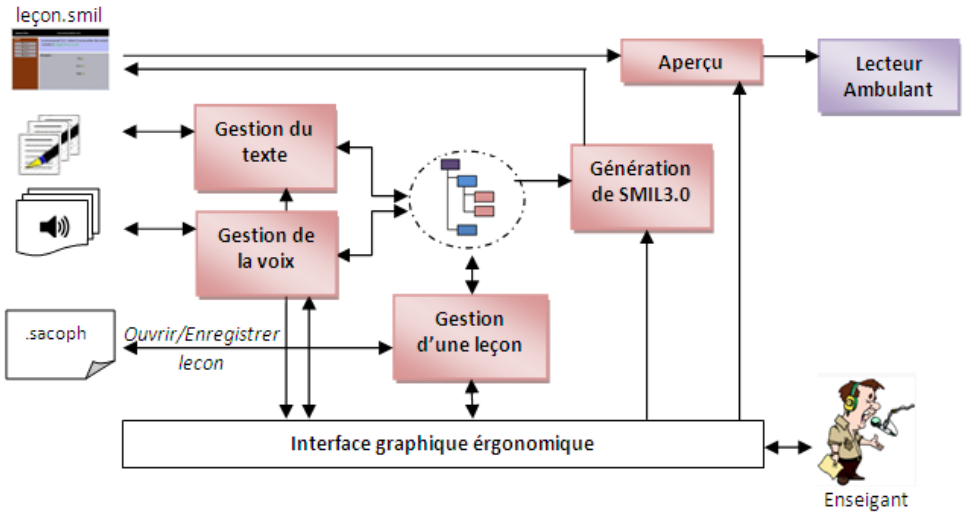


Figure 26. *Architecture générale de SACoPh*

3.3.2.1 Gestion d'une leçon

Ce module offre à l'enseignant la possibilité de gérer la leçon en gérant un arbre. Il peut créer, modifier ou supprimer un élément (objet) pédagogique règle ou exemple. La suppression d'une règle supprimera par conséquent tous les exemples qu'elle contient. Un document édité par notre système est sauvegardé dans un fichier XML (.sacoph) pour une mise à jour ultérieure. Un exemple de contenu d'une leçon enregistrer dans ce type de fichier est illustré comme suit :

```
<Lesson Name="the pronunciation of the final s" NomProfesseur="Mrs. Ameur"
       NomEtablissement="University de Batna">
  <Rule Name="Tips 1"/>
  <Rule Name="Tips 2">
       <Exemple Name="example 1"/>
       <Exemple Name="example 2"/>
       <Exemple Name="example 3"/>
       <Exemple Name="example 4"/>
  </Rule>
  <Rule Name="Tips 3">
       <Exemple Name="example 5"/>
       <Exemple Name="example 6"/>
       <Exemple Name="example 7"/>
       <Exemple Name="example 8"/>
  </Rule>
…
</Lesson>
```

3.3.2.2 Gestion du texte (marqueur typographique et symbole phonétique)

L'utilisateur, via les fonctionnalités de ce module, peut saisir le texte (d'une règle ou d'un exemple), insérer des caractères de l'alphabet phonétique en les sélectionnant sur une liste, comme il peut ajouter des marqueurs typographiques (une couleur, un font, un style et/ou une taille) sur des parties du texte pour attirer l'attention de l'apprenant

sur un phénomène de phonétique. Ce module génère, pour chaque élément pédagogique règle ou exemple, un code XHTML qui correspond à une telle spécification pour une exploitation ultérieur, soit pour sa mise à jour soit pour faciliter la génération du code SMIL par la suite. Nous prenons l'exemple d'une règle qui a le texte suivant :

s is pronounced /iz/ after the sounds:

/s/, /z/, /ʃ/, /ʧ /, /ʒ /, /ʤ/.

Ce module enregistre ce texte dans le code XHTML suivant :

```
<!DOCTYPE HTML PUBLIC… >
<html>
  <head><meta name="qrichtext" content="1" /><style type="text/css">
    p, li { white-space: pre-wrap; }</style>
  </head>
  <body style=" font-family:'MS Shell Dlg 2'; font-size:8.25pt; font-weight:400; font-
              style:normal;">
    <p style=" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px;
      -qt-block-indent:0; text-indent:0px;">
    <span style=" font-size:20pt; font-weight:600; color:#aa00ff;"> s is pronounced
    </span>
    <span style=" font-size:20pt; font-weight:600; color:#aa0000;"> /iz/ </span>
    <span style=" font-size:20pt; font-weight:600; color:#aa00ff;"> after the
          sounds:</span>
    </p>
    <p style=" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px;
      -qt-block-indent:0; text-indent:0px;">
     <span style=" font-size:20pt; font-weight:600; color:#ff0000;"> /s/, /z/,
        /&#8747;/, /&#679;/, /&#658;/, /&#676;/ </span>
    </p>
  </body>
</html>
```

3.3.2.3 Gestion de la voix

Ce module est concerné par l'enregistrement de la voix de l'enseignant dans un fichier audio sous format 'wav'. Il permet de réécouter ce qui a été enregistré ou de le supprimer comme il offre des informations sur l'objet tel que la durée.

3.3.2.4 *Générateur de la présentation SMIL*

L'enseignant peut exporter ses documents via ce module vers le format d'affichage SMIL 3.0 pour les publier ou les partager. Jusqu'à présent, seul le lecteur appelé Ambulant peut lire les présentations de la version 3.0. Donc, ce module doit parcourir l'arbre des objets règles et exemples pour générer le code SMIL3.0. La synchronisation entre ces objets est déduite implicitement à partir de leur ordre dans l'arbre. La durée

d'affichage d'une règle ou d'un exemple est déduite à partir de la durée de l'objet audio associé. Prenons une partie du code générer en l'expliquant par la suite.

```
……
<-- spécifier le segment temporelle  contenant la règle 3 et les exemples associés -->
<par xml:id="3" dur="46s">
<-- spécifier la prononciation de la règle1  et ses exemples -->
  <audio begin="1s" src="Tips 3.wav"/>
  <audio begin="20s" src="example 5.wav"/>
  <audio begin="26s" src="example 6.wav"/>
  <audio begin="31s" src="example 7.wav"/>
  <audio begin="37s" src="example 8.wav"/>
  <-- spécifier le texte de la règle  -->
  <smilText region="Regle">  <p>
   <span textFontWeight="bold" textColor="#aa00ff" textFontSize="20px">s is pronounced
      </span>
   <span textFontWeight="bold" textColor="#aa0000" textFontSize="20px"> /iz/</span>
   <span textFontWeight="bold" textColor="#aa00ff" textFontSize="20px"> after the
        sounds:</span></p>
   <p>
   <span textFontWeight="bold" textColor="#ff0000" textFontSize="20px"> /s/, /z/,
         /&#8747;/, /&#679;/, /&#658;/, /&#676;/.</span>
   </p>
  </smilText>
<-- spécifier l'exemple 1 qui commence à la 20ème seconde après la règle 1 -->
   <smilText begin="20s" region="Exemple"><p>
    <span textFontWeight="bold" textColor="#000000" textFontSize="20px">misse</span>
    <span textFontWeight="bold" textColor="#ff0000" textFontSize="20px">s</span></p>
   <-- spécifier l'exemple 2 : qui commence au 6ème seconde après l'exemple 1  -->
    <tev begin="6s"/> <p>
    <span textFontWeight="bold" textColor="#000000" textFontSize="20px">washe</span>
    <span textFontWeight="bold" textColor="#ff0000" textFontSize="20px">s</span></p>……
   </smilText>
</par> ……
```

Nous montrons ci-dessus une partie du code SMIL3.0 généré par SACoPh qui permet de synchroniser le texte d'une règle en parallèle avec sa prononciation. On remarque bien qu'après 11s, le texte du premier exemple se présente avec sa prononciation. On remarque les différents types de balises et attributs : celles qui servent à la synchronisation <par>, <tev>, begin et celles qui servent à la présentation de marqueur typographique <smilText>, <textFontFamily>, <textColor>,…etc.

Une table d'index est calculée automatiquement. Nous avons proposé une segmentation temporelle automatique du document. Ceci consiste à identifier le début et la durée de chaque segment, ensuite à placer des ancres pour chacun d'eux pour être référencé par la suite. Nous proposons qu'un segment temporel corresponde au temps de présentation d'une règle avec tous ses exemples : **c'est une entité indivisible**. On a

fait ce choix parce que la règle ne peut pas être parfaitement comprise qu'à travers ses exemples explicatifs, et ces derniers ne peuvent être dissociés de la règle. A chaque entrée de la table d'index, on associe un lien vers le segment temporel qui lui correspond via son marqueur. Le code suivant montre comment les segments sont marqué, et comment sont référencié par les entrés de la table d'index.

```
<-- spécifier en séquence les segments temporels et marquer chacun d'eux par
l'attribut xml : id-->
<seq>
   <-- segment 1 -->
   <par xml:id="1" dur="28s">
     <smilText region="Tips 1">  …… </smilText>
     <smilText region="exemple 1"> ……  </smilText>  ……
   </par>
   <-- segment 2 -->
   <par xml:id="2" dur="15s">  …… </par>  ……
</seq>
<-- remplir la table d'index;'Rule 1' est un ancre vers le segment identifié par 1-->
   <a href="#1"> <smilText region="Index1"> Rule 1</smilText>  </a>
   <a href="#2"> <smilText region="Index2"> Rule 2</smilText>  </a>
   ……
```

La table index offre une navigation temporelle durant la représentation de la leçon. L'étudiant peut avancer ou reculer la présentation vers le début d'une règle qu'il veut réécouter en cliquant sur son lien indiqué dans cette table. La figure 27 montre la présentation finale d'une leçon par le lecteur Ambulant.

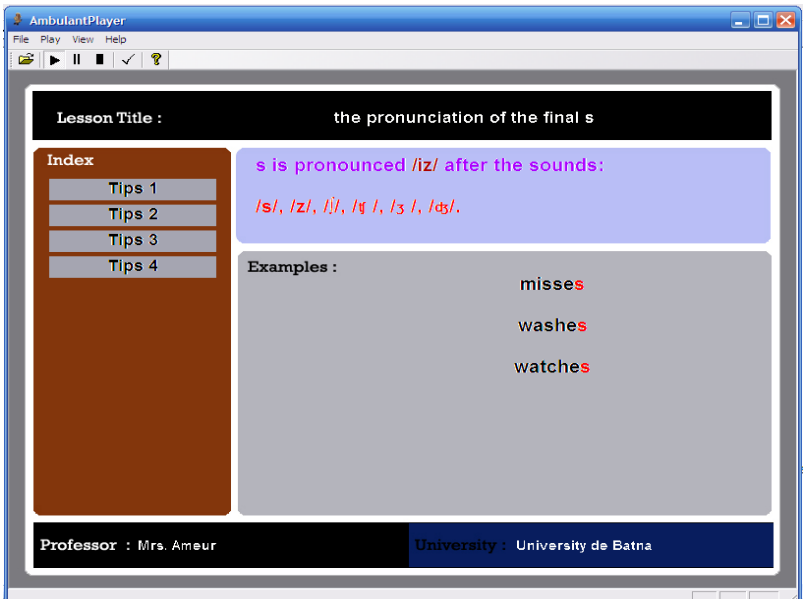


Figure 27. *Présentation du cours généré en SMIL 3.0 avec le lecteur ambulant*

3.3.3 Aspect statique du système : diagramme de classes

Le diagramme de classes est considéré comme le diagramme le plus important de la modélisation orientée objet. Comme le diagramme de cas d'utilisation montre le système du point de vue de l'enseignant, le diagramme de classes montre la structure interne du système, qui est composée principalement de huit classes. Chaque classe est décrite par des attributs et des méthodes. L'enseignant interagit qu'avec la classe MainWindow, et c'est à travers cette classe qu'il peut manipuler les autres classes du diagramme, du fait qu'elle est reliée par des relations de composition aux classes Leçon, Configuration et AudioInputOutput.

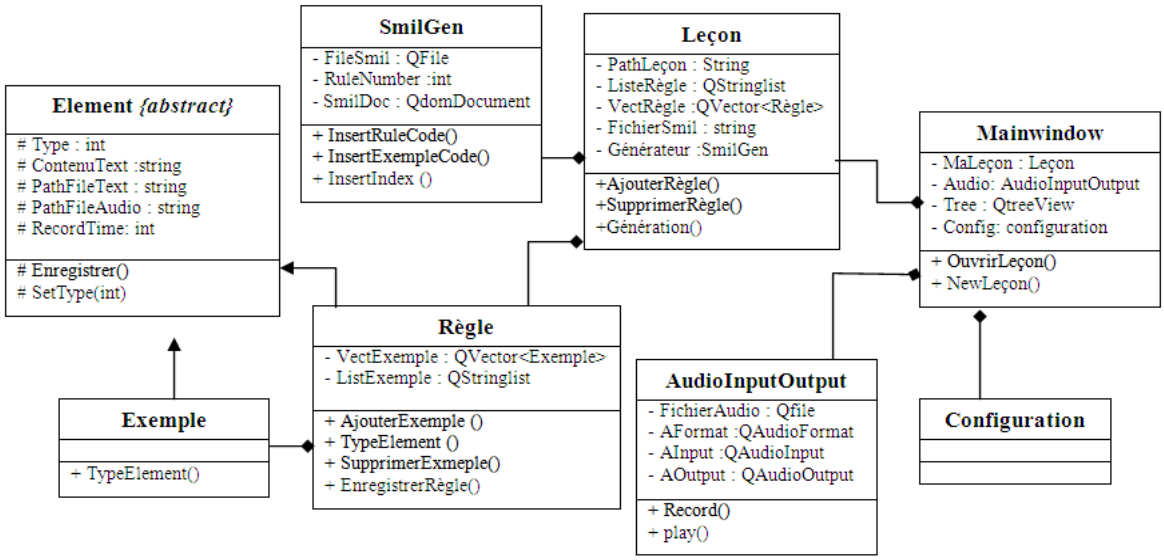


Figure 28. *Diagramme de classes*

3.3.4 Aspect dynamique : diagrammes de séquence

L'aspect dynamique de la modélisation est apporté par les diagrammes d'interaction. Ces diagrammes permettent d'établir un pont entre la vision externe du système (cas d'utilisation) et la vision statique et structurelle (diagramme de classes). Ils montrent comment les instances communiquent entre elles pour réaliser un cas d'utilisation.

Les diagrammes de séquence montrent les interactions entre les acteurs et les grandes fonctions du système. La Figure 29 montre l'interaction entre l'acteur enseignant et la fonction MainWindow::OuvrirLecon() qui affiche la fenêtre pour sélectionner une leçon. Une leçon est composée de plusieurs règles, et une règle est composée à son tour de

plusieurs exemples. Donc pour ouvrir une leçon, on a besoin de deux boucles (loop) imbriquées.

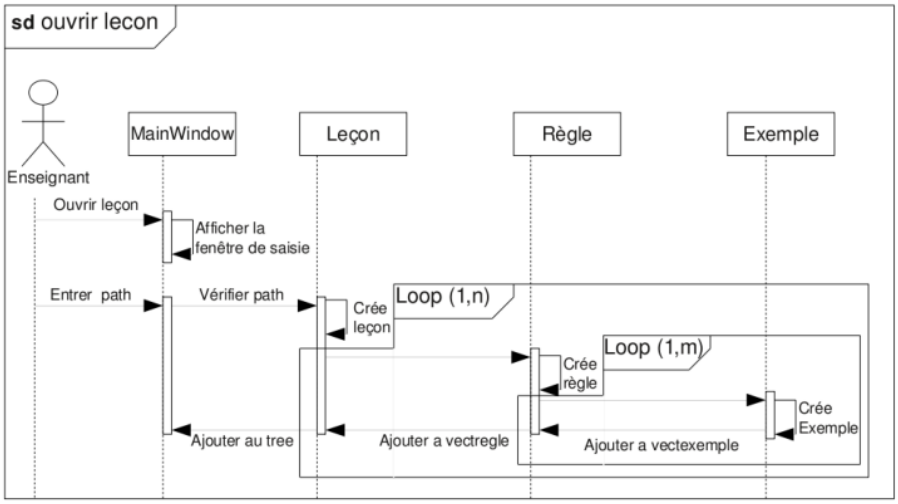


Figure 29. *Diagramme de séquence du cas 'ouvrir leçon'*

Un autre diagramme de séquence pour le cas d'utilisation 'générer SMIL' de la leçon en cours est illustré dans la figure 30. L'objet MainWindow demande à l'utilisateur de sélectionner la destination du fichier smil à générer. Puis il envoi le message génération à l'objet leçon. Ce dernier va créer un objet SmilGen qui s'occupe de la construction du code SMIL. Donc l'objet SmilGen doit parcourir le vecteur de règles et les vecteurs des exemples pour chaque règle (deux boucles imbriquées).

