CAMLearn: Une Architecture de Système de Recommandation Sémantique Sensible au Contexte. Application au Domaine du M-Learning.

Fayrouz Soualah-Alila

▶ **To cite this version:**

Fayrouz Soualah-Alila. CAMLearn: Une Architecture de Système de Recommandation Sémantique Sensible au Contexte. Application au Domaine du M-Learning.. Informatique [cs]. Université de Bourgogne, 2015. Français. NNT : . tel-01238233v2

HAL Id: tel-01238233
https://hal.science/tel-01238233v2

Submitted on 14 Jan 2016

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

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

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

CAMLearn*: Une architecture de système de recommandation sémantique sensible au contexte. Application au domaine du m-learning.

FAYROUZ SOUALAH ALILA

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

THÈSE présentée par

FAYROUZ SOUALAH ALILA

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne

Spécialité : **Informatique**

CAMLearn*: Une architecture de système de recommandation sémantique sensible au contexte. Application au domaine du m-learning.

Soutenue publiquement le 18/03/2015 devant le Jury composé de :

M. JACKY AKOKA	Rapporteur	Professeur, CNAM Paris
MME. MYRIAM LAMOLLE	Rapporteur	Professeur, IUT Montreuil
M. IOAN ROXIN	Examinateur	Professeur, Univ. Franche-Comté
MME. CATHERINE FARON ZUCKER	Examinatrice	Maître de conférences, Univ. Nice Sophia Antipolis
MME. SANDRA BRINGAY	Examinatrice	Maître de conférences, Univ. Montpellier
M. CHRISTOPHE NICOLLE	Directeur de thèse	Professeur, Univ. Bourgogne
MME. FLORENCE MENDES	Encadrante de thèse	Maître de conférences, Univ. Bourgogne

*CONTEXT-AWARE M-LEARNING

REMERCIEMENTS

Au terme de ce travail de recherche, je suis convaincue que cette thèse est loin d'être un travail solitaire. Je tiens donc à remercier tous ceux qui, de loin ou de près, ont contribué à la réalisation de ce projet.

En premier lieu, je remercie vivement les membres du jury, qui ont accepté d'évaluer mon travail. Je remercie les rapporteurs de cette thèse, Jacky Akoka (professeur associé au CNAM à Paris), et Myriam Lamolle (professeur associé à l'IUT de Montreuil). Je les remercie pour la rapidité avec laquelle ils ont lu mon manuscrit, malgré un emploi du temps sans doute très chargé. Je leur suis très reconnaissante pour l'intérêt qu'ils ont porté à mes travaux de recherche.
Je remercie également Ioan Roxin (professeur à l'université de Franche-Comté), Catherine Faron Zucker (maître de conférences à l'université de Nice Sophia Antipolis) et Sandra Bringay (maître de conférences à l'université de Montpellier) d'avoir accepté de faire partie de mon jury.

Je tiens à exprimer mes plus vifs remerciements à Christophe Nicolle qui fut pour moi un directeur de thèse attentif et disponible malgré ses nombreuses charges. Je le remercie pour la confiance qu'il m'a toujours accordée, pour son soutien et ses conseils au cours de l'élaboration de cette thèse. J'ai énormément appris à ses côtés et ces quelques lignes sont peu de choses par rapport à tout ce qu'il m'a apporté.

J'adresse tout ma gratitude à Florence Mendes, encadrante de cette thèse, pour ces encouragements, son aide et son soutien sans faille tout au long de ces années de travail. Je la remercie aussi pour toutes nos discussions et ses conseils qui ont été et resteront des moteurs de mon travail de chercheur.

J'adresse aussi mes remerciements à Erik Gebers qui a participé à mon encadrement dans cette thèse et qui a contribué au bon déroulement de ce travail de recherche. Je lui souhaite d'ailleurs bon courage pour ses projets futurs.

Je remercie Florian Orpelière pour sa grande patience et son aide inconditionnelle, et Valériane Eymenier pour sa bonne humeur et son appui. Ces deux ingénieurs de l'équipe Checksem ont permis de faire avancer ce travail de thèse.