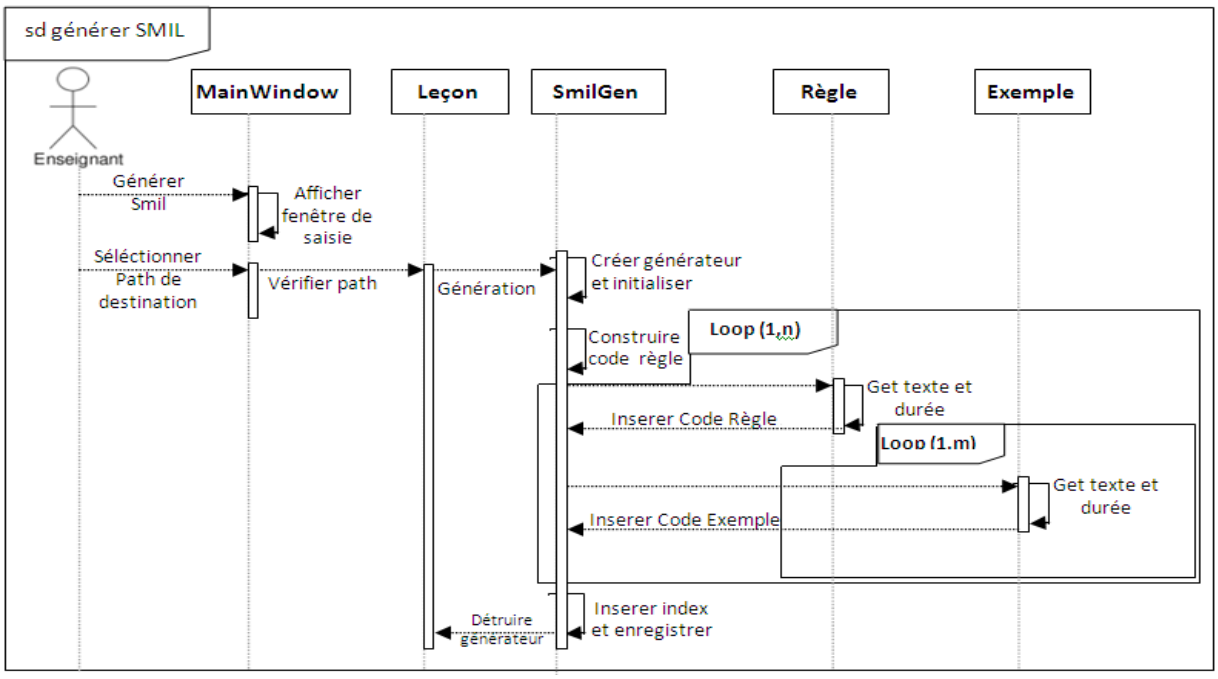


Figure 30. *Diagramme de séquence du cas 'générer SMIL'*

3.3.5 Interface de SACoPh

Nous avons conçu l'interface du système de tel sorte qu'elle se rapproche du principe de WYSIWYG (voir paragraphe 2.21.2). Elle sera composée de deux vues. L'une représente la structure de la leçon sous forme d'un arbre hiérarchique. A travers cette vue, l'enseignant pourra ajouter, déplacer, supprimer des éléments pédagogique à savoir règle ou exemple. L'autre présente l'élément pédagogique de la leçon (règle ou exemple) sélectionné dans la première vue, où l'enseignant peut médiatiser en temps réel cet élément en lui éditant un texte formaté et enregistrant sa prononciation sous format audio. De ce fait, la vue hiérarchique représente la table d'index dans la présentation finale de la leçon dans le même endroit, et la vue édition d'un élément pédagogique (règle ou exemple) correspond à la présentation finale de ce dernier presque dans la même région.

L'interface de notre système supporte de plus les fonctionnalités suivantes :

- Permettre une Manipulation direct des objets média : modification interactive des attributs temporels des objets média, impliquant une propagation automatique des contraintes temporelles ;
- Tous les services supportés par notre système sont fournis via une interface graphique conviviale, ergonome et facile à utiliser et qui tient compte des comportements déjà ancrés chez les enseignants ;
- Le système offre un clavier virtuel pour les symboles phonétiques ;
- Mode d'édition basé sur les templates : les documents générés ayant le même modèle.

3.3.6 Mise en œuvre

Après avoir explicité les différentes fonctionnalités envisagées par le système, ainsi que sa conception détaillée, nous allons présenter dans ce qui suit le prototype réalisé. Il consiste en l'implémentation des fonctions principales proposées, en vue de tester l'opérationnalité du système.

3.3.6.1 Environnement de mise en œuvre

Pour développer notre application, nous avons choisi l'environnement Linux (Ubuntu

9.10 & 10.04) qui est un système d'exploitation multitâches et multi-utilisateurs très puissant basé sur la distribution Debian GNU/Linux. Ce système est connu pour sa rapidité et sa facilité d'utilisation et son environnement de développement très riche.

Notre système est développé en langage de programmation orienté objet C++. Elle est basée sur le Framework Qt, qui est un Framework d'applications multiplateforme (écrire une fois et compiler n'importe où) et d'interface homme machine (IHM) orienté objet. Qt offre toutes les fonctionnalités nécessaires pour le développement d'applications à interface graphique avancée pour plates-formes de bureaux et embarquées. Qt utilise les interfaces de programmation graphiques natives de chaque plate-forme prise en charge, afin de tirer pleinement parti des ressources du système et pour que les applications possèdent une apparence et un ressenti natif.

3.3.6.2 Architecture Modèle/Vue

Notre application est basée sur l'architecture d'implémentation ***Modèle/Vue*** qui fournit un cadre d'implémentation plus simple basée sur les mêmes principes que l'architecture MVC (Modèle-Vue-Contrôler).

La partie modèle représente le comportement de l'application : traitements et interactions avec les données. Il décrit où sont contenus les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

Tandis que la partie vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). L'interaction avec l'utilisateur est assurée par un ensemble de modules et de dialogues qui apparaissent au fur à mesure de l'exécution de l'application.

3.3.6.3 *Implémentation*

Ci-dessous une figure qui illustre une vue générale de l'interface de SACoPh. L'enseignant peut sélectionner une fonction (ajouter, modifier, réutiliser et/ou

supprimer) un élément pédagogique (règle ou exemple) via un menu contextuel voir figure31.

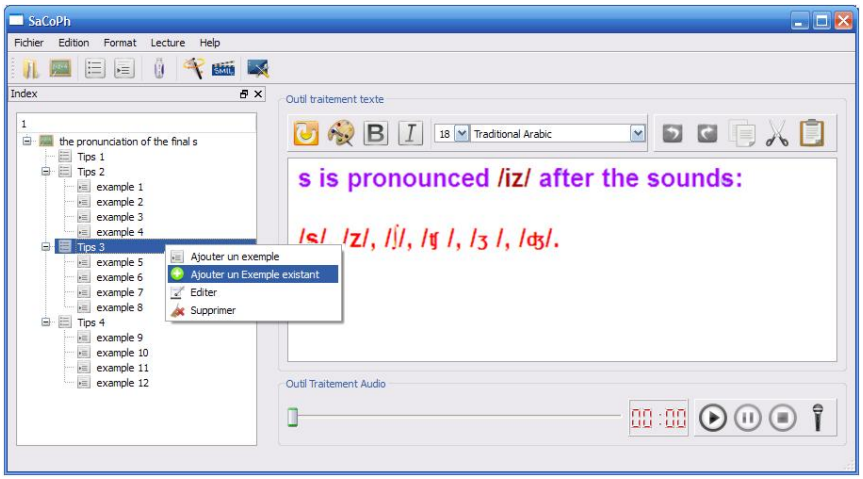


Figure 31. *Interface de SACoPh*

L'outil utilisé pour toute manipulation audio est illustré par la partie graphique qui est composée d'un slider, d'un timer, et de boutons de contrôles comme le montre la figure 32. Si l'utilisateur clique sur un de ces boutons, un signal est envoyé par ce dernier et réceptionné par une des fonctions de la classe AudioInputOutput comme par exemple la fonction Play.

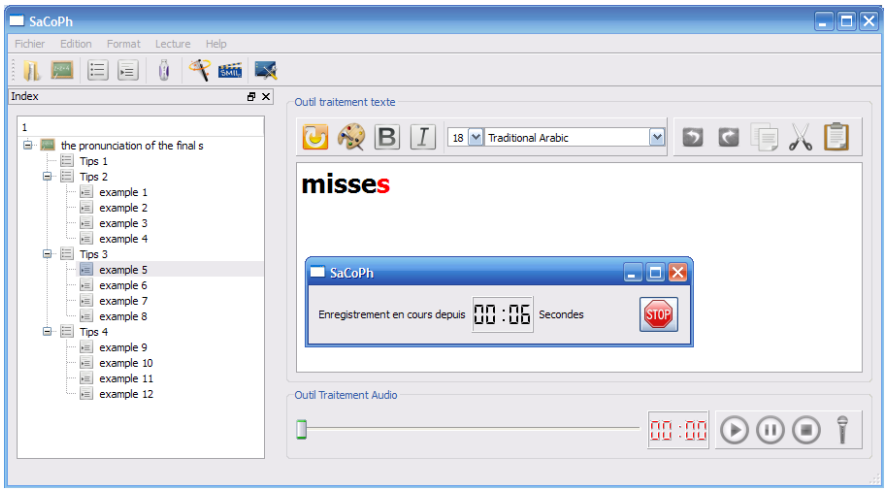


Figure 32. *Cas d'enregistrement de la prononciation de l'exemple 5*

Le système offre aussi la possibilité de configurer le système en choisissant : le périphérique audio pour la sortie et l'entrée de la voix, le path du lecteur de SMIL3.0 et un font globale pour le texte associé aux éléments de la leçon.

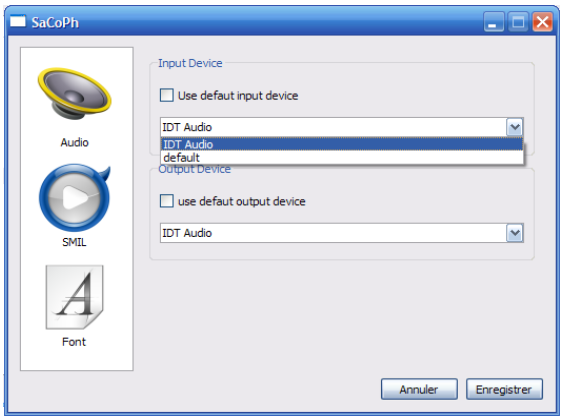


Figure 33. *Paramètres de configuration : périphérique audio, lecteur de smil et font de texte globale.*

L'approche incrémentale (voir paragraphe 2.2.2) est réalisée dans notre système par la fonction aperçu dans la barre d'outils. Cette dernière provoque la création d'une instance du lecteur Ambulant et ouvre la présentation SMIL générée dans ce même lecteur. Notons bien que ce lecteur est le seul (jusqu'à présent) qui peut jouer une présentation en SMIL3.0.

3.4 Conclusion

Notre premier axe de recherche s'articule sur la modélisation et la construction d'un contenu pédagogique multimédia structurés et temporisés appliqués dans l'enseignement à distance, et plus précisément, à l'enseignement de la phonétique d'une langue. Nous avons présenté l'apport important de l'approche du double codage dans l'amélioration de l'apprentissage. Pour cela, nous avons conçu un modèle de document multimédia temporisé qui concrétise cette approche. Nous avons développé un système auteur appelé SACoPh, qui génère des cours de phonétique publiables via le web, selon ce modèle, tout en exploitant les nouvelles fonctionnalités de la version 3.0 du standard SMIL. Ce système est conçu pour être destiné aux enseignants non informaticiens, il offre une interface plus simple, ergonome et plus conviviale que possible.

Le deuxième Axe de recherche qui nous intéresse est l'indexation et la recherche dans un contenu pédagogique multimédia. Nous utilisons dans ce cas les ontologies afin de traiter la sémantique du contenu. Le quatrième chapitre présente notre contribution dans cet axe.

Chapitre 4 : Intégration des ontologies dans la construction et l'indexation de contenu pédagogique Multimédia

4.1 Introduction

Dans un contexte de recherche d'information vidéo par le contenu sémantique, la modélisation est une tâche importante et nécessaire à partir de laquelle les index seront formulés et grâce à laquelle le processus de recherche sera plus efficace et plus précis. Notre approche s'inscrit dans un contexte de modélisation et d'indexation des cours vidéo pédagogiques et de recherche par la sémantique des segments dans ce type de vidéos.

Au niveau théorique, notre contribution consiste à la proposition et à la construction de deux types d'ontologies, une pour la structuration pédagogique d'un cours vidéo et l'autre pour la description du contenu sémantique de ses différents granules. Ces deux ontologies seront utilisées dans la phase d'annotation conceptuelle.

Notre contribution au niveau expérimental consiste, en premier lieu, à l'annotation conceptuelle d'un corpus de cours vidéo d'une formation professionnelle continue diffusés sur le web par l'université NETTUNO dans le cadre du projet MedNet'U. L'annotation ce fait plus précisément sur une dizaine de leçons vidéo en relation avec le module '*structure de données et technique de programmation*'. En deuxième lieu, notre contribution consiste à la conception et à l'implémentation du prototype IRSeCoV : un système d'Indexation et de Recherche Sémantique des segments Vidéo pédagogiques via les annotations conceptuelles associées au corpus en question. Cela sera suivi par une expérimentation.

4.2 Construction des Ontologies

On a besoin de deux ontologies pour modéliser le contenu de cours en format vidéo. La première sera construite pour structurer pédagogiquement un cours vidéo et sera appelée ontologie pédagogique d'un cours vidéo, et la deuxième est l'ontologie de domaine d'enseignement et comme son nom l'indique, elle modélisera les connaissances d'un domaine d'enseignement (un module à enseigner) pour une description sémantique plus profonde de ce type de cours.

4.2.1 Ontologie de domaine d'enseignement

Un domaine d'enseignement correspond à un module dans une formation. Un module aborde ou enseigne un ou plusieurs concepts. Un concept peut être décomposé en plusieurs concepts, il peut dépendre d'un ou de plusieurs concepts comme il peut être le prérequis d'un ou de plusieurs concepts aussi. Donc trois types de relations peuvent exister ente deux concepts : 'est_décomposé_en', 'dépend_de' et 'est_pré_requis_de'. On remarque que la relation 'est_pré_requis_de' a la transitivité comme caractéristique tandis que la relation 'dépend_de' est symétrique et que la relation 'est_decomposé_en' est antisymétrique.

On constate que l'exploitation des caractéristiques de ces relations peut engendrer ou inférer des instances n'existant pas dans la base des faits initiaux.

Prenons l'exemple de domaine d'enseignement : « structure de données et algorithmique » qui aborde les concepts : fonction, paramètre, passage_paramètre_par_valeur, liste, pointeur et enregistrement. Les instances des relations qui peuvent exister entre ces concepts sont représentées comme suit:

est_composé_de (fonction, paramètre). Signifie : fonction est composée de paramètre.

depend_de (fonction, passage_paramètre_par_valeur). Puisque cette relation est symétrique, le système d'inférence peut déduire l'instance suivante : *depend_de* (passage_paramètre_par_valeur, fonction).

est_pré_requis de (pointeur, liste) et *est_pré_requis de* (liste, arbre) **=>** *est pré_requis de* (pointeur, arbre). La relation est transitive.

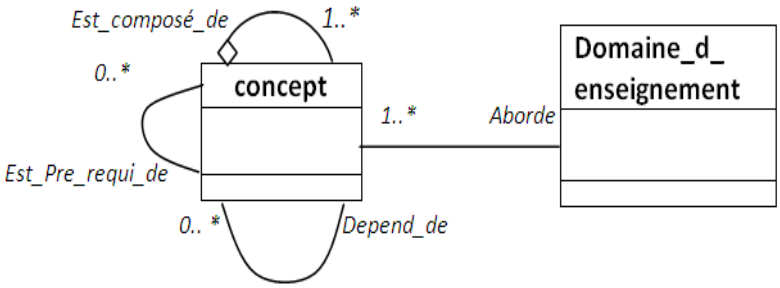


Figure 34. *Ontologie de domaine d'enseignement.*

On peut mentionner l'existence des instances de la classe concepts qui sont identiques. On cite comme exemple : *boucle et instruction de répétition, adresse et pointeur, enregistrement et structure, tableau à deux dimensions et matrice, paramètre et attribut, etc.* Cette sémantique est spécifiée en langage OWL (OWL, 2004) par la propriété 'SameAs' entre individus.

Une instance de cette ontologie engendre une ontologie du domaine à enseigner D (un module bien précis). Elle peut être manuelle ou semi-automatique. Dans le deuxième cas, ce processus fait appel à des outils d'ingénierie linguistique (comme LEXTER) pour l'extraction des termes candidats à partir d'un ou de plusieurs supports de cours textuels d'un domaine particulier D. Ces termes représentent l'extension de la classe concept. Ces termes doivent ensuite être sélectionnés, triés par un expert dans le domaine et organisés hiérarchiquement selon la relation 'composé_de'. Puis, la sémantique du domaine est affinée par la précision des instances identiques de la classe concept et des instances des deux relations 'est_pré_requis_de' et 'depend_de' qui peuvent exister entre les différents concepts.

Dans notre cas, nous avons créé une instance de cette ontologie pour le module SDD : 'Structure De Données' avec l'éditeur Protégé 2000. Ci-dessous un extrait du code OWL généré par cet éditeur.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns=http://www.owl-ontologies.com/Ontology1277939276.owl#
      ……
<domaine_d_enseignement rdf:ID="structure_de_données">
<aborde>
 <concept rdf:ID="instruction">
     <composer_de rdf:resource="#affectation"/>
     <composer_de rdf:resource="#instruction_de_controle"/>
     <composer_de df:ID="#instruction_de_repetition"/>
```

```
 </concept>
 <concept rdf:ID="boucle"/>
       <owl:sameAs rdf:resource= "#instruction_de_repetition"/>
 </concept>
 <concept rdf:ID="passage_parametre_par_valeur">
            <depend_de rdf:resource="#fonction"/>
 </concept>
 <concept rdf:ID="pointeur">
   <est_pre_requi_de rdf:resource="#liste"/>
 </concept>
      …
</aborde>
```

4.2.2 Ontologie pédagogique de cours vidéo

Un cours vidéo est présenté dans une ou plusieurs leçons vidéo. Chaque leçon vidéo (elle peut correspondre à un chapitre ou sous chapitre) est subdivisée ou segmentée en plusieurs segments temporels. Le segment correspond à l'explication d'un ou de plusieurs diapositifs ayant le même titre. Donc le segment dans ce cas doit représenter une idée ou un sujet ou une unité d'intérêt qui sera retourné par notre système de recherche s'il représente une réponse pertinente à une requête.

Un segment ou une diapositive contient un ou plusieurs objets pédagogiques 'OP'. Ces derniers peuvent être une définition, un exemple, un exercice, une solution_exercice, une illustration, une règle, un théorème, une démonstration, etc…

Pendant la consultation de quelques leçons vidéo du corpus que nous avons choisi, nous avons constaté qu'une diapositive contient une définition d'un concept suivi par un petit exemple. Comme on a remarqué qu'un exemple peut être présenté dans deux ou trois diapositives ayant le même titre, on a opté pour cette manière de structuration des segments.

Cet OP concerne un ou plusieurs concepts d'un domaine d'enseignement. La relation '*concerne*' gère l'alignement des deux ontologies à savoir l'ontologie pédagogique de cours vidéo (CV) et celle de domaine d'enseignement (DE). Pour se faire, il faut importer la deuxième (DE) dans la première (CV).

Une question qui se pose : pourquoi un objet pédagogique concerne plusieurs concepts et non pas un. Si on prend l'exemple de la leçon vidéo ayant le titre 'les fonctions', où il y a une diapositive avec un *exercice* concernant l'utilisation d'un tableau comme

paramètre d'une fonction, on constate que l'OP de type 'exercice' concerne les trois concepts (soulignés) de l'ontologie de domaine d'enseignement SDD (il n'existe pas à notre connaissance de moteur de recherches qui répond à une requête de ce type).

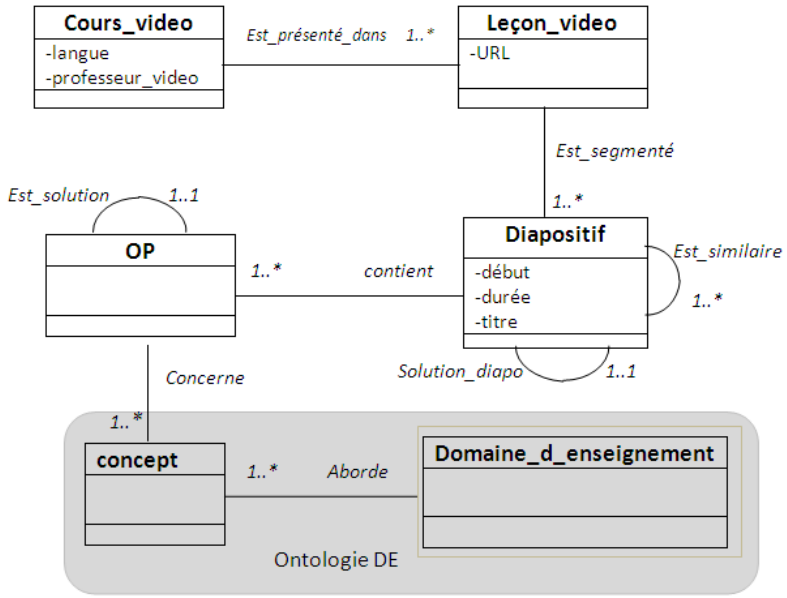


Figure 35. *Ontologie pédagogique de cours vidéo.*

La figure 35 montre deux relations qui peuvent exister entre deux instances de la classe diapositive à savoir '*est_similaire*' et '*est_solution*'. Ce type d'instances ne peut exister que si une condition est vérifiée. Par exemple deux instances de type diapositive ne peuvent être similaire que s'ils contiennent des objets pédagogique qui abordent deux concepts similaires. Dans le cas de la deuxième relation, s'il existe une instance de la relation est_solution entre deux OP et que ces deux objets sont contenus dans deux instances diapositives différents, alors il y a une instance de la relation solution_diapo entre ces deux derniers.

La codification terminologique (définition des concepts et relations avec un langage tel qu'OWL) des deux ontologies a été faite par l'outil Protégé 2000 qui génère un code OWL (voir Annexe 4). La construction et la codification de la base des faits (les instances) sont détaillées dans la section suivante.

4.3 Processus d'annotation

Certains auteurs comme Charhad considèrent ou nomment la phase d'annotation comme l'indexation manuelle ou assistée. Dans notre cas, cette phase décrit les

documents vidéos pédagogiques en considérant deux aspects: pédagogique qui identifie les éléments constituant la structure pédagogique du document (diapositive, objet pédagogique de type définition, exemple,…) et thématique qui décrit ce que représente chaque élément comme concept dans le domaine considéré.

Troncy et Isaac ont utilisés l'outil segment Tool, tandis que Charhad a utilisé VidéoAnnex pour segmenter le corpus des documents AV traité. Dans notre cas, nous avons développé un outil de segmentation et d'annotation des cours vidéo, appelé OntoCoV, sur la base des ontologies que nous avons créés.

La description ou l'annotation, via OntoCoV, commence par la localisation ou la segmentation temporelle en entités qui portent sur une idée indivisible. Il s'agit de repérer dans le temps des segments, où chacun correspond à l'exposition d'un ou de plusieurs diapositives ayant le même titre. Ensuite, chaque segment doit caractériser sa structure pédagogique. Ces deux étapes engendrent l'instanciation de l'ontologie de cours vidéo. Ensuite, la description du contenu sémantique de chaque segment se fait par l'association des concepts de l'ontologie d'un domaine d'enseignement particulier aux différents objets pédagogiques (OP). Cette ontologie doit avoir une relation étroite avec la leçon vidéo en cours d'annotation. Donc, l'outil OntoCoV donne la possibilité à son utilisateur d'intégrer une ontologie d'un domaine particulier. Cette ontologie est présentée dans notre outil sous forme d'arbre graphique, ce qui permet à l'usager de la parcourir rapidement et de sélectionner, à tous les niveaux (hiérarchie des instances concept), le concept qui lui semble pertinent pour son indexation. A la fin, le système génèrera l'ensemble des annotations dans le langage opérationnel OWL. Cette annotation constitue une base qui sera exploitée dans les phases suivantes.

La figure 34 montre l'interface de l'outil OntoCoV qui est divisée en deux régions :

a) Région pour visualiser la vidéo avec des boutons pour lire la vidéo, la stopper, créer un nouveau segment, etc.

b) Région pour afficher l'ensemble des segments diapositives construits.

Figure 36. *Interface principale de l'outil OntoCoV.*

Une petite fenêtre indépendante s'affiche en cliquant sur un segment de la zone (b). Elle contient une liste des objets pédagogiques constituant le segment sélectionné. On peut décrire chaque objet pédagogique en lui associant une liste des concepts de l'ontologie de domaine d'enseignement intégré dans l'outil au tout début.

Après l'annotation d'une leçon vidéo du cours '*structure de donnée et technique de programmation*', l'outil génèrera le code OWL suivant :

```
<cours_video rdf:ID="structure_de_données">
 <est_presente_dans>
   <lesson_video rdf:ID="fonction">
    <URL rdf:datatype="http://www.w3...#string"> http://.../fonction.wmv </URL>
    <est_segmente rdf:resource="#diapositif_2"/>
    <est_segmente rdf:resource="#diapositif_3"/>
    <est_segmente rdf:resource="#diapositif_7"/>
   </lesson_video>
 </est_presente_dans>……
</cours_video>
<diapositive rdf:ID="diapositive_2">
  <Duree rdf:datatype="&xsd#time">00:03:22 </Duree>
  <Debut rdf:datatype="&xsd#time">00:06:01 </Debut>
  <Titre_diapositive rdf:datatype="&xsd#string"> introduction aux fonctions
  </Titre_diapositive>
  <contient>
     <OP rdf:ID="Definition_1"> <concerne rdf:resource="&p1#fonction"/> </OP>
  </contient>
  <contient>
    <OP rdf:ID="definition_2">
        <concerne rdf:resource="&p1#adresse"/>
        <concerne rdf:resource="&p1#fonction"/>
    </OP>
  </contient>
  <contient>
    <OP rdf:ID="exemple_1">
```