Les travaux présentés dans cette thèse ont fait l'objet d'une convention CIFRE entre la société CrossKnowledge et l'équipe Checksem.
Je tiens à avoir une pensée sympathique pour toutes les personnes sans exception, collègues et amis de la société CrossKnowledge, qui savent si bien rendre agréable le

cadre de travail. Je remercie particulièrement Muriel Richelet pour sa patience et sa gentillesse. Une pensée particulière à Bruno Larosa qui nous a quittés il y'a un peu plus d'un an pour un monde meilleur. Reposes en paix.
Il m'est impossible d'oublier de remercier mes collègues de l'équipe Checksem pour leur bonne humeur et leur soutien. C'est un plaisir de les côtoyer tous les jours. Je ne vais pas les citer de peur d'en oublier quelqu'un. Merci à mes voisins de bureau qui ont du supporter ma présence. Je remercie particulièrement le trio Yoan Chabot, Benjamin Harbelot et David Werner. Les pauses café et les PhD nights vont me manquer.

Mes remerciements vont aussi à ma famille et mes amis qui, avec cette question récurrente, « quand est-ce que tu la soutiens cette thèse ? », bien qu'angoissante en période fréquente de doutes, m'ont permis de ne jamais dévier de mon objectif final.
Cela va de soi, je remercie avec grande émotion mes sœurs soussou et doudou, mon petit frère rourou, mon oncle migo, pour leur irremplaçable soutien, et ma petite Mélie pour sa bonne humeur permanente. Ils m'ont toujours encouragé à aller de l'avant dans la vie malgré la difficulté d'être loin de ses proches. Merci d'avoir été là pour écarter les doutes, soigner les blessures et partager les joies. Un grand remerciement à papa pour son soutien et ses encouragements.

Ces remerciements seraient incomplets si je n'en adressais pas à mon héroïne : ma maman. Sa présence et ses encouragements sont pour moi les piliers fondateurs de ce que je suis et de ce que je fais. Cette thèse elle est pour toi chouchou !

Last but not least, je remercie chaleureusement Hadrien Levionnois, pour ses encouragements et sa grande patience (je te promets je vais essayer de râler moins). Tu auras connu tous les moments difficiles mais aussi les petites victoires. Dans chacun de ces moments ton amour sans faille m'aura donné la force d'un géant.

J'en oublie certainement encore et je m'en excuse.

Encore un grand merci à tous pour m'avoir conduit à ce jour mémorable.

RÉSUMÉ

En raison de la croissance rapide des technologies de l'information et de la communication, les méthodes d'apprentissage sont en constante évolution. Le e-learning (ou apprentissage électronique) a été considéré comme la première application des systèmes hypermédia adaptatifs. Les systèmes de recommandation sont basés sur les systèmes hypermédia adaptatifs développés durant la fin des années 90.
Maintenant, le grand défi des entreprises e-learning est de combler le fossé entre le e-learning statique et l'apprentissage mobile (ou m-learning). Parmi les caractéristiques du m-learning : la flexibilité, l'accessibilité et l'informalité. La flexibilité est la capacité du système à accéder librement à la formation d'apprentissage sans contraintes dans le temps et l'espace. L'accessibilité est la capacité du système à trouver des informations sur la formation d'apprentissage indépendamment de l'organisation initiale de cet apprentissage. L'informalité est la capacité du système à développer un processus d'apprentissage non intentionnel qui découle des activités de la vie quotidienne, contrairement à l'apprentissage prédéfini par le formateur.
Nos travaux de recherche portent sur le développement d'une architecture pour le m-learning. Nous proposons une approche pour un système m-learning contextuel et adaptatif intégrant des stratégies de recommandation de scénarios de formations sans risque de rupture. Notre approche respecte les caractéristiques de flexibilité, d'accessibilité et d'informalité. Ce système est une extension des travaux effectués sur les systèmes de recommandation basés sur le contenu.
Dans le domaine du e-learning, la plupart des propositions existantes de systèmes de recommandation sont basées sur un ensemble de couches qui sont inspirées des architectures hypermédia adaptatifs. Une architecture formée de trois couches minimum est nécessaire pour la création d'un système hypermédia adaptatif. Ces trois couches sont : le modèle de domaine, le modèle utilisateur et le modèle d'adaptation. Nous avons adapté ces trois modèles à notre architecture.
Nous proposons une architecture d'un système de recommandation formé de deux parties : une partie statique représentant à la fois les connaissances sur les contenus d'apprentissage et le contexte de l'apprenant ; et une partie comportementale contenant les règles et les métaheuristiques, qui visent à combiner des modules d'apprentissage. Notre système prend en compte le contexte spatio-temporel de l'apprenant, l'évolution du profil de l'apprenant et l'adaptation dynamique des modules durant le processus d'apprentissage dans un environnement mobile.