```
        <concerne rdf:resource="&p1#valeur_retournee"/>
        <rdfs:comment rdf:datatype="&xsd#string"> différents type de valeurs retournées
        </rdfs:comment>
    </OP>
  </contient>
</diapositif>
…
```

4.4 Indexation Conceptuelle

Une fois que les concepts des deux ontologies ont été identifiés dans les segments temporels, on passe à la phase de pondération des concepts pour chaque segment diapositive.

Dans cette partie de notre travail, nous présentons une structure d'index qui permet de poser des requêtes sur des segments temporels de document vidéo. Pour cela nous utilisons le modèle vectoriel de Salton (Salton, 1983) tout en adaptant le calcul du poids TF_IDF (Terme Frequency_Inverse document Frequency) à nos besoins, en nous inspirant des travaux de (Martinet, 2004) et de (Zargayouna, 2005). On propose que le segment d'un document correspondant à une leçon vidéo soit représenté par un vecteur de concepts.

4.4.1 Structure de la table d'index

La structure d'index conçu est illustrée par la figure suivante. La table d'index est une matrice (tableau de deux dimension) tel que : Table-Index [concept (c), liste-leçon-Vidéo]. *Concept : représente une clé ou index. List des leçons Vidéo (LVi): c'est le vecteur de la table.*

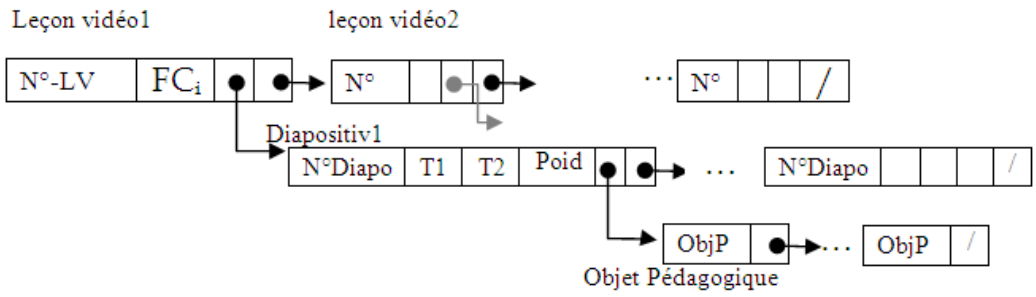
Index	Liste des objets leçon-vidéo
Concept C_i	
.	…

Table 3. *Structure de la table d'index*

- N° LV : numéro de la leçon vidéo.
- FC_i : la fréquence du concept C_i dans la leçon vidéo1.
- N°Diap: numéro de la diapositive qui contienne le concept C_i. C'est le segment qui correspond à la balise <diapositive> dans le fichier OWL. Ce dernier correspond à l'annotation sémantique de la leçon vidéo effective.
- T_1 et T_2 : début et durée de la diapositive ;
- Poids est calculé par la formule CF-ISDF (voir ci-dessous).
- ObjP : Objet pédagogique (i) qui contient dans sa description le concept C_i.

4.4.2 Pondération des concepts dans les segments temporels

Puisque les segments sont décrits par des concepts au lieu de termes, on calcule le poids des concepts par rapport aux segments dans lesquels les concepts apparaissent. Ainsi nous définissons la formule CF_ISDF (Concept Frequency_Inverse Segment and Document Frequency), de la manière suivante:

$$CF-ISDF(c,s,d) = CF(c,s,d).ISF(c,s,d).IDF(c,d)$$
$$ISF(c,s,d) = \log\left(\frac{S_d}{SegF(c,s)}\right) \quad (1)$$
$$IDF(c,s) = \log\left(\frac{D}{DF(c)}\right)$$

- $CF(c,s,d)$ *: Le nombre d'occurrences du concept c dans le segment s du document d.*
- *D : l'ensemble de tous les documents (leçons vidéo) du corpus.*
- S_d*: Le nombre de segments dans le document d.*
- $SegF(c,s)$ *: Nombre de segments dans le document d et dans lequel le concept c apparaît.*
- $DF(c)$ *: Nombre de documents qui contiennent le concept.*

Cette formule nous permet de pondérer le concept non seulement par sa fréquence dans le segment **s** relative à un document **d**, mais aussi par sa répartition dans le document (ISF(c,s,d)). Cette dernière mesure représente la force discriminatoire d'un concept c pour un segment relatif à un document d. La répartition du concept dans le corpus (IDF(c)) est importante aussi. Si un concept apparait dans plusieurs documents, il est moins représentatif pour un document donné qu'un concept qui apparait seulement dans ce document. C'est la force discriminatoire d'un concept dans le corpus.

4.5 Recherche Conceptuelle

Nous arrivons maintenant à la phase de recherche qui s'appelle aussi la phase d'interrogation. Elle comprend :

- la formulation du besoin d'information à travers une requête,
- la traduction de la requête dans une représentation interne défini par un modèle de requête,
- la comparaison de la requête aux index des documents du corpus par la fonction de correspondance
- la présentation des résultats en ordre de pertinence.

Formulation du besoin : Le corpus de cours vidéos englobe plusieurs matières ou domaines d'enseignement DE où à chaque DE est associée une ontologie selon le modèle conçu ci-dessus (voir 4.2.1). Ces ontologies, qui ont été utilisé pendant la phase d'annotation, serons utilisées pour aider l'utilisateur à formuler sa requête. Notre système offre une interface de visualisation et d'exploration d'une ontologie d'un DE particulier (choisi par l'usager) pour guider l'usager à naviguer dans l'arborescence de cette ontologie, en lui donnant l'occasion de choisir les concepts de sa requête. La raison qui nous a poussés à opter pour un tel mode de formulation de requête est double :

- l'utilisateur a parfois du mal à spécifier son besoin et à l'exprimer
- il faut lever les ambiguïtés et améliorer ainsi la précision et le rappel de notre système.

Modèle de requête : Un vecteur lui est associé comme pour les segments indexés. On peut attribuer un poids aux concepts de la requête. Dans un premier temps, nous attribuons des 1 et des 0 pour signifier l'existence ou non des concepts.

Fonction de correspondance : nous adaptons une mesure de pertinence classique du modèle vectoriel. La pertinence d'une requête Q par rapport à un segment S du document D sera :

$$pertinence(S_D, Q) = \cos(V_{S,D}, V_Q)$$

$V_{S,D}$ et V_Q sont les vecteurs des poids de concepts dans le segment S du document D et de la requête Q respectivement.

Les résultats de la recherche : est une liste des références vers les segments les plus pertinents visualisés en ordre de pertinence dans une page codé en HTML+time, d'où l'usager peut consulter et lire un segment vidéo sélectionné dans la même page.

4.6 Raisonnement

Pour améliorer la pertinence de la recherche, nous utilisons l'inférence sémantique dans la recherche de contenu. Les résultats retournées par la recherche conceptuelle peuvent être compléter par des résultats ou assertions implicites dérivées ou inférées à partir de la base de connaissance (ontologies) en exploitant les relations sémantiques qui existent entre les concepts (ex. la transitivité, la similarité, etc.).

Par exemple, si une diapositive expose un exemple qui montre l'expression ou la syntaxe de l'*instruction de répétition* en C, et que la requête contient le concept *boucle*, le système doit déduire que ce diapositive contribue lui aussi aux réponses pertinentes.

Dans le cas où l'utilisateur cherche des exercices concernant un ou plusieurs concepts avec leurs solutions correspondantes, le système doit retrouver et associer chaque diapositive contenant l'énoncé de l'exercice avec la ou les diapositives contenant ses différentes solutions.

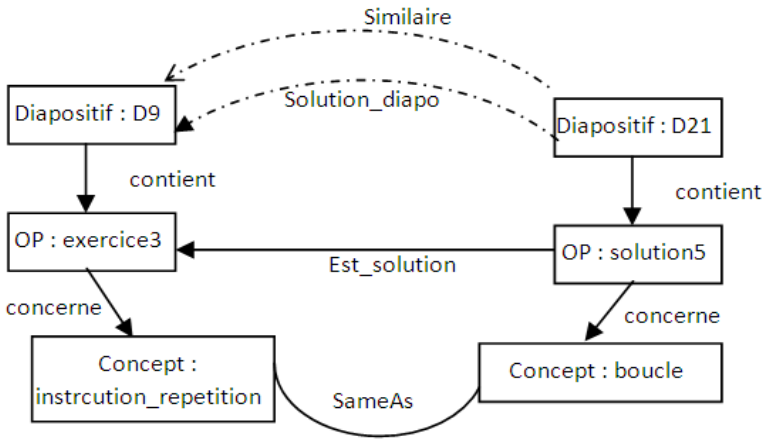


Figure 37. *Des assertions explicites complétées par des connaissances inférées*

Les arcs en pointillés dans la figure 37 représentent les assertions (instances de relation) inférées. On peut citer un autre exemple : notre système doit retrouver des diapositives qui présentent des explications sur des concepts prérequis de celui cité dans la requête.

Pour réaliser ce raisonnement et bénéficier des pleines possibilités des techniques du web sémantique, nous devons étudier les manières d'encoder les connaissances de raisonnement comme les règles de production, et de les articler avec la gestion des ontologies utilisées.

Le langage OWL-DL permet de spécifier des propriétés algébriques pour les relations. Ceci est grandement utile, mais il manque de possibilités pour encoder des connaissances qui soient relatives à la composition de plusieurs relations en particulier. Nous avons par exemple besoin de définir les règles suivantes:

Règle1 : *Contient(?D1, ?OP1) ET concerne(?OP1,?C1) ET SameAs(?C1,?C2) ET concerne(?OP2,C2) ET contient(?D2,OP2)* → *similaire (D1,D2).*

Règle2 : *Contient(?D9, OP :excercie3) ET Est_solution(OP: solution5, OP :exercice3) ET contient(?D21 ,OP :solution5)* → *solution_diapo(?D21, ?D9).*

La première règle énonce que :

> Si (l'instance D1 du concept diapositive contient un objet pédagogique OP1 qui concerne le concept C2, les deux concepts C1 et C2 sont similaire et le concept C2 est concerné par l'OP2 contenant dans la diapositive D2)
>
> Alors D1 et D2 sont similaires.

Ces règles sont reconnues dans la communauté du web sémantique et commencent à être traitées par des langages et des outils appropriés. Ainsi, SWRL (SWRL, 2004) constitue un rapprochement entre les langages OWL et les règles logiques. Il combine OWL-DL avec le Rule Markup Langage (RuleML).

Nous avons choisi SWRL car il nous permet de spécifier nos règles pour raisonner sur les individus OWL et d'inférer de nouvelles connaissances sur ces individus. Il porte une syntaxe XML basée sur celle d'OWL, ce qui nous permet d'inclure les règles SWRL associées à notre ontologie dans le même fichier XML.

Nous résumons la conception de notre deuxième approche par un schéma fonctionnel (voir figure 38) de notre système que nous baptisons IRSeCoV (Indexation et Recherche Sémantique dans des Cours Vidéo). Ce système a pour vocation de permettre une recherche plus précise et plus pertinente de segments vidéo pédagogiques. Il possède plusieurs composantes, ce qui lui permet d'être modulaire.

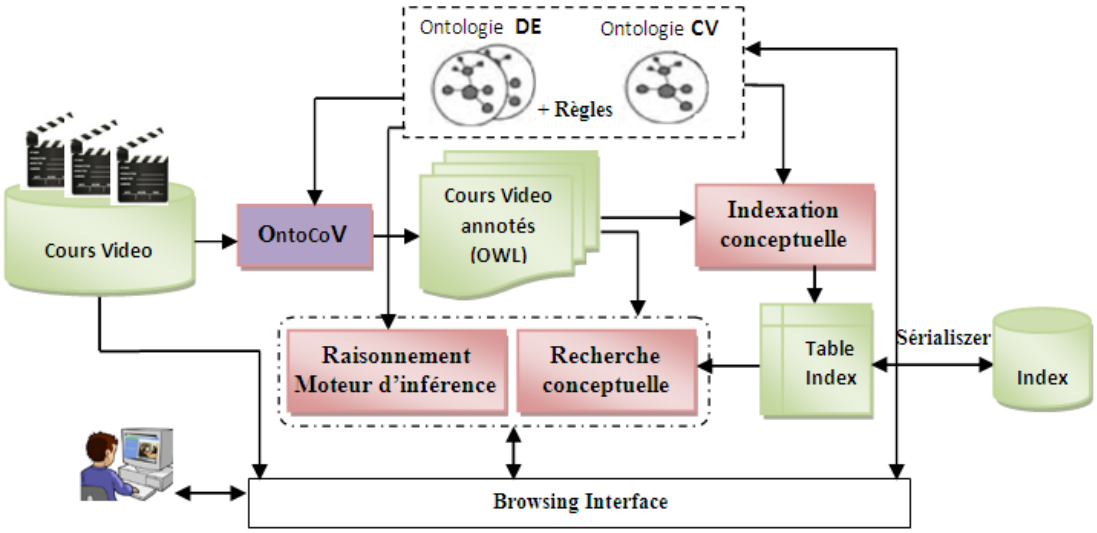


Figure 38. *Architecture générale du système IRSeCoV.*

Nous mentionnons que le module « Raisonnement » est implémenté indépendamment du système pour des fins d'évaluation (voir le point ''raisonnement avec moteur d'inférence'' dans 4.7.3), mais nous pensons l'intégrer par la suite. Donc, exception du module Raisonnement, la structure interne du système est composée principalement de sept classes, où chacune d'elle est décrite par ses attributs et ses opérations dans le diagramme de classes illustré par la figure 39. Les utilisateurs n'interagissent qu'avec la classe MainWindow et c'est à travers elle que les utilisateurs peuvent manipuler les autres classes.

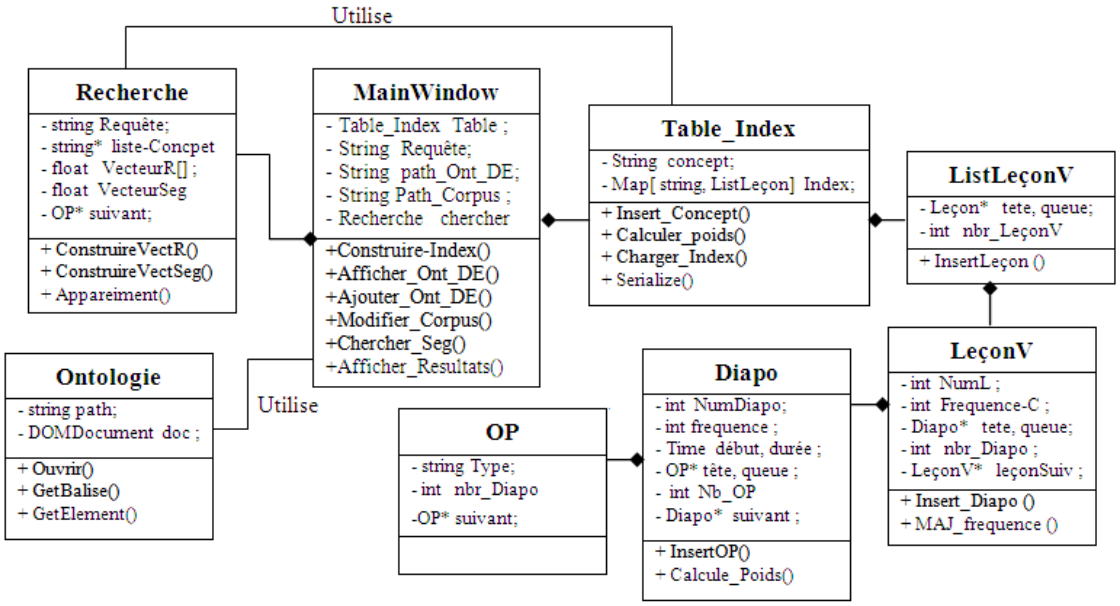


Figure 39. *Structure interne du système IRSeCoV (Diagramme de classes)*

4.7 Prototype et Expérimentation

4.7.1 Implémentation

Nous avons implémenté un prototype pour le système IRSeCoV avec le langage de programmation orienté objet C++ qui se base sur le Framework Qt.

L'interface du système est divisée en deux grandes régions : la première qui est à gauche, est réservée à la formulation de la requête. La deuxième est réservée à l'affichage des résultats (voir figure 40). La première région visualise les concepts d'une ontologie d'un DE particulier (sélectionné par l'usager) sous forme d'arborescence pour guider l'usager (apprenant ou enseignant) à naviguer (explorer) dedans, tout en lui donnant l'occasion de choisir les termes de sa requête. La raison qui nous a poussés à opter pour un tel mode de formulation de requête est d'éviter l'ambiguïté dans les termes saisis et de bien pouvoir exprimer les besoins.

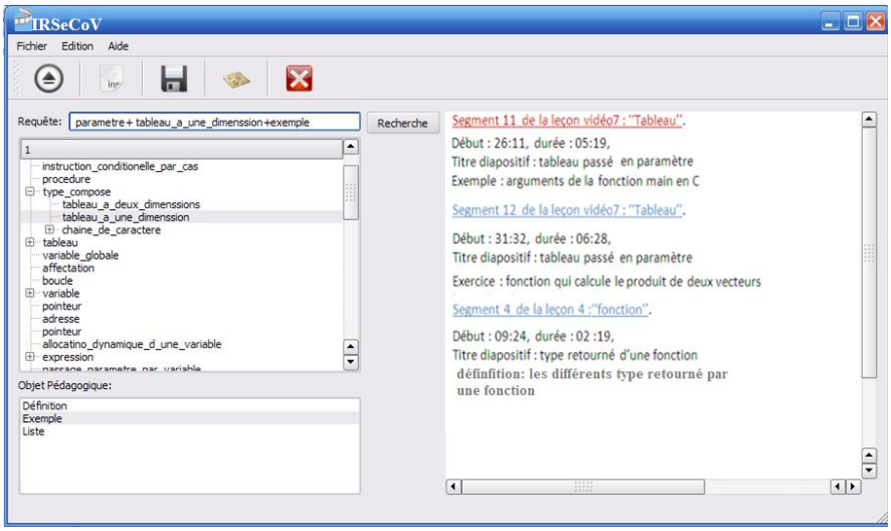


Figure 40. *Interface du système IRSeCoV.*

L'utilisateur peut spécifier dans sa requête l'objet pédagogique avec les concepts qu'il cherche. Le système retourne et affiche dans la région 2 une liste des segments triés par ordre de pertinence. Pour chaque segment, il affiche le nom de la leçon, son début, sa durée, son titre et, surtout, les objets pédagogiques inclus dans le segment avec un commentaire. L'utilisateur peut ainsi sélectionner le segment qu'il veut.

Le système donne la possibilité aux enseignants d'ajouter de nouveaux cours vidéo avec leurs annotations y compris l'ontologie DE. Ces cours peuvent correspondre à un

domaine d'enseignement déjà existant ou à un domaine nouveau. Donc le système régénère la table d'index automatiquement.

4.7.2 Expérimentation et évaluation

Nous essayons d'analyser les résultats du module Recherche conceptuelle, puis celui de raisonnement avec moteur d'inférence.

Pour évaluer notre système, on a deux approches à savoir évaluer la structure de l'index et évaluer sa pertinence. Dans la première approche, il s'agit de calculer le temps d'indexation, l'espace de stockage de l'index par rapport à la taille du corpus, la construction des ontologies et l'annotation du corpus. Le calcul du temps de construction de l'index ne permet pas de juger de la valeur de l'index. C'est pour cette raison que nous avons opté pour la deuxième approche d'évaluation, qui permet d'évaluer la pertinence de l'index en testant son impact sur la recherche, ceci en utilisant les mesures de pertinence classiques (rappel et précision). Mais comme le corpus est trop petit (seulement 9 documents vidéo), nous avons essayé d'interpréter les résultats de quelques requêtes. Dans un premier temps nous avons testé les poids de quelques concepts extraits de la table d'index générée par le système.

L'expérimentation de notre système a été faite sur un corpus (annoté par OntoCoV) de 9 leçons vidéo parmi 25 du module 's*tructure de donnée et techniques de programmation*'. Le tableau 4 montre quelques concepts avec la liste des segments où ils apparaissent. Le segment est défini par le numéro du document (leçon vidéo) et le numéro du segment (diapositive) dans ce document.

Concepts	Liste des segments retournés
Pointeur	{(D_1,S_7), (D_4,S_1), (D_5,S_5), (D_5,S_{12}), (D_6,S_2), (D_8,S_9), (D_8,S_{14}), (D_9,S_4)}
Parametre_formel	{(D_4,S_2)}

Table 4. *Liste des concepts avec les segments associés.*

CF-ISDF(Pointeur,S_2,D_6) = 1* log(**13/2**) * log(9/6) = 0,7589.

CF-ISDF(Pointeur,S_9,D_8) = **6*** log(**16/11**) * log(9/6) = 0,9110.

CF-ISDF(Pointeur,S_{14},D_8)= **1*** log(16/11) * log(9/6) = 0,3038.

CF-ISDF(Parametre_f,S_2,D_4) = 1*log(14/2)*log(9/**1**)= 4,2756.

- Si un concept apparait dans 2 segments de la même leçon, c'est sa fréquence qui détermine le segment le plus pertinent. Voir le concept 'pointeur' dans les deux segments (S9, D8) et (S14, D8).
- Si un concept apparait dans des segments de différentes leçons, c'est la force discriminatoire ISF qui détermine le segment le plus pertinent. Voir le concept 'pointeur' dans les deux segments (S2, D6) et (S14, D8).
- Si un concept apparait dans un segment ou deux au plus par rapport au corpus, il aura un poids très grand à cause de la valeur discriminatoire IDF. Comme le cas du concept parametre_formel.

Nous proposons à élargir l'expérimentation de notre système sur un corpus de cours vidéo pour différents domaines d'enseignement afin d'évaluer sa pertinence en calculant le rappel et la précision.

Raisonnement et inférence

Cette partie a été réalisée indépendamment du système IRSeCoV. Rappelant que nous avons construits deux ontologies à savoir '*domaine d'enseignement*' et '*cours vidéo*'. La première a été construite en utilisant l'outil protégé 2000, et son code OWL généré a été importé dans la deuxième ontologie. L'ontologie résultante de cette combinaison a été utilisé par l'outil OntoCov (voir 4.3) afin d'annoter un corpus de 9 leçons vidéo du domaine d'enseignement '*structure de données et algorithmique*'. L'outil OntoCoV génère un code OWL pour chaque leçon. Donc le corpus proprement traité est un ensemble de fichiers (.owl) qui font références aux fichiers vidéo (.wmv) correspondant.

Pour faire un raisonnement sur ce corpus et obtenir des résultats implicites et complémentaires à ceux obtenus par la méthode vectorielle adoptée, nous avons procédé de la manière suivante :

- Rassembler le code OWL (plus précisément les instances) de toutes les leçons dans un seul fichier 'ontologie_totale.owl'.
- Ouvrir ce dernier avec l'outil protégé 2000 version 3.4.8 et faire la vérification et la validation de l'ontologie en question. Les problèmes de cohérence, de satisfiabilité, de subsomption sont alors vérifiés grâce à la connexion au moteur

d'inférence Pellet 1.5.2. La figure 41 montre la vérification de l'inconsistance de la première ontologie avant son intégration dans la deuxième.

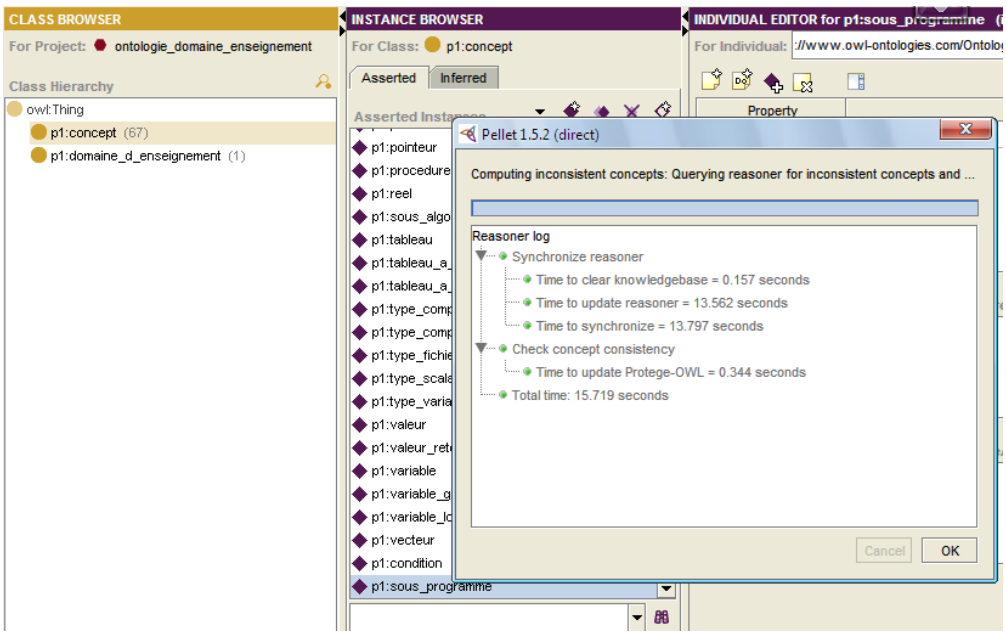


Figure 41. *Vérification de l'ontologie 'domaine d'enseignement'.*

La figure 42 illustre l'ontologie totale sous forme d'un graphe en utilisant le plugin jambalaya.

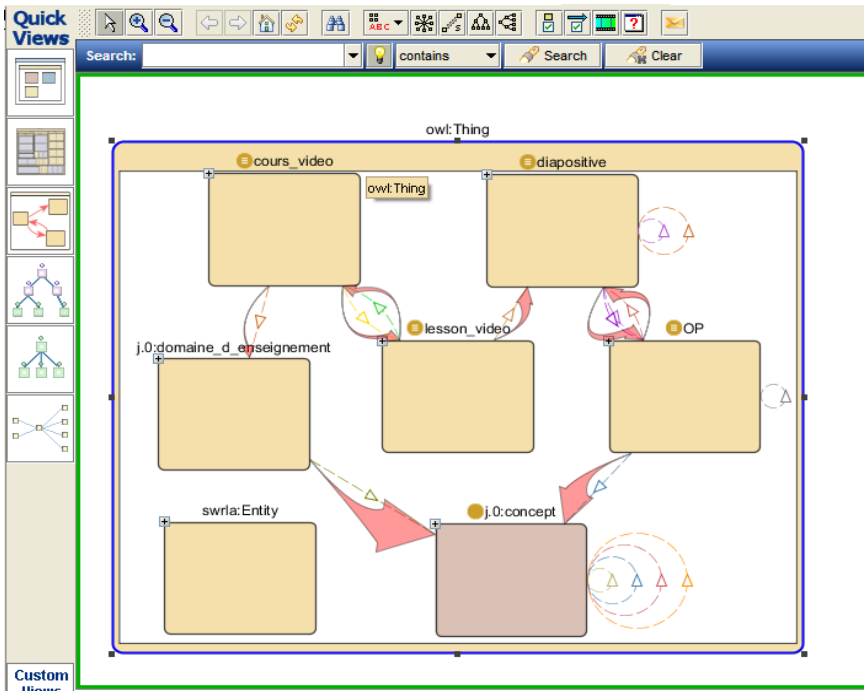


Figure 42. *Graphe de l'ontologie totale avec le plugin Jambalya.*

- Spécifier les règles (que nous avons définies dans le point 4.6) dans l'ontologie totale avec le langage SWRL. Nous avons utilisé le plugin **SWRLTab** pour l'édition de ces règles.

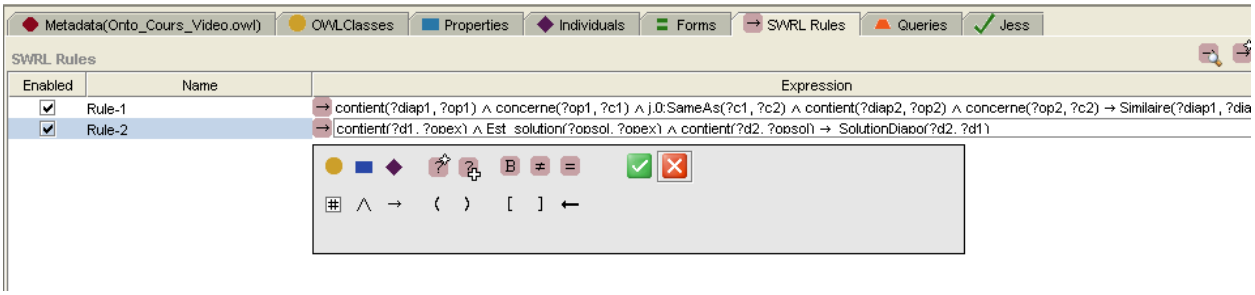


Figure 43. *Edition des règles SWRL.*

- Une fois l'ontologie validée, elle est donc prête à être interrogée par les requêtes des utilisateurs et inférés des résultats implicites et complémentaires. Nous avons en premier temps interrogé l'ontologie par une question exprimée dans langage SPARQL. La requête que nous avons choisi est comme suit : *quelles sont les segments temporelles (ou diapositive) qui montre l'utilisation d'un* ***tableau*** *comme* ***paramètre*** *dans une* ***fonction***. Voir le résultat de cette requête dans la figure 44.

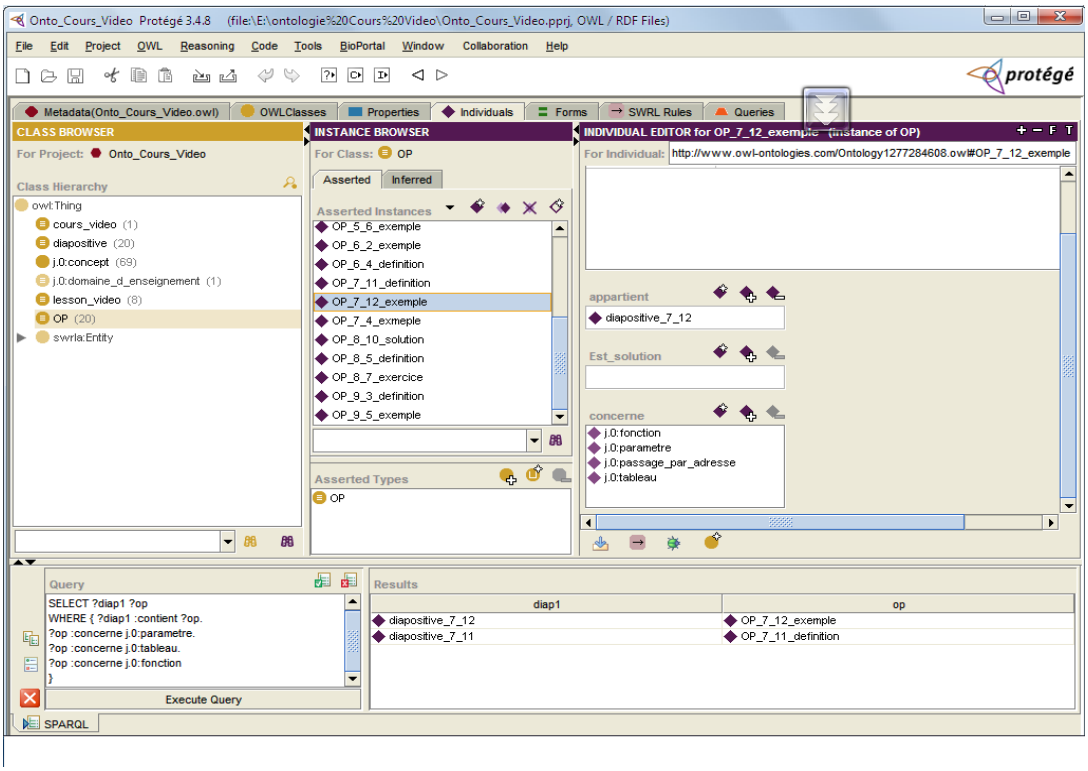


Figure 44. *Requête en SPARQL et son résultat.*

- Inférence proprement dite : Afin de traiter les règles, nous avons utilisé le moteur spécifiquement dédiés aux règles, tel que le moteur **Jess11**. Ce moteur de règles possède un langage propre pour l'expression des connaissances sous forme de règles. Il peut être utilisé depuis Protégé grâce à l'existence d'un pont qui permet de traduire un modèle d'ontologie dans le langage de Jess, d'exécuter les règles dans Jess et finalement de récupérer le résultat dans Protégé. Nous avons utilisé le plugin **SWRLJessTab** qui s'intègre dans l'onglet de SWRL Rules sous forme d'icone J en haut à droite (voir figure 45). Pour lancer le moteur, nous activons les 3 boutons en succession '*OWL+SWRL-> Jess*', '*Run Jess*' et '*Jess -> OWL*'. Ensuite, nous pouvons voir les résultats inférés dans l'onglet '*Inffered Axioms*'.

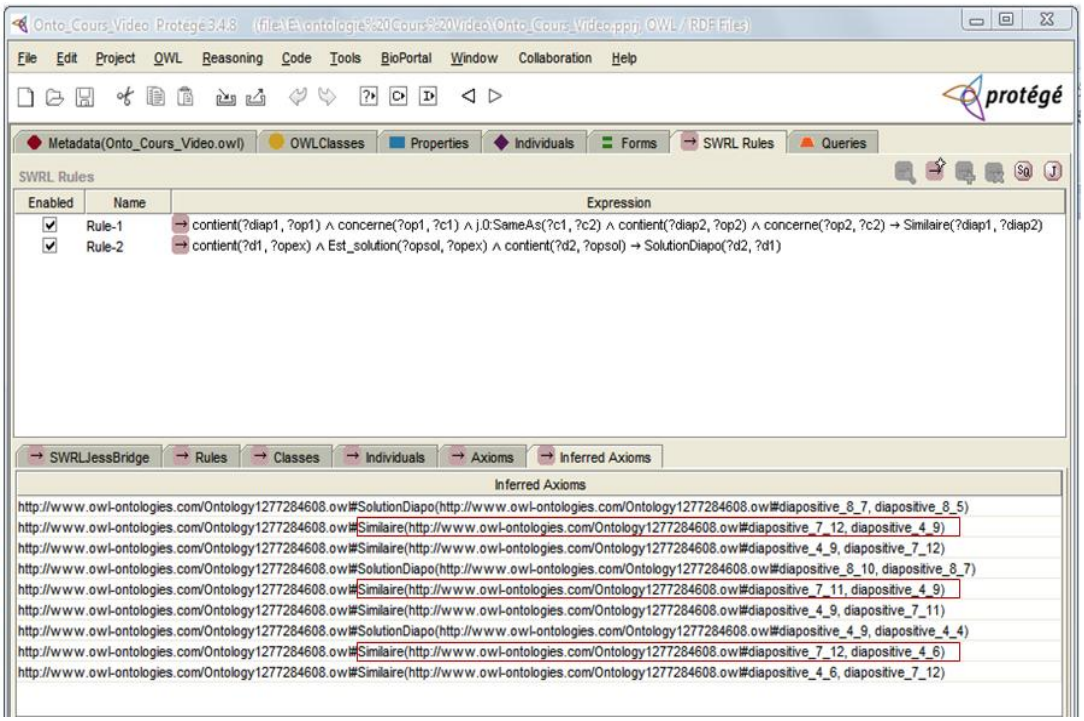


Figure 45. *Résultats inférés par le moteur d'inférence Jess*

Nous remarquons que le moteur exécute la règle 1 et il déduit que la diapositive 12 de la leçon 7 '*diapositive_7_12*' est similaire au *diapositive_4_9* en exploitant la similarité des deux concepts *'argument'* et *'paramètre'* qui se trouvent chacun dans une diapositive. Même chose pour les deux diapositives 7_11 et 4_9.

4.8 Conclusion

Nous avons présenté dans ce chapitre une approche de recherche par le contenu sémantique des segments vidéo pédagogiques utilisant des ontologies. Nous avons construit deux ontologies, la première modélise les connaissances d'un domaine d'enseignement et la deuxième structure un cours vidéo pédagogiquement.

Nous avons réalisé un nouvel outil appelé OntoCoV qui génère l'annotation de leçons vidéo en OWL basé sur les ontologies construites.

Ensuite, nous avons détaillé l'indexation et la recherche conceptuelle sur un ensemble de leçons vidéo annotées et ceci en adaptant la méthode vectorielle. Nous avons défini une nouvelle formule CF-ISDF pour calculer le poids d'un concept dans un segment vidéo. Pour mettre en œuvre cette approche, nous avons développé le prototype IRSeCoV. Nous avons essayé d'évaluer la pertinence de l'index en testant son impact sur la recherche en utilisant les mesures de pertinence classiques (rappel et précision). Mais comme le corpus est trop petit (9 documents vidéo), nous avons essayés d'interpréter les poids de quelques concepts extrait de la table d'index généré par le système.

Nous avons essayé aussi d'améliorer les résultats obtenus par des résultats complémentaires et pertinents en exploitant les relations sémantiques qui existent entre les concepts et entre les objets pédagogiques définis dans l'ontologie. Pour ce faire, nous avons utilisé protégé 2000 avec les plugins : SWRLTab pour éditer les règles SWRL et SWRLJessTab pour exécuter ces règles et inférer de nouvelles réponses.

Les résultats obtenus montrent la faisabilité et les avantages de l'utilisation des ontologies pour la recherche par le contenu sémantique des segments vidéo pédagogique.

Conclusion générale et perspectives

Bilan des travaux et apports de la thèse

Notre axe de recherche s'articule sur la création et l'indexation sémantique d'un contenu pédagogique multimédia. Ce contenu a pour supports des documents ayant la dimension temporelle, en leur offrant une caractéristique d'être des séquences vidéo multimédia.

Avec l'approche du double codage, nous avons présenté un apport, que nous considérons très important, dans l'amélioration de l'apprentissage. Pour cela, nous avons conçu un modèle de document multimédia temporisé qui concrétise cette approche. Nous avons développé un système auteur appelé SACoPh, qui génère des cours de phonétique publiables via le web, selon ce modèle, en exploitant les nouvelles fonctionnalités de la version 3.0 du standard SMIL. Ce système est conçu pour être les enseignants de langues, qui ont peu de connaissances en ce qui concerne les outils informatique. Il offre une interface qui soit la plus simple et la plus conviviale possible.

Nous avons présenté également dans cette thèse, une approche de recherche par le contenu sémantique des segments vidéo pédagogiques utilisant les ontologies. Nous avons construit deux ontologies, la première structure pédagogiquement un cours vidéo et la deuxième modélise les connaissances d'un domaine d'enseignement.

Nous avons réalisé un nouvel outil appelé OntoCoV qui génère l'annotation de leçon vidéo en OWL basé sur les ontologies construites.

Ensuite, nous avons détaillé l'indexation et la recherche conceptuelle de l'ensemble des leçons vidéo annotées en adaptant la méthode vectorielle. Nous avons défini une nouvelle formule CF-ISDF pour calculer le poids d'un concept dans un segment vidéo. Pour mettre en œuvre cette approche, nous avons développé un prototype baptisé IRSeCoV. Nous avons essayé d'évaluer la pertinence de l'index en testant son impact sur la recherche en utilisant les mesures de pertinence classiques (rappel et précision).

Mais comme le corpus est trop petit (9 documents vidéo), nous avons essayé d'interpréter les poids de seulement quelques concepts extraits de la table d'index généré par le système.

Nous avons essayé aussi d'améliorer les résultats obtenus avec des résultats complémentaires et pertinents en exploitant les relations sémantiques qui existent entre les concepts et entre les objets pédagogiques définis dans l'ontologie. Pour ce faire, nous avons utilisé protégé 2000 avec les plugins : SWRLTab pour éditer les règles SWRL et SWRLJessTab pour exécuter ces règles et pour inférer de nouvelles réponses.

Les résultats obtenus montrent la faisabilité et les avantages de l'utilisation des ontologies pour la recherche par le contenu sémantique des segments vidéo pédagogiques.

Perspectives

Il est néanmoins important de noter que notre approche est loin d'être finie et qu'elle doit évoluer (nous l'espérons dans un futur proche).

- Pour ce qui concerne la production du contenu pédagogique multimédia, le système SACoPh permet la création de ce contenu selon un modèle figé. Donc, nous suggérons d'améliorer le système pour être **générique**. C'est-à-dire, qu'il doit donner la possibilité de créer plusieurs modèles de contenus pédagogiques multimédia (chacun dédié à un domaine spécifique) en spécifiant leurs quatre dimensions (structurelle, spatiale, temporelle et navigation temporelle) de manière graphique. Une fois que l'enseignant ait sélectionné un modèle, l'interface du système auteur doit s'adapter au modèle choisi en offrant des outils de création des différents objets média constituant le modèle, chacun dans sa région affectée. Ceci pour toujours préserver le critère WYSIWYG d'un côté et pour que, d'un autre côté, l'enseignant ne soit pas obligé, lors de l'élaboration de son cours, d'utiliser plusieurs outils externes pour préparer les différents types de médias qui veut les intégré dans son cours. Dans le cas où il veut utiliser un média déjà existant, il pourra juste importer son url. Il faudrait aussi ne pas

oublier de semi-automatiser la synchronisation des objets média dans le cas où les médias sont importés.

- Pour avoir un contenu pédagogique multimédia sous forme d'un objet pédagogique **réutilisable** dans n'importe quelle plate-forme d'enseignement, le système doit offrir la possibilité de générer le contenu créé dans le standard **SCORM**.

- Notre système de recherche des segments vidéo pédagogiques est une application monoposte ou off-line. On propose d'élargir son utilisation sur le web par la réalisation d'un moteur de recherche on-line qui permette une recherche sémantique des segments temporels, sur tout cours audiovisuel ou ayant le format SMIL. Ceci doit se faire tout en intégrant l'inférence dans le système en utilisant les bibliothèques logiciels appelées API qui traitent les ontologies OWL et les règles SWRL.

- En ce qui concerne les ontologies, nous proposons de réaliser des outils qui automatisent l'instanciation de l'ontologie de 'domaine d'enseignement' à partir d'un support de cours ayant le format texte. Cet outil doit utiliser les techniques TAL (traitements automatiques de langue). Nous suggérons aussi d'étendre le modèle ontologique en intégrant des connaissances sur le profil des apprenants afin d'orienter notre système vers l'adaptation des segments vidéos selon ses profils.

Bibliographie

Allen J. F., « Maintaining Knowledge about Temporal Intervals » Communications of the ACM (Association for Computing Machinery), Volume 26 Issue 11, Nov. 1983, pages: 832-843, ISSN:0001-0782.

Ambulant, "Ambulant open SMIL player" www.ambulantplayer.org, 10 mars 2010.

Atif Y., titre « A constructive multimedia approach to e-learning » Emirates Journal for Engineering Research, 8 (1), 25-33 (2003).

Beck A. et al, « SWANS, un système auteur de synchronisation et d'annotation pour un apprentissage multimodale de phénomène accentuels en langue vivante L2 » Environnements Informatiques pour l'Apprentissage Humain, Montpellier 2005.

Behaz A. et Djoudi M., "Contribution de génération d'un hypermédia d'enseignement adaptatif à base d'ontologies". 3es Journées Francophones sur les Ontologies JFO Poitiers France, 3-4 Décembre 2009.

Benayache A., « construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : le projet MEMORAe ", Thèse pour l'obtention du grade de Docteur de l'UTC. Soutenu le 5 décembre 2005.

Blandine M., Wu Y. et Kuang R., « la robustesse des systèmes auteurs multimédias », thèse de doctorat, Université de Paris VIII spécialité : informatique.

Bousbia N. et Balla A., Scénarisation des contenus pédagogiques destinés à la FOAD. Conférence Internationale sur l'Informatique et ses Applications, CIIA06, 15-16 Mai 2006, Saida, Algérie

Bousbia N., « Contribution théorique et méthodologique à l'élaboration d'un environnement de FOAD » Mémoire de magister, Institut National d'information INI, option système d'information, Alger, 2005.

Bouzeghoub A., Defude B., Duitama J. F. et Lecocq C. «Un modèle de description sémantique de ressources pédagogiques basé sur une ontologie de domaine» revue sticef 'Sciences et

Technologies de l´Information et de la Communication pour l´Éducation et la Formation' Article de recherche Volume 12, 2005.

Buchanam C. et Zellweger P.T., «Specifying Temporal Behavior in Hypermedia Documents », Proc. of the ACM Conf. on Hypertext, pp. 262-271, December 1992.

Buffa M., Dehors S., Faron-Zucker C. and Sander P., « Towards a Corporate Semantic Web Approach in Designing Learning Systems". Review of the TRIAL Solution Project", workshop conference AIED 2005.

Bulterman D. and Hardman L., « Structured Multimedia Authoring », in Transactions on Multimedia Computing, Communications and Applications. 1(1). February (2005). New York: ACM Press. pp. 89–109.

Carbonaro A., "Ontology-based Video Retrieval in a Semantic-based Learning Environment", Journal of e-Learning and Knowledge Society. Vol. 4, n. 3, September 2008 (pp. 203 - 212).

Charhad M., « Modèles de Documents Vidéo basés sur le Formalisme des Graphes Conceptuels pour l'Indexation et la Recherche par le Contenu Sémantique » thèse de doctorat de l'université Joseph Fourier- Grenoble I soutenue le 28 novembre 2005.

Charhad M., Zrigui M. et Guénot G. « Une approche conceptuelle pour la modélisation et la structuration sémantique des documents vidéos », SETIT 2005, 3rd international conférence : Sciences of Electronics, Technologies of information and Telecommunications, March 27-31, 2005- Tunisia.

Chehata N., « Indexation vidéo orientée objet : Combinaison des techniques de segmentation d'images en régions avec l'extraction des descripteurs de couleur » mémoire de stage de DEA, PHOTONIQUE, IMAGE ET CYBERNETIQUE. Université Louis Pasteur Strasbourg. THOMSON MULTIMEDIA R&D France.

Chisogne C., mémoire de fin d'étude : « Système de contraintes temporelles appliquées à l'édition de documents multimédia structurés », éditeur FUNDP, Institut d'informatique, Lundi le 30 août 1999.

DCMI Dublin Core Metadata Initiative, from http://dublincore.org/documents/dcmi-terms/

Deltour R., Guerraz A. and Roisin C., « Multimedia Authoring for CoPs ». 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, Crete, Greece, 2 October 2006, pp. 60-69

Desmoulins C. et Grandbastien M., « Des ontologies pour indexer des documents techniques pour la formation professionnelle » conférence : IC'2000 Ingénierie des connaissances. Toulouse (Centre pour l'UNESCO), 10-12 Mai 2000.

Dong A., Li H. and Wang B., "Ontology-driven annotation and Access of Educational Video Data in E-learning", in E-learning Experiences and Future, Edited by: Safeeullah Soomro, Publisher: InTech, (pp. 305-326, April 2010, ISBN 978-953-307-092-6).

Euzenat J., Layaïda N. et Dias V., "A semantic framework for multimedia document adaptation" In Proceedings of the 18th International Joint Conference on Artificial Intelligence, pages 31–36, 2003.

Gruber T. R., "A Translation Approach to Portable Ontology Specifications". In Knowledge Acquisition, 5(2), pp. 199-220, 1993. See also "What is an Ontology?" http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

Haav H.M. and Lubi T.L., "A Survey of Concept-based Information Retrieval Tools on the Web", In Proceedings of the 5th East-European Conference ADBIS, Vol 2, pp 29-41, 2001.

Hammache A. et Ahmed-Ouamer R., « Un système de recherche d'information pour le e-learning» Revue Document numérique 1279-5127 - VOL 11/1-2 - 2008 - pp.85-105.

Hernandez N., « Ontologie de domaine pour la modélisation du contexte en recherche d'information » thèse de doctorat de l'université de Paul Sabatier de Toulouse, Soutenue le mardi 06 décembre 2006.

Hervieu A., « Analyse de trajectoires vidéos à l'aide de modélisations markoviennes pour l'interprétation de contenus » thèse de doctorat devant l'Université de Rennes 1, Mention : Traitement du Signal et Télécommunications, Soutenue le 5 mars 2009.

Horn A., "On sentences which are true of direct unions of algebras". Journal of Symbolic Logic, 16(1):14–21, 1951.

HyTime, ISO/IEC JTC1/SC18/WG8 N1920, Information Technology: Hypermedia/Time-based Structuring Language (HyTime), Second edition, ISO/IEC, August 1997. [En Ligne] : http://www.ornl.gov/sgml/wg8/docs/n1920/html/n1920.html.

Isaac A. et Troncy R. « Ontologie et description du contenu de document audiovisuel : une expérimentation dans le domaine médical » CDT 2005.

ISO/IEC 2002, Overview of the MPEG-7 Standard (version 8), ISO/IEC JTC1/SC29/WG11/N4980, Klagenfurt, July 2002.

Jedidi A., « Modélisation générique de document multimédia par des méta-données : mécanisme d'annotation et d'interrogation » thèse de doctorat soutenu à l'université Toulouse III – Paul Sabatier U.F.R Mathématique Informatique Gestion, le 06 juillet 2005.

Jourdan M., Layaïda N. et Roisin C., « Le temps dans les documents », vol. H 7 228, Techniques de l'ingénieur, 249, rue de Crimée, Paris, 1999.

Layaïda N., «MADEUS : Système d'édition et de présentation de documents multimédia structurés », thèse de doctorat. Université Joseph Fourier de Grenoble, France, 1997.

Layaïda N. et Sabry-Ismail L., « MADEUS : un modèle de document multimédia structuré », Techniques et Sciences de l'Informatique (TSI), vol. 15, num. 9, numéro thématique Multimédia Collecticiels, 1996.

LCM Ministère de la Jeunesse, de l'Education Nationale et de la Recherche (France), «Etude des outils de gestion de ressources numériques pour l'enseignement », Pôle conseil Business Interactif, 29/07/03

Lee J., Munther H. Abualkibash and Padmini K. Ramalingam, « Ontology- based Shot Indexing for Video Surveillance System» Proc. of CISSE 2007, Dec. 3 - 12, 2007.

Lin C-Y, Tseng B. L., and Smith J. R., "VideoAnnEx: IBM MPEG-7 Annotation Tool for Multimedia Indexing and Concept Learning," IEEE Intl. Conf. on Multimedia & Expo (ICME), Baltimore, July 2003.

Macromedia, Flash et Director, {En ligne : http://www.macromedia.com, 1998}.

Martin J., « Un modèle vectoriel relationnel de recherche d'information adapté aux images » thèse de doctorat de L'UNIVERSITÉ JOSEPH FOURIER – GRENOBLE I, soutenue le 22 décembre 2004.

Martin P., « WinPitch LTL, un logiciel multimédia d'enseignement de la prosodie », Alsic, Vol. 8, n° 2 | 2005, [En ligne], mis en ligne le 15 décembre 2005. URL : http://alsic.revues.org/index332.html. Consulté le 22 mars 2012.

Martinet J., « Un modèle vectoriel relationnel de recherche d'information adapté aux images» thèse de doctorat DE L'UNIVERSITÉ JOSEPH FOURIER – GRENOBLE I, Présentée et soutenue publiquement le 22 décembre 2004.

Mayer, R., Multimedia learning. Cambridge, MA, United States: Cambridge University Press (2001).

MCours.com
www.mcours.com
Cours, exercices & projets de fin d'études

ME, Livre « Multimedia and e-Learning: A New Direction for Productivity Promotion and Enhancement » ©APO 2003, ISBN: 92-833-2344-0, Report of the APO Seminar on Multimedia for Productivity Promotion and Enhancement (With Special Focus on e-Learning) Republic of China, 25–29 March 2002.

Merzougui G., Djoudi M. et Zidani A., "Editeur de cours médiatisés en SMIL", Conférence Internationale: Sciences Electroniques, Technologies de l'Information et des Télécommunications, IEEE SETIT 2004, ISBN 9973-41-902-2, Sousse, Tunisie, 15-20 Mars 2004.

MHEG, Meyer-Boudnik T. and Effelsberg W., « MHEG Explained », IEEE Multimedia Magazine, vol. 2, num. 1, pp. 26-38, 1995.

Mielnikoff M., « Qu'est-ce que l' E-Learning ? » CRITT-TTI ; Centre Régional d'innovation et de transfert de technologie- Télécommunication et technologies de l'information, le 07/11/2005.

MPEG-7, Marinez J. M., Koenen R. and Pereira F., « MPEG-7, The Generic Multimedia Content Description Standard, Part 1 », Editor: Peiya Liu Siemens Corporate Research, 1070-986X/02/$17.00 © 2002 IEEE.

Murray T., Blessing S. and Ainsworth S., "Authoring Tools for Advanced Technology Learning Environments" Dordrecht, Hardbound, Kluwer Academic Publishers: 571 p, 2003.

Najjar, L. J., "Dual Coding as A possible Explanation for the Effects of Multimedia on Learning". GIT-GVU-95-29. Atlanta: Georgia Institute of Technology, Graphics, Visualization and Usability Center, 1996.

Najjar, L. J. "Multimedia Information and Learning". Journal of Educational Multimedia and Hypermedia, 5, 129-150, 1998.

OWL, Smith M. K., Welty C. and McGuinness D. L., " OWL: Web Ontology Language Guide." W3C Recommendation, February 2004. Url : http://www.w3.org/TR/owl-guide/, consulter le 13/10/2012.

Péchou A. et Stenton A., «Encadrer la médiation- le cas de l'intonation » colloque compréhension et hypermédia, Albi, Octobre 2002.

Psyché V., Olavo et Bourdeau J. , « Apport de l'ingénierie ontologique aux environnements de formation à distance » revue sticef.org : science et technologie de l'information et de la communication pour l'Education et la formation, volume 10, 2003.

Pullum, Geoffrey K. *Phonetic symbol guide.* Chicago: University of Chicago Press. (1986)

RDF, Lassila O. et Swick R. R., «RDF: Resource Description Frame Work ». Recommandation 22 Février 1999 de W3C, Disponible à : http://www.w3.org/TR/REC-rdf-syntax/.

Reyes E. G., « L'objet technique hypermédia : repenser la création de contenu éducatif sur le Web » thèse de doctorat DE L'UNIVERSITÉ PARIS VIII, Discipline : Sciences de l'information et de la communication le 14 février 2007.

Roisin C., « Documents structurés multimédia » mémoire d'habilitation à diriger les recherches présentée le 22 septembre 1999 à l'institut national polytechnique de Grenoble.

Roisin C., Mikač J. and Le Duc B., "The LimSee3 Multimedia Authoring Model". ACM Symposium on Document Engineering, 10-13 October 2006, Amsterdam, the Netherlands, pp. 173-175

Roisin C. and Mikác J., « Comment bâtir un cours multimédia avec LimSee3 ? » EpiNet : Revue Electronique de l'EPI (2008)

Rosenberger T. and MacNeil R. L., «Prosodic Font: Translating speech into graphics» Proceedings of CHI'99 Extended Abstracts. http://www.media.mit.edu/~tara/CHI1999.pdf

Salton G., The SMART Retrieval System. Prentice Hall, 1971.

Salton G. and McGill M., « Introduction to Modern Information Retrieval», McGraw-Hill, 1983.

Sénac P., Diaz M., Léger A. and De Saqui-Sannes P., «Modeling Logical and Temporal Synchronization in Hypermedia Systems», IEEE Journal of Selected Areas on Communications, vol. 14, num. 1, pp. 84-103, 1996.

SMIL 1.0 «Synchronized Multimedia Integration Language (SMIL) 1.0 », P. Hoschka. Recommandation de W3C, 15 Juin 1998, on ligne : http://www.w3.org/TR/REC-smil.

SMIL 2.0 «Synchronized Multimedia Integration Language (SMIL 2,0) ». Recommandation de W3C, 2001. On ligne : http://www.w3.org/TR/smil20/.

SMIL 3.0 «Synchronized Multimedia Integration Language (SMIL 3.0)" Synchronized Multimedia Working Group (SYMM-WG) of the W3C; "http://www.w3.org/TR/SMIL/, visited on March,10 2010.

Stenton A., Péchou A., Caillant-Sirdey C. et Tricot A., «Effet du double cordage synchrone de l'accentuation en L2 selon des modalités de restitution de l'apprenant » 1er colloque international de didactique cognitive, Toulouse, 26-28 janvier 2005.

SWRL, I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet,B. Grosof et M. Dean "SWRL : A Semantic Web Rule Language Combining OWL and RuleML". W3C Member Submission 21 May 2004. Url : http://www.w3.org/Submission/SWRL/, accédé le 13/10/2012.

Tracy Allan Hall, « Syllable : Phonology », dans Keith Brown (dir.), *Encyclopedia of Language and Linguistics*, vol. 12, Oxford, Elsevier, 2006, 2e éd. (ISBN 0-08-044299-4), p. 329

Troncy R., « Formalisme des connaissances documentaires et des connaissances conceptuelles à l'aide d'ontologie : application à la description de documents audiovisuels » thèse de doctorat, université de Grenoble, soutenue le 5 mars 2004.

Troncy R., « Nouveaux outils et documents audiovisuels : les innovations du web sémantique » 392 Documentaliste – science de l'information 2005, Vol. 42, n°6.

UML, Charroux B., Osmani A. et Thierry-Mieg Y., "UML 2.", pearson education, 2005

Van Rossum G., Jansen J., Mullender K. and Bulterman D., «CMIFed : a presentation Environment for Portable Hypermedia Documents», Proc. of the ACM Multimedia Conf., California, 1993.

XHTML+SMIL "XHTML+SMIL Profile" Newman D., Schmitz P. and Patterson A., "W3C Working Draft", work in progress. Available at http://www.w3.org/TR/XHTMLplusSMIL/, visited on Ferbruary,15 2010.

Zargayouna H., « Indexation sémantique de documents XML» thèse de doctorat à l'université Paris XI Orsay. Soutenue le 15 décembre 2005.

Liste des publications personnelles

Publications dans des revues internationales

Ghalia Merzougui, Mahieddine Djoudi and Behaz Amel “Conception and Use of Ontologies for Indexing and Searching by Semantic Contents of Video Courses” *revue 'IJCSI-2011-8-6-702' Volume 8, Issue 6, November 2011.*

Ghalia Merzougui and Mahieddine Djoudi “An Authoring System for Editing Lessons in Phonetic English in SMIL3.0” revue 'IJCSI-2011-8-6-645' Volume 8, Issue 6, November 2011.

Ghalia Merzougui, Mahieddine Djoudi et Abdelmadjid Zidani “Editeur de cours médiatisés en SMIL” revue ISDM n°18 – 2004, Colloque TICE MEDITERRANEE 26 - 27 novembre 2004.

Communications dans des conférences internationales

Ghalia Merzougui et Mahieddine Djoudi « Une approche de conception et de production des cours de phonétique en SMIL 3.0 » conférence ICIST'2011 -Tébéssa, Algerie du 24 au 26 Avril 2011.

Ghalia Merzougui, Amel Behaz, Mahieddine Djoudi and Aouadj Messaoud “An Authoring System for Editing Lessons in Phonetic English in SMIL 3.0” *14th East-European on Advances in Database and Information Systems ADBIS 2010* .

Ghalia Merzougui, Amel Behaz and Mahieddine Djoudi, "Annotation de cours vidéo à base d’ontologie pour une indexation sémantique", *Conférence internationale des technologies de l’information et de la communication, CITIC’09, Sétif,* 04-05 mai 2009.

Amel Behaz, Ghalia Merzougui et Mahieddine Djoudi, "Approche de production de documents pédagogiques multimédias multi-formats", *Séminaire Euo-Mditerranéen pour l'Approfondissement de la Formation à Distance, SEMAFORAD, Bejaia, Algérie.,* 12-14 Nov 2005.

Amel Behaz, Ghalia Merzougui et Mahieddine Djoudi, "Approche de production de documents pédagogiques multimédias multi-formats", *Congrès International en Informatique Appliquée, CIIA'05, Bordj Bou Arréridj, Algérie.*, November 19-21 2005.

Ghalia Merzougui, Mahieddine Djoudi and Abdelmadjid Zidani, "Editor of Courses Mediatized on SMIL", *2004 International Arab Conference on Information Technology (ACIT 2004) Constantine, Algeria, ISSN: 1812-0857,* December 12-15 2004.

Ghalia Merzougui, Mahieddine Djoudi et Abdelmadjid Zidani, "Editeur de cours médiatisés en SMIL", *Conférence Internationale: Sciences Electroniques, Technologies de l'Information et des Télécommunications, IEEE SETIT 2004, ISBN 9973-41-902-2, Sousse, Tunisie,* 15-20 Mars 2004.

Ghalia Merzougui, Mahieddine Djoudi et Abdelmadjid Zidani, "Editeur de cours médiatisés en SMIL", *Colloque TICE Méditerranée,* NICE les 26 -27 Novembre 2004.

Communications dans des congrès nationaux (avec comité de lecture)

Ghalia Merzougui et Mahieddine Djoudi, "Approche de production de documents pédagogiques multimédias", *Journées d'étude sur les TIC, JeTIC2006, Bechar, Algérie*, 15-16 Avril 2006.

Annexes

Annexe 1 : **Partie du code source du système SACoPh**

Classe SmilGen : La classe SmilGen est la classe la plus importante de notre programme, c'est la partie responsable de la génération du document SMIL 3.0 et afin d'y parvenir, elle utilise l'interface DOM présente dans le module QtXml. La classe Leçon utilise la classe SmilGen au niveau de la fonction Génération comme suit :