ABSTRACT

Due to the rapid growth of information and communications technology, learning methods are evolving. E-learning (or electronic learning) was considered as the first application of adaptive hypermedia systems. Recommender systems are based on adaptive hypermedia systems developed during the end of the 90's.
Now, the great challenge of e-learning companies is to bridge the gap between the static e-learning and the mobile learning (or m-learning). M-learning features are numerous but can be focused on these three mains : flexibility, accessibility and informality. Flexibility is the ability of the system to access freely to learning courses without constraints in time and space. Accessibility is the ability of the system to find any piece of information about learning courses independently from the initial organization of this learning. Informality is the ability of the system to develop a process of learning beyond the original path of learning defined by the teacher.
Our research focuses on the development of a new architecture for a m-learning recommandation system. We propose an approach incorporating contextual and adaptive strategies and respecting flexibility, accessibility and informality characteristics of m-learning. The proposed systeme is an extension of works on content based recommender systems.
In the e-learning domain, most of the existing proposals are based on a set of layers which are closed to adaptive hypermedia architectures. A consensus of proposals described a basic set of three layers made of a domain model, a user model and an adaptive model. We adapted these three layers to our architecture.
We propose a new architecture as an extension of content based recommender system, made of a static part representing both teachers knowledge and learners profile and context, and a behavioral part containing rules and metaheuristics, which aims at combining learning modules. Our system takes into account learners spatio-temporal context, the evolution of learner's profile and the dynamic adaptation of modules during the learning process in a mobile environment.

SOMMAIRE

I Introduction 1

Introduction 3

Contexte 4

Approche 5

Structure du document 6

II État de l'art 9

1 Mobile Learning 13

1.1 Introduction à l'apprentissage en situation de mobilité 13

1.1.1 Du d-learning au m-learning 14

1.1.2 Le m-learning dans les entreprises 18

1.1.3 Définition du m-learning 21

1.1.4 Caractéristiques du m-learning 23

1.2 Plateformes m-learning 26

1.3 Le m-learning, une activité en contexte 29

1.4 Conclusion 30

2 Contexte et sensibilité au contexte 31

2.1 Contexte 32

2.1.1 Définition du contexte 32

2.1.2 Dimensions du contexte 35

2.1.2.1 Dimension spatiale 36

2.1.2.2 Dimension temporelle 36

2.1.2.3 Dimension utilisateur 37

2.1.2.4 Dimension dispositif 38

2.1.3 Acquisition des informations de contexte 38
2.1.4 Modélisation du contexte 39
2.2 Systèmes sensibles au contexte 43
2.2.1 Définition de la sensibilité au contexte 43
2.2.2 Architecture générale d'un système sensible au contexte 44
2.2.3 Exemples d'architectures de systèmes sensibles au contexte 46
2.3 Conclusion 54

3 Systèmes de recommandation 57
3.1 Introduction aux systèmes de recommandations 58
3.2 Approches de recommandation 60
3.2.1 Recommandations basées sur le contenu 60
3.2.2 Recommandations par filtrage collaboratif 65
3.2.3 Avantages et inconvénients des approches de recommandation . . 67
3.2.4 Approches hybrides 68
3.2.5 Systèmes de recommandation sensibles au contexte 70
3.3 Conclusion 70

III Approche sémantique et combinatoire pour un système de recommandation sensible au contexte 73

4 Autour des objets pédagogiques 77
4.1 Objet pédagogique, définition et caractéristiques 78
4.1.1 Définition d'un objet pédagogique 78
4.1.2 Caractéristiques d'un objet pédagogique 79
4.2 Cycle de vie d'un objet pédagogique 80
4.3 Normes pour la description d'un objet pédagogique 83
4.3.1 Dublin Core 84
4.3.2 LOM 85
4.3.3 IMS-LD 87
4.3.4 SCORM 89
4.3.5 TIN CAN 92