```
void Lecon::Generation(QString SmilPath)
{
    Generateur = new SmilGen(PathLecon,SmilPath,text());
    Generateur->Initialiser();
    Generateur->setIndexRegions(VectRegle.size());
    Generateur->setStaticRegion(text(), NomProfesseur,NomEtablissement,
                      getListeRegle());
 for(int i=0;i<VectRegle.size();i++)
{
Generateur->InsertRule(VectRegle[i]-> getRecordTimeList(), VectRegle[i]->
getListeExemple(),VectRegle[i]->text());
}
delete Generateur;
}
```

Fonction 'insertion de l'index'

La fonction setIndexRegions permet d'insérer l'index de navigation hypermédia dans le fichier SMIL. Pour cela elle utilise une fonction récursive nommée InsertIndexRegions:

```
void SmilGen::InsertIndexRegions(QDomNode Node,int Nbr)
{
if(Node.toElement().tagName()=="layout")
{
QDomElement IndexElement = SmilDocument.createElement("region");
IndexElement.setAttribute("left",40);
IndexElement.setAttribute("right",590);
IndexElement.setAttribute("height",25);
IndexElement.setAttribute("textFontSize","18px");
IndexElement.setAttribute("textColor","black");
IndexElement.setAttribute("textFontWeight","bold");
IndexElement.setAttribute("textAlign","center");
IndexElement.setAttribute("backgroundColor","#a5a5b1");
for(int i=1;i<=Nbr;i++)
{
    QDomNode Clone = IndexElement.cloneNode(false);
    QString Attribute;
    Attribute.setNum(i);
```

```
    Attribute.prepend("Index");
    Clone.toElement().setAttribute("xml:id",Attribute);
    Clone.toElement().setAttribute("top",Top);
    Node.appendChild(Clone);
    Top+=30;
}
}
if(Node.hasChildNodes())
{InsertIndexRegions(Node.firstChild(),Nbr);}
if(!Node.nextSibling().isNull())
{InsertIndexRegions(Node.nextSibling(),Nbr);}
}
```

Annexe 2 : **Partie du code source du système IRSeCoV**