4.3.6 LRMI . . . 93
4.3.7 Que choisir pour notre modèle ? . . . 93
4.4 Conclusion . . . 95

5 Architecture du système 97

5.1 Présentation générale de l'architecture . . . 98
5.2 Serveur sémantique de connaissances . . . 100
5.2.1 Le Web sémantique . . . 101
5.2.2 Modélisation par les ontologies . . . 104
5.2.3 Modèle des LO . . . 105
5.2.4 Modèle de contexte . . . 110
5.2.5 Modèle de contraintes . . . 120
5.3 Modèle d'adaptation . . . 124
5.4 Conclusion . . . 125

6 Problème d'optimisation 127

6.1 Définition du problème d'optimisation combinatoire . . . 128
6.1.1 Formalisation des LO . . . 129
6.1.2 Pondération des LO . . . 130
6.1.3 Graphe des LO . . . 132
6.1.4 Pertinence d'une solution . . . 135
6.2 Les métaheuristiques . . . 137
6.2.1 Introduction aux métaheuristiques . . . 137
6.2.2 Méthodes métaheuristiques approchées . . . 138
6.2.3 Méthodes constructives . . . 139
6.2.4 Méthodes de recherche locale / à solution unique . . . 140
6.2.4.1 Le Hill-Climbing . . . 141
6.2.4.2 Le recuit simulé . . . 142
6.2.4.3 La méthode tabou . . . 145
6.2.4.4 La recherche à voisinage variable . . . 146
6.2.5 Méthodes évolutives / à population de solution . . . 147
6.2.5.1 Les algorithmes génétiques . . . 149

6.2.5.2 Les algorithmes à estimation de distribution 151
6.2.5.3 Les algorithmes de colonies de fourmis 152
6.3 Implémentation des métaheuristiques 154
6.3.1 Construction d'une solution initiale par un algorithme de type glouton 154
6.3.2 Amélioration de recherche locale par un algorithme Hill-Climbing . . 156
6.3.3 Amélioration de recherche locale par un algorithme de recuit simulé 157
6.3.4 Amélioration de recherche locale par un algorithme tabou 158
6.3.5 Amélioration de recherche locale par un algorithme VNS 160
6.4 Résultats expérimentaux 163
6.4.1 Jeux de données 163
6.4.2 Paramétrage des algorithmes et résultats 164
6.5 Conclusion 166

IV Conclusion et travaux futurs 169

Conclusion et travaux futurs 171

Conclusion 171
Travaux futurs 173

V Annexes 175

A Les hypermédia adaptatifs 177

A.1 Les hypermédia classiques 177
A.2 Les hypermédia adaptatifs 180
A.2.1 Méthode et techniques d'adaptation 180
A.2.2 Architecture des hypermédia adaptatifs 183
A.3 Avantages et inconvénients 190

B La logique de description 191

B.1 Les langages formels et le Web sémantique 191
B.1.1 Les bases de connaissances 192
B.1.2 Les familles de logiques de description 193

B.1.3 Le langage ontologique OWL 197
B.2 Les raisonneurs 200
B.2.1 Inférence 201
B.2.2 Approches de raisonnement 202
B.2.3 Les raisonneurs pour OWL 204
B.3 Conclusion 205

C Étude comparative des triplestores 207
C.1 Évolution des modèles de données 208
C.1.1 Le modèle hiérarchique 208
C.1.2 Le modèle réseau 208
C.1.3 Le modèle relationnel 209
C.1.4 Le modèle objet 209
C.1.5 Le modèle sémantique 210
C.2 Les solutions triplestores 213
C.2.1 Syntaxes RDF supportés 213
C.2.2 Langages d'interrogation 215
C.2.3 Inférence 216
C.2.4 Langages supportés pour les clients 217
C.2.5 Licences 217
C.3 Conclusion 218

Rapports techniques 219

Publications 221

Bibliographie 236

I

INTRODUCTION

INTRODUCTION

L'apprentissage constitue aujourd'hui une préoccupation majeure. Propulsées par les technologies en constante évolution, les théories et les méthodes pédagogiques se succèdent dans l'objectif d'élaborer et de mettre en place des outils susceptibles de rendre plus efficaces et pertinents les processus d'apprentissage.

La source de l'apprentissage à distance est apparu il y a plus de deux cent ans avec l'invention de la poste moderne au Royaume-Uni et par conséquent l'apparition des cours par correspondance destinés aux apprenants qui étaient dans l'impossibilité de s'inscrire dans des établissements d'enseignement (cause de la distance, maladie, etc.). Sa principale caractéristique est la séparation entre l'enseignant et l'apprenant. L'apprentissage commence alors à sortir des salles de cours et à entrer dans des environnements moins classiques. Au gré de l'apport des technologies de l'information et de la communication, l'apprentissage à distance se décline en de multiples façons d'apprentissage : serious game (application dont l'objectif est de combiner l'aspect sérieux de l'apprentissage avec des ressorts ludiques issus des jeux vidéo), apprentissage électronique, apprentissage mobile ou encore blended learning (combinaison de l'apprentissage électronique à un apprentissage classique présentiel), sont aujourd'hui de nouvelles méthodes de l'apprentissage en plein essor.
L'évolution dans la formation à distance peut être caractérisée comme un passage de l'apprentissage classique présentiel, à l'apprentissage électronique puis aujourd'hui à l'apprentissage mobile. Ces trois stades de développement correspondent à l'influence de la révolution industrielle dans les pays en développement, sur la société du 18 ème au 19 ème siècle, la révolution électronique des années 1980 et la révolution des technologies mobiles et sans fil des dernières années du 20 ème siècle (les téléphones portables, les baladeurs numériques, les ordinateurs portables, les assistants personnels, les Net Books et maintenant les tablettes et les Smartphones).
Personnel, portable, collaboratif, interactif, adapté au contexte, l'apprentissage mobile a des qualités spécifiques qui le distinguent de l'apprentissage à distance classique. Il privilégie l'apprentissage juste à temps et une transmission des connaissances qui peut être effectuée en tout lieu et à tout moment. L'apprentissage mobile peut être ainsi défini comme une activité dont le but est de localiser et de délivrer des contenus pédagogiques à cet apprenant en fonction de son besoin en informations et du contexte dans lequel il se situe. Jusqu'à présent l'environnement d'apprentissage était soit défini par un cadre pédagogique (exemple, l'école, le formateur, etc.) soit imposé par le contenu

pédagogique (l'apprenant devait alors organiser son environnement pour recevoir la formation). Dans notre travail, nous cherchons à l'inverse à adapter le cadre pédagogique et le contenu pédagogique au contexte dynamique de l'apprenant.
Afin d'atteindre cet objectif, il est nécessaire d'adapter les systèmes d'apprentissage électronique actuels avec de nouvelles technologies pour la représentation, la modélisation, l'indexation des contenus pédagogiques dans un contexte de mobilité.

CONTEXTE

Le travail de recherche élaboré dans le cadre de cette thèse s'inscrit dans un projet de collaboration entre la société CrossKnowledge [1] et l'équipe Checksem [2] (Semantic Intelligence Research) du laboratoire LE2I [3] (Laboratoire Electronique, Informatique et Image – Université de Bourgogne).
CrossKnowledge est le leader européen des solutions d'apprentissage électronique et du développement des compétences managériales sur internet. Fondée en 2000, CrossKnowledge est une entreprise internationale de plus de 200 salariés, présente dans de nombreux pays d'Europe, d'Asie et d'Amérique. CrossKnowledge conçoit, développe et commercialise le catalogue le plus complet du marché avec plus de 17 mille formations multilingues en apprentissage électronique abordant l'ensemble des thématiques de leadership, de management et d'accompagnement du changement. Ces contenus sont conçus en partenariat avec des professeurs et des experts d'HEC, ESSEC, HARVARD, INSEAD, etc. CrossKnowledge compte plus de 3 million d'apprenants au sein de grandes entreprises internationales de tous secteurs.
La recherche dans l'équipe Checksem est basée sur plusieurs éléments fondateurs. Les travaux concernent premièrement l'ingénierie d'ontologies, à savoir la modélisation, le peuplement, la persistance et l'évolution des ontologies. Les travaux au sein de Checksem concernent aussi l'ingénierie de règles (règles logiques, métaheuristiques). Le dernier aspect des recherches au sein de l'équipe vise la qualification formelle des connaissances sémantiques ainsi que la gestion des profils et contextes utilisateurs lors de l'exploitation des ontologies, tout au long du cycle de vie du système d'information. Cet ensemble de compétences forme un axe de recherche cohérent sur le thème de la modélisation sémantique et de son exploitation contextuelle par des systèmes intelligents.