```
// Fonction pour parser un fichier owl afin de remplir la table d'index

void MainWindow::parseLessonowl(QStringList list)
{
    QStringList::Iterator itl = list.begin();
    int nomblv=0 ;
    while (itl != list.end())
    {
    QFile file(*itl);
    if (!file.open(QFile::ReadOnly | QFile::Text)) {
        return;
    }

    QString errorStr;
    int errorLine;
    int errorColumn;

    QDomDocument doc;
    if (!doc.setContent(&file, false, &errorStr, &errorLine,
                        &errorColumn)) {
        return ;
    }
    file.close();

    QDomElement root = doc.documentElement();
    if (root.tagName() != "rdf:RDF")
    {
        return ;
    }
    QDomNode child = root.firstChildElement ("cours_video");


    QDomNode n=child.firstChildElement ("est_presente_dans");
    QDomNode m=n.firstChildElement ("lesson_video");
    QDomElement j=m.toElement ();
    QDomNodeList nbrdiap = j.childNodes ();
    int nbrdia = nbrdiap.count ();


    int nlv = m.toElement ().attribute ("rdf:ID").remove (0,6).toInt ();
    QDomNode i=m.firstChild();



    while (!i.isNull ())
```

```
    {
        QDomNode diap = i.firstChild();
        QDomElement n = diap.toElement();

       int ndiap = n.attribute("rdf:ID").remove (0,11).section ('_',-1,-
1).toInt ();

            QDomElement debut = diap.firstChildElement ("Debut");

            QString deb = debut.text ();

            QDomElement duree = diap.firstChildElement ("Duree");
            QString dur = duree.text ();


            QDomElement cont = diap.firstChildElement ("contient");
            QDomNode contient = cont;

            while (!contient.isNull ())
            {
                QDomNode op = contient.firstChild();
                QString nop = op.toElement().attribute ("rdf:ID");


                QDomNode  conc = op.firstChild();
                while (!conc.isNull ())
                {
                    if (conc.toElement ().tagName () == "concerne")
                    {
                      QString concepte = conc.toElement ().attribute
("rdf:resource").remove (0,53);
chargerDansMap(nlv,nbrdia,ndiap,deb,dur,nop,concepte);
                    }
                    conc=conc.nextSibling ();
                }
                contient = contient.nextSibling();
            }

        i=i.nextSibling ();
    }
    ++itl;
    ++nomblv;
   }
//parcourir la table d'index pour calculer les poids des concepts dans
//chaque segment
      CalcPoid(nomblv);
      QMessageBox::information(this, "Concept Selection", "Le chargement
de MAP est terminer");

}
```

Annexe 3 : **Code SMIL d'une leçon généré par SACoPh**

Définition

Synchronized Multimedia Integration Language (ou SMIL) est une spécification du W3C. L'objectif de SMIL est de permettre l'intégration des contenus multimédias

diversifiés (images, sons, textes, vidéo, animations, flux de texte) en les synchronisant afin de permettre la création des présentations multimédias. SMIL est un langage de la famille XML. La structure XML d'un document SMIL décrit le déroulement temporel et spatial des différents composants intégrés. En d'autres termes, SMIL permet d'indiquer le moment où un contenu sera affiché, pendant combien de temps et dans quelle partie de la fenêtre d'affichage.

Versions de SMIL

• SMIL 1.0 : c'est la première version qui a été développé en juin 1998 par un groupe de travail appelé SYMM qui a été formé au sein de W3C. Cette version a pour objectif de permettre l'échange et l'évolution interopérable du web. SMIL 1.0 est devenu une recommandation du W3C en Juin 1998.

• SMIL 2.0 : une nouvelle version a été conçue par le même groupe en 2001 et qui avait comme objectif : de définir un langage basé sur XML et de permettre de réutiliser la syntaxe et la sémantique de SMIL dans d'autres langages basés sur XML et en particulier ceux qui nécessitent de représenter une temporisation et une synchronisation. Par exemple les composants de SMIL 2.0 sont utilisés pour intégrer la temporisation dans XHTM et SVG.

SMIL 2.0 est devenu une recommandation du W3C en août 2001.

Dans SMIL, les éléments et les attributs sont groupés dans des paquets indépendants appelés **modules.** Par exemple, *la disposition* et *les éléments* de région sont dans le module de disposition, et *les éléments* d'animateColor *et* d'animateMotion sont dans le module d'animation. Des modules de SMIL peuvent être groupés dans une langue, appelée **un profil**. Il y a deux profils de SMIL, "profil du langage de SMIL 2.0" et une version simplifiée, "profil de base de SMIL 2.0", conçu pour de petits dispositifs. Développés à partir de SMIL 1.0, les modules SMIL 2.0 apportent de nouvelles fonctionnalités qui se partagent entre 10 domaines fonctionnels comme le montre la figure ci-dessus :

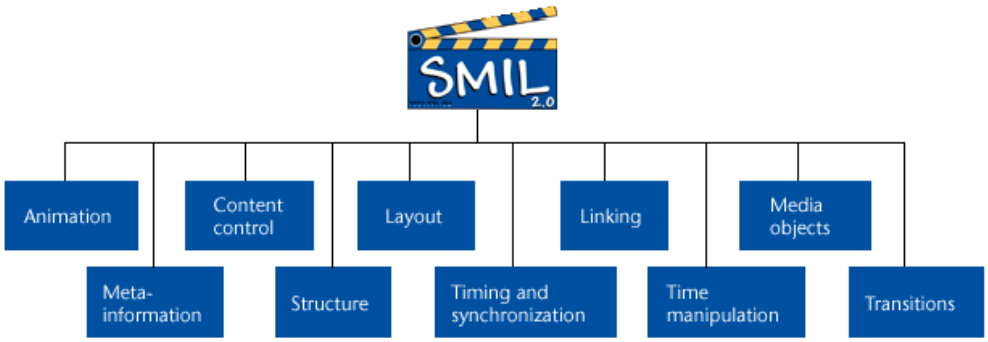


Figure 46. *Domaines fonctionnels de SMIL 2.0*

• SMIL 2.1 : a été reconçu sous la forme d'une série de modules composables fondés sur l'infrastructure fournie par la recommandation modularisation de XHTML de la W3C. Un profil de SMIL 2.1 représente un ensemble de module. La recommandation de SMIL 2.1 définit 3 nouveaux profils qui sont utiles à une large gamme d'applications :

– Le profile langage SMIL 2.1

– Le profile mobile SMIL 2.1

– Le profile mobile étendu SMIL 2.1

SMIL 2.1 est devenu une recommandation du W3C en Décembre 2005.

• SMIL 3.0 : les nouvelles fonctionnalités de SMIL 3.0 sont une réponse directe à la demande de l'industrie et des utilisateurs. Par exemple le standard permet d'insérer directement dans la présentation des animations, des légendes et des libellés synchronisés. Ainsi SMIL devient alors un langage descriptif très utile pour le développement des ressources du web sémantique qui évolue au fil du temps. On peut citer quelques balises supplémentaires de cette version :

<smilText> </smilText> : est une nouvelle balise qui fournis un élément conteneur de texte.

<textStyling> </textStyling> : permet d'ajouter des possibilités de formater un texte.

  : est une balise conteneur d'attributs de format au sein de la balise <smilText>.

<tev> </tev> : est une balise qui définit un point temporel, elle s'appuie sur la valeur des attributs begin et next.

SMIL 3.0 est devenu une recommandation du W3C en Décembre 2008.

Fonctionnalités de SMIL

- Gestion des médias (Les contenus).
- Gestion de l'espace (Les lay-out).
- Gestion du temps (Synchronisation).
- Gestion de l'interaction avec l'utilisateur (hypermédia).

Structure d'un document SMIL

Un document SMIL a pour racine l'élément <*smil*>, et se compose d'un en-tête et d'un corps. La partie <*head*> contient toutes les informations descriptives et relatives à la mise en page, alors que la partie <*body*> contient le corps du document, les informations sur les médias inclus et sur leurs comportements et la partie dynamique de la présentation.

La structure type d'un document SMIL peut être la suivante :

```
<smil>
<head>
<meta/>
<layout>
  <root-layout/>
  <region/>
</layout>
</head>
<body>
<switch> … </switch>
<par> … </par>
<seq> … </seq>
</body>
</smil>
```

A l'intérieur de la section ***<head>***, nous avons une partie ***<meta>*** qui contient des informations descriptives, comme l'auteur ou la date de conception de la présentation, et une partie ***<layout>*** qui définit le positionnement et la mise en page de la fenêtre dans laquelle va se jouer la présentation. Cette section contient elle-même deux types de balises. En premier lieu, la balise ***<root-layout>*** qui permet de définir les propriétés de la fenêtre de présentation, comme sa dimension et sa couleur de fond. Enfin, la balise ***<region>*** permet de définir des zones dans la fenêtre de présentation dans lesquelles les différents médias seront affichés. Chaque région doit avoir un identificateur, un

emplacement, et peut avoir une couleur. Dans l'exemple suivant, deux régions sont définies, une pour le texte, et une autre pour l'image :

```
<layout>
 <root-layout id="exemple" title="exemple" width="800" height="600"/>
 <region id="image" title="image" top="15" left="290"/>
 <region id="text" title="texte" top="270" left="290"/>
</layout>
```

La balise **<*body*>** et ses descendants permettent de placer les composants dans les régions définies dans l'élément <*layout*> décrit ci-dessus. Deux balises de base de cette section permettent de contrôler la ligne temporelle : <*par*> et <*seq*>. Les différents médias référencés dans un tag **<*par*>** sont joués simultanément. En revanche, les médias se trouvant à l'intérieur d'une balise **<*seq*>** sont joués en séquence, c'est-à-dire les uns après les autres. Il est possible d'imbriquer ces deux types de tag les uns dans les autres pour créer des sous-présentations. Dans l'exemple suivant, les images sont affichées en séquence (les unes à la suite des autres) mais en même temps qu'un texte descriptif :

```
<body>
<seq>
<par>
    <text region="text" src="./media/text1.txt" dur="10s"/>
    <img region="image" src="./media/img1.gif" dur="10s"/>
</par>
<par>
    <text region="text" src="./media/text2.txt" dur="10s"/>
    <img region="image" src="./media/img2.gif" dur="10s"/>
</par>
</seq>
</body>
```

Nous examinons maintenant plus précisément comment sont introduits les différents médias dans la présentation. On peut remarquer que chaque média possède sa propre balise. Ainsi sept éléments XML correspondant chacun à un type de média vont permettre de placer les composants dans une présentation. On distingue deux catégories de médias : les médias continus qui ont une durée intrinsèque (son, vidéo) et les médias discrets qui n'ont pas de durée prédéfinie (images, texte) mais qui pourront en avoir une dans le cadre de la présentation. Les différentes balises associées sont les suivantes :

<audio> pour les composants sonores (voix, son …)

<video> pour les séquences vidéos

<textstream> pour les textes dynamiques, avec déroulement automatique.

<animation> pour les animations, en Flash par exemple.

<ref> pour tout média continu qui n'est pas compris dans un des types cités ci-dessus.

<img> pour des images.

<text> pour des composants textuels.

Pour tous ces différents tags, l'attribut *src* permet de spécifier l'emplacement du média, qu'il se trouve dans le répertoire courant ou sur un serveur web distant. L'attribut région permet de définir l'emplacement dans lequel le média va se jouer (cet emplacement doit avoir été créé préalablement, voir partie <head>). Il est important de noter qu'à l'intérieur des balises <par> ou <seq>, le temps se déroule de manière automatique ; en effet, un média référencé dans un tag <seq> va se jouer directement à la suite de celui qui le précède. Cependant, si l'auteur de la présentation veut contrôler lui-même la ligne temporelle, il est possible de définir le début, la fin ou la durée d'un média. Tous les intervalles de temps sont mesurés par rapport à la ligne temporelle de SMIL, qui est propre à l'application de présentation. C'est l'utilisateur qui la contrôle : il a le choix de jouer la présentation, de la faire revenir en arrière, de l'avancer ou encore de la stopper.

Code SMIL d'une leçon généré par SACoPh

//en tête

```
<?xml version='1.1' encoding='UTF-8'?>
<!DOCTYPE smil PUBLIC '-//W3C//DTD SMIL 3.0 Language//EN'
'http://www.w3.org/2008/SMIL30/SMIL30Language.dtd'>
<smil version="3.0" xmlns="http://www.w3.org/ns/SMIL" baseProfile="Language">
 <head>
  <layout>
//spécifier les différentes regions (aspect spaciale)
   <root-layout width="800" xml:id="RootLayout" height="600"
src="BackgroundImage.jpg"/>
   <region textFontWeight="bold" xml:id="Lecon" textAlign="center" right="35"
left="165" height="35" top="43" textColor="white" textFontSize="18px"/>
   <region textFontWeight="bold" xml:id="Regle" right="35" left="250" height="92"
top="97" textColor="black" textFontSize="18px"/>
   <region textFontWeight="bold" xml:id="Exemple" textAlign="center" right="35"
left="340" height="270" top="235" textColor="black" textFontSize="18px"/>
   <region textFontWeight="bold" xml:id="Univ" right="35" left="515" height="30"
top="542" textColor="white" textFontSize="15px"/>
   <region textFontWeight="bold" xml:id="Prof" right="400" left="130" height="30"
top="542" textColor="white" textFontSize="15px"/>
//spécifier les regions pour les different index
   <region textFontWeight="bold" xml:id="Index1" textAlign="center" right="590"
left="40" height="25" backgroundColor="#a5a5b1" textColor="black" top="125"
textFontSize="18px"/>
```

```
   <region textFontWeight="bold" xml:id="Index2" textAlign="center" right="590"
left="40" height="25" backgroundColor="#a5a5b1" textColor="black" top="155"
textFontSize="18px"/>
   <region textFontWeight="bold" xml:id="Index3" textAlign="center" right="590"
left="40" height="25" backgroundColor="#a5a5b1" textColor="black" top="185"
textFontSize="18px"/>
  </layout>
 </head>
//le corps du document: contient l'agencement temporel de différents objets média
 <body>
  <par repeatCount="indefinite">
   <img fill="freeze" region="RootLayout" src="BackgroundImage.jpg"/>
   <smilText region="Lecon">the combination th</smilText>
   <smilText region="Prof">Mrs. Ameur</smilText>
   <smilText region="Univ">unhversity de Batna</smilText>
   <seq>
// Spécifier le premier segment (règle 1 avec ses exemples)
    <par xml:id="1" dur="30s">
     <audio begin="1s" src="tips 1.wav"/>
     <audio begin="13s" src="example 1.wav"/>
     <audio begin="18s" src="example 2.wav"/>
     <audio begin="24s" src="example 3.wav"/>
 // règle 1
     <smilText region="Regle" abstract="the combination 'th' is pronounced / /
voiceless in some words.">
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#aa00ff" textFontSize="20px">the combination th is pronounced </span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff5500" textFontSize="20px">/&#952;/ voiceless</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#aa00ff" textFontSize="20px"> in some words.</span>
      </p>
     </smilText>
    // exemple 1
     <smilText begin="13s" region="Exemple">
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff007f" textFontSize="20px">th</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">ink</span>
      </p>
    // exemple 2
      <tev begin="5s"/>
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff557f" textFontSize="20px">th</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">ief</span>
      </p>
    // exemple 3
      <tev begin="11s"/>
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">mon</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff557f" textFontSize="20px">th</span>
      </p>
     </smilText>
    </par>
// Spécifier le deuxième segment (règle 2 avec ses exemples)
    <par xml:id="2" dur="28s">
     <audio begin="1s" src="tips 2.wav"/>
     <audio begin="12s" src="example 4.wav"/>
     <audio begin="18s" src="example 5.wav"/>
     <audio begin="23s" src="example 6.wav"/>
     <smilText region="Regle" abstract=" 'th' is pronounciation /&#240;/ voiced in
some words .">
      <p>
```

```
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#aa00ff" textFontSize="22px">th is pronounciation /&#240;/ voiced in
some words .</span>
      </p>
     </smilText>
     <smilText begin="12s" region="Exemple">
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff557f" textFontSize="20px">th</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">ose</span>
      </p>
      <tev begin="6s"/>
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff557f" textFontSize="20px">th</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">e</span>
      </p>
      <tev begin="11s"/>
      <p>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">mon</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#ff557f" textFontSize="20px">th</span>
       <span textFontWeight="bold" textFontFamily="Lucida Calligraphy"
textColor="#000000" textFontSize="20px">er</span>
      </p>
     </smilText>
    </par>
// Spécifier le troisième segment (règle 3 avec ses exemples)
    <par xml:id="3" dur="21s">
     <audio begin="1s" src="tips 3.wav"/>
     <audio begin="9s" src="example 7.wav"/>
     <audio begin="15s" src="example 8.wav"/>
     <smilText region="Regle" abstract=" 'th' is pronounced /t/in some words.">
      <p>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#000000"  textFontSize="20px">th is pronounced </span>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#ff557f"  textFontSize="20px">/t/</span>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#000000"  textFontSize="20px">in some words.</span>
      </p>
     </smilText>
     <smilText begin="9s" region="Exemple">
      <p>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#ff557f"  textFontSize="20px">th</span>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#000000"  textFontSize="20px">ames</span>
      </p>
      <tev begin="6s"/>
      <p>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#ff557f"  textFontSize="20px">th</span>
       <span textFontWeight="bold"  textFontFamily="Lucida Calligraphy"
textColor="#000000"  textFontSize="20px">yme</span>
      </p>
     </smilText>
    </par>
   </seq>
// Spécifier la table d'index
   <a href="#1">
    <smilText region="Index1">tips 1</smilText>
   </a>
   <a href="#2">
    <smilText region="Index2">tips 2</smilText>
   </a>
```

```
   <a href="#3">
    <smilText region="Index3">tips 3</smilText>
   </a>
  </par>
 </body>
</smil>
```

Annexe 4 : Code OWL des ontologies élaborées

Code OWL de l'Ontologie de domaine d'enseignement

// En tête

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.owl-ontologies.com/Ontology1277939276.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1277939276.owl">
  <owl:Ontology rdf:about=""/>
```

// Déclaration des classes

```
//un domaine d'enseignement
```

```
<owl:Class rdf:ID="domaine_d_enseignement">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
               1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="aborde"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
represente un cours ou un module ou bien matiere a enseigner pour une formation
donnee</rdfs:comment>
  </owl:Class>

<owl:Class rdf:ID="concept">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="composer_de"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

// Déclaration des relations

```
<owl:ObjectProperty rdf:about="#aborde">
    <rdfs:range rdf:resource="#concept"/>
    <rdfs:domain rdf:resource="#domaine_d_enseignement"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
         enseigne </rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#composer_de">
    <rdfs:range rdf:resource="#concept"/>
    <rdfs:domain rdf:resource="#concept"/>
  </owl:ObjectProperty>

<owl:TransitiveProperty rdf:ID="pre_requis">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="#concept"/>
    <rdfs:domain rdf:resource="#concept"/>
  </owl:TransitiveProperty>
<owl:SymmetricProperty rdf:ID="depend">
    <rdfs:range rdf:resource="#concept"/>
    <rdfs:domain rdf:resource="#concept"/>
    <owl:inverseOf rdf:resource="#depend"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </owl:SymmetricProperty>
```

// Les instances

```
  <domaine_d_enseignement rdf:ID="structure_de_donnee">
  <aborde>
     <concept rdf:ID="donnee">
        <composer_de rdf:resource="#type_variable"/>
        <composer_de rdf:resource="#variable"/>
      </concept>
    </aborde>
    <aborde>
      <concept rdf:ID="instruction">
        <composer_de rdf:resource="#affectation"/>
        <composer_de rdf:resource="#instruction_de_controle"/>
        <composer_de rdf:resource="#instruction_conditionelle_imbrique"/>
        <composer_de>
          <concept rdf:ID="instruction_de_repetition_imbrique"/>
        </composer_de>
        <composer_de>
          <concept rdf:ID="instruction_de_repetition"/>
        </composer_de>
        <composer_de>
          <concept rdf:ID="instruction_conditionelle"/>
        </composer_de>
        <composer_de rdf:resource="#instruction_conditionelle_par_cas"/>
      </concept>
    </aborde>
    <aborde rdf:resource="#expression"/>
    <aborde rdf:resource="#sous_algorithme"/>
  </domaine_d_enseignement>
 <concept rdf:ID="instruction_de_controle"/>
<concept rdf:ID="parametre_formelle"/>
<concept rdf:ID="operateur_arithmetique"/>
<concept rdf:ID="instruction_conditionelle_par_cas"/>
<concept rdf:ID="procedure"/>
<concept rdf:ID="sous_algorithme">
    <composer_de rdf:resource="#procedure"/>
    <composer_de rdf:resource="#fonction"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >sous programme</rdfs:comment>
  </concept>
<concept rdf:ID="instruction_conditionelle_imbrique"/>

<concept rdf:ID="type_compose">
    <composer_de>
```

```
      <concept rdf:ID="tableau_a_deux_dimenssions"/>
    </composer_de>
    <composer_de>
      <concept rdf:ID="tableau_a_une_dimenssion"/>
    </composer_de>
    <composer_de>
      <concept rdf:ID="chaine_de_caractere">
        <composer_de>
          <concept rdf:ID="caractere"/>
        </composer_de>
      </concept>
    </composer_de>
  </concept>
  <concept rdf:ID="tableau">
    <composer_de rdf:resource="#tableau_a_deux_dimenssions"/>
    <composer_de rdf:resource="#tableau_a_une_dimenssion"/>
  </concept>
  <concept rdf:ID="variable_globale"/>
  <concept rdf:ID="affectation"/>
  <concept rdf:ID="boucle"/>
    <owl:sameAs rdf:resource="#instruction_de_repetition"/>
  </concept>


  <concept rdf:ID="variable">
    <composer_de>
      <concept rdf:ID="allocation_statique_d_une_variable"/>
    </composer_de>
    <composer_de>
      <concept rdf:ID="allocatino_dynamique_d_une_variable">
        <depend>
          <concept rdf:ID="pointeur">
            <depend>
              <concept rdf:ID="adresse">
                <depend rdf:resource="#pointeur"/>
              </concept>
            </depend>
            <depend rdf:resource="#allocation_dynamique_d_une_variable"/>
          </concept>
        </depend>
      </concept>
    </composer_de>
  </concept>
  <concept rdf:ID="expression">
    <composer_de>
      <concept rdf:ID="operateur_logique"/>
    </composer_de>
    <composer_de>
      <concept rdf:ID="operateur_binaire"/>
    </composer_de>
    <composer_de>
      <concept rdf:ID="operateur_ternaire"/>
    </composer_de>
    <composer_de rdf:resource="#operateur_arithmetique"/>
  </concept>
  <concept rdf:ID="passage_parametre_par_variable">
    <depend>
      <concept rdf:ID="fonction">
        <pre_requis>
          <concept rdf:ID="bloc_d_instructions"/>
        </pre_requis>
        <depend>
          <concept rdf:ID="fonction_recurssive">
            <depend rdf:resource="#fonction"/>
          </concept>
        </depend>
        <depend>
          <concept rdf:ID="passage_parametre_par_valeur">
            <depend rdf:resource="#fonction"/>
```

```
        </concept>
      </depend>
      <composer_de>
        <concept rdf:ID="parametre">
          <composer_de>
            <concept rdf:ID="parametre_effectif"/>
          </composer_de>
          <composer_de rdf:resource="#parametre_formelle"/>
        </concept>
      </composer_de>
      <composer_de>
        <concept rdf:ID="variable_locale"/>
      </composer_de>
      <composer_de rdf:resource="#variable_globale"/>
      <depend rdf:resource="#passage_parametre_par_variable"/>
      <composer_de>
        <concept rdf:ID="valeur_retournee"/>
      </composer_de>
      <composer_de rdf:resource="#passage_parametre_par_valeur"/>
      <composer_de rdf:resource="#bloc_d_instructions"/>
    </concept>
  </depend>
</concept>
<concept rdf:ID="arbre"/>
<concept rdf:ID="passage_par_adresse">
  <owl:sameAs rdf:resource="#passage_parametre_par_variable"/>
</concept>
<concept rdf:ID="logique"/>
<concept rdf:ID="type_variable">
  <composer_de rdf:resource="#type_compose"/>
  <composer_de>
    <concept rdf:ID="type_complexe">
      <composer_de>
        <concept rdf:ID="graphe"/>
      </composer_de>
      <composer_de>
        <concept rdf:ID="liste_chainee">
          <depend>
            <concept rdf:ID="enregistrement">
              <depend rdf:resource="#liste_chainee"/>
              <composer_de>
                <concept rdf:ID="champ_d_un_enregistrement"/>
              </composer_de>
            </concept>
          </depend>
        </concept>
      </composer_de>
      <composer_de>
        <concept rdf:ID="file">
          <pre_requis rdf:resource="#pointeur"/>
        </concept>
      </composer_de>
      <composer_de>
        <concept rdf:ID="pile"/>
      </composer_de>
      <composer_de>
        <concept rdf:ID="fichier"/>
      </composer_de>
      <composer_de rdf:resource="#arbre"/>
      <composer_de rdf:resource="#enregistrement"/>
    </concept>
  </composer_de>
  <composer_de>
    <concept rdf:ID="type_scalaire">
      <composer_de rdf:resource="#pointeur"/>
      <composer_de rdf:resource="#caractere"/>
      <composer_de>
        <concept rdf:ID="reel"/>
      </composer_de>
```

```
        <composer_de rdf:resource="#logique"/>
        <composer_de>
          <concept rdf:ID="entier"/>
        </composer_de>
      </concept>
    </composer_de>
  </concept>

<concept rdf:ID="argument">
    <owl:sameAs rdf:resource="#parametre"/>
  </concept>
  <concept rdf:ID="instruction_de_repetition_en_cascade">
    <owl:sameAs rdf:resource="#instruction_de_repetition_imbrique"/>
  </concept>
  <concept rdf:ID="ouvrir_fichier">
    <depend rdf:resource="#fichier"/>
  </concept>
  <concept rdf:ID="matrice">
    <owl:sameAs rdf:resource="#tableau_a_deux_dimenssions"/>
  </concept>
  <concept rdf:ID="vecteur"/>
  <concept rdf:ID="type_fichier">
    <depend rdf:resource="#fichier"/>
  </concept>
  <concept rdf:ID="mode_d_ouverture">
    <depend rdf:resource="#fichier"/>
  </concept>
  <concept rdf:ID="fermer_fichier">
    <depend rdf:resource="#fichier"/>
  </concept>
  <concept rdf:ID="creer_fichier">
    <depend rdf:resource="#fichier"/>
  </concept>
….

</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 3.3.1, Build 430)
http://protege.stanford.edu -->
```

Code OWL de l'Ontologie de cours video

// en tête

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
    <!ENTITY p1 "http://www.owl-ontologies.com/Ontology1277939276.owl#" >
]>

<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1277284608.owl#"
     xml:base="http://www.owl-ontologies.com/Ontology1277284608.owl"
     xmlns:p1="http://www.owl-ontologies.com/Ontology1277939276.owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology rdf:about="">
```

// Importer l'ontologie de domaine d'enseignement

```
<owl:imports rdf:resource="http://www.owl-ontologies.com/Ontology1277939276.owl"/>
    </owl:Ontology>
```

// Déclaration des classes

//un cours vidéo concerne un domaine d'enseignement

```
<owl:Class rdf:ID="cours_video">
     <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="est_presente_dans"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Class>
```

// leçon_video est equivalent au chapitre dans un cours

```
 <owl:Class rdf:ID="lesson_video">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="est_divisee"/>
        </owl:onProperty>
        <owl:minCardinality  rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
 </owl:Class>
```

// diapositif = slide

```
<owl:Class rdf:ID="diapositif">
     <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="contient"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
```

// OP : Objet Pédagogique

```
<owl:Class rdf:ID="OP">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="concerne"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Class>
```

// Déclaration des relations

```
<owl:ObjectProperty rdf:ID="est_presente_dans">
        <rdfs:domain rdf:resource="#cours_video"/>
        <rdfs:range rdf:resource="#lesson_video"/>
    </owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="est_divisee">
        <rdfs:domain rdf:resource="#lesson_video"/>
        <rdfs:range rdf:resource="#diapositif"/>
    </owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="contient">
        <rdfs:domain rdf:resource="#diapositif"/>
        <rdfs:range rdf:resource="#OP"/>
    </owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="concerne">
        <rdfs:domain rdf:resource="#OP"/>
        <rdfs:range rdf:resource=" &p1;#concept"/>
    </owl:ObjectProperty>
```

// Déclaration des attributs

```
<owl:DatatypeProperty rdf:ID="Debut">
        <rdfs:domain rdf:resource="#diapositif"/>
        <rdfs:range rdf:resource="&xsd;time"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Duree">
        <rdfs:domain rdf:resource="#diapositif"/>
       <rdfs:range rdf:resource="&xsd;time"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="langue">
        <rdfs:domain rdf:resource="#cours_video"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="Titre_diapositif">
        <rdfs:domain rdf:resource="#diapositif"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="URL">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#lesson_video"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="professeur_video">
    <rdfs:domain rdf:resource="#cours_video"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
```

//les instances

```
<cours_video rdf:ID="structure_de_donnee">
    <est_presente_dans>
      <lesson_video rdf:ID="fonction">
        <est_divisee rdf:resource="#diapositif_2"/>
        <est_divisee rdf:resource="#diapositif_3"/>
        <est_divisee rdf:resource="#diapositif_7"/>
      </lesson_video>
    </est_presente_dans>
    <est_presente_dans rdf:resource="#tableau"/>
    <langue rdf:datatype="http://www.w3.org/2001/XMLSchema#string">francais
      </langue>
    <professeur_video rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
```

```
      Abdellatif El Faker</professeur_video>
</cours_video>

<diapositif rdf:ID="diapositif_2">

    <Duree rdf:datatype="http://www.w3.org/2001/XMLSchema#time">00:03:22 </Duree>
    <Debut rdf:datatype="http://www.w3.org/2001/XMLSchema#time">00:02:01 </Debut>
    <Titre_diapositif rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
           introduction au function </Titre_diapositif>
    <contient>
      <OP rdf:ID="Definition_1">
        <concerne rdf:resource="&p1#fonction"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
             decalration d'une fonction en C</rdfs:comment>
      </OP>
    </contient>
    <contient>
      <OP rdf:ID="definition_2">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >nom de la fonction est une adresse</rdfs:comment>
        <concerne rdf:resource="&p1#adresse"/>
        <concerne rdf:resource="&p1#fonction"/>
      </OP>
    </contient>
    <contient>
      <OP rdf:ID="exemple_1">
        <concerne rdf:resource="&p1#valeur_retournee"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
              differents type  de valeurs retournee</rdfs:comment>
      </OP>
    </contient>
</diapositif>

<diapositif rdf:ID="diapositif_3">
    <Debut rdf:datatype="http://www.w3.org/2001/XMLSchema#time">00:05:27</Debut>
    <Duree rdf:datatype="http://www.w3.org/2001/XMLSchema#time">00:03:43</Duree>
    <Titre_diapositif rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
defintion d'un prototype </Titre_diapositif>
    <contient>
      <OP rdf:ID="exemple_2">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >
 pourquoi un prototype</rdfs:comment>
        <concerne rdf:resource="&p1#parametre_formelle"/>
        <concerne rdf:resource="&p1#parametre_effectif"/>
      </OP>
    </contient>
    <contient>
      <OP rdf:ID="definition_3">
        <concerne rdf:resource="&p1#parametre_effectif"/>
        <concerne rdf:resource="&p1#parametre_formelle"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
definition d'un prototype d'une fonction</rdfs:comment>
      </OP>
    </contient>
</diapositif>


<diapositif rdf:ID="diapositif_7">
  <Debut rdf:datatype="http://www.w3.org/2001/XMLSchema#time" >00:21:14</Debut>
  <Duree rdf:datatype="http://www.w3.org/2001/XMLSchema#time" >00:02:04</Duree>
  <Titre_diapositif rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
passage de paramete par valeur </Titre_diapositif>
  <contient>
      <OP rdf:ID="exemple_3">
        <concerne rdf:resource="&p1#passage_parametre_par_valeur"/>
      </OP>
    </contient>
</diapositif>
```

```
<lesson_video rdf:ID="tableau"/>

</rdf:RDF>
```

// La définition de la règle 1 avec SWRL

```
<swrl:Imp rdf:ID="Rule-1">
    <swrl:body>
      <swrl:AtomList>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:rest>
              <swrl:AtomList>
                <rdf:rest>
                  <swrl:AtomList>
                    <rdf:first>
                      <swrl:IndividualPropertyAtom>
                        <swrl:propertyPredicate rdf:resource="#contient"/>
                        <swrl:argument2>
                          <swrl:Variable rdf:ID="op2"/>
                        </swrl:argument2>
                        <swrl:argument1>
                          <swrl:Variable rdf:ID="d2"/>
                        </swrl:argument1>
                      </swrl:IndividualPropertyAtom>
                    </rdf:first>
                    <rdf:rest>
                      <swrl:AtomList>
                        <rdf:first>
                          <swrl:IndividualPropertyAtom>
                            <swrl:argument2>
                              <swrl:Variable rdf:ID="c2"/>
                            </swrl:argument2>
                            <swrl:propertyPredicate rdf:resource="#concerne"/>
                            <swrl:argument1 rdf:resource="#op2"/>
                          </swrl:IndividualPropertyAtom>
                        </rdf:first>
         <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                      </swrl:AtomList>
                    </rdf:rest>
                  </swrl:AtomList>
                </rdf:rest>
                <rdf:first>
                  <swrl:IndividualPropertyAtom>
                    <swrl:argument1>
                      <swrl:Variable rdf:ID="c1"/>
                    </swrl:argument1>
                       <swrl:propertyPredicate
rdf:resource="http://www.owlontologies.com/Ontology1277939276.owl#SameAs"/>
                    <swrl:argument2 rdf:resource="#c2"/>
                  </swrl:IndividualPropertyAtom>
                </rdf:first>
              </swrl:AtomList>
            </rdf:rest>
            <rdf:first>
              <swrl:IndividualPropertyAtom>
                <swrl:propertyPredicate rdf:resource="#concerne"/>
                <swrl:argument2 rdf:resource="#c1"/>
                <swrl:argument1>
                  <swrl:Variable rdf:ID="op1"/>
                </swrl:argument1>
              </swrl:IndividualPropertyAtom>
            </rdf:first>
          </swrl:AtomList>
        </rdf:rest>
        <rdf:first>
```

```
          <swrl:IndividualPropertyAtom>
            <swrl:argument2 rdf:resource="#op1"/>
            <swrl:propertyPredicate rdf:resource="#contient"/>
            <swrl:argument1>
              <swrl:Variable rdf:ID="d1"/>
            </swrl:argument1>
          </swrl:IndividualPropertyAtom>
        </rdf:first>
      </swrl:AtomList>
    </swrl:body>
    <swrl:head>
      <swrl:AtomList>
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        <rdf:first>
          <swrl:IndividualPropertyAtom>
            <swrl:argument1 rdf:resource="#d1"/>
            <swrl:propertyPredicate rdf:resource="#Similaire"/>
            <swrl:argument2 rdf:resource="#d2"/>
          </swrl:IndividualPropertyAtom>
        </rdf:first>
      </swrl:AtomList>
    </swrl:head>
  </swrl:Imp>
<!-- Created with Protege (with OWL Plugin 3.4.8, Build 629)
http://protege.stanford.edu -->
```