Ce projet de collaboration a pour objectif la proposition d'une nouvelle architecture pour la conception d'une plateforme d'apprentissage mobile. Cette plateforme doit permettre par le développement d'un système de recommandation sémantique orienté métier appliqué au contexte de l'apprentissage mobile, la personnalisation de l'expérience pédagogique

1. http ://www.crossknowledge.com/fr_FR/elearning/home.html
2. http ://checksem.u-bourgogne.fr/www/
3. http ://le2i.cnrs.fr/

selon les besoins des apprenants (connaissances et compétences à acquérir) et les situations d'apprentissage (contexte de mobilité : informel, en entreprise, etc.).
L'enjeu derrière cet effort d'innovation est d'augmenter l'efficacité de l'apprentissage, que ce soit du point de vue des apprenants, des formateurs ou des organisations finançant la formation.

APPROCHE

Pour atteindre les objectifs fixés par l'entreprise, nous avons souhaité étudier dans un premier temps les systèmes sensibles au contexte. La sensibilité au contexte est un concept indispensable dans la conception de systèmes d'apprentissage mobile. L'étude des architectures de ces systèmes nous a permis de dégager les principaux concepts pour la conception d'un système contextuel. Ces architectures constituent un cadre général pour la conception, l'acquisition d'informations contextuelles et l'adaptation des contenus aux utilisateurs. L'objectif est de rendre le système capable de réagir de façon pertinente aux changements de l'environnement dans lequel se situe l'apprenant.

Nous avons étudié dans un second temps les systèmes hypermédia adaptatifs pour comprendre comment fonctionnent les mécanismes d'adaptation d'un ensemble de données. L'hypermédia adaptatif concerne le développement de système Web où le contenu et la navigation sont dynamiquement adaptés en fonction d'un profil utilisateur. Nous avons pu identifier que l'architecture d'un système hypermédia adaptatif est construite par un empilement de couches. Une architecture minimale formée de trois couches peut servir de base à tous types de systèmes hypermédia adaptatifs. Cette architecture est formée d'un modèle de domaine, un modèle utilisateur et un modèle d'adaptation. Le premier modèle contient l'ensemble des données du système, le second décrit le profil de l'utilisateur et le dernier contient les règles d'adaptation.

Notre recherche nous a permis d'identifier un sous-ensemble des systèmes hypermédia adaptatifs qui sont les systèmes de recommandations. Nous avons présenté une étude approfondie de ces systèmes qui nous a permis d'identifier les principales caractéristiques de ces systèmes et de proposer une classification des principales approches. Ces approches proposent des mécanismes pour recommander un item [4] à un utilisateur soit en se basant sur une analyse du contenu des offres à recommander, soit en se basant sur une analyse des avis des utilisateurs. Bien que nous ayons pu identifier certains mécanismes utiles pour notre problème, tels que ceux utilisant des ontologies pour définir la sémantique des contenus, aucun projet de recherche identifié ne permet ni de proposer des combinaisons d'items, ni de proposer des mécanismes de recommandation basés sur le savoir-faire des experts métiers. Le travail de recherche présenté

4. Item est le terme usuel dans le domaine des systèmes de recommandation pour parler d'un élément à recommander

dans ce mémoire a pour principal objectif de développer une architecture générique qui permet la construction d'un système de recommandation qui répond aux limites de ces systèmes.

Nous avons aussi étudié les standards et normes actuelles de modélisation des contenus d'apprentissage. Un problème d'efficacité se pose aujourd'hui avec le développement des contenus pédagogiques ces dernières années, qui a conduit à un important volume d'informations hétérogènes et difficilement réutilisables. Les efforts de standardisation qui ont vu le jour suite à ce problème, doivent nécessairement être pris en compte dans les nouveaux projets du domaine afin d'assurer l'interopérabilité, le partage et la réutilisation des contenus.

A l'intersection de ces différents domaines de recherche, ce travail présente une architecture pour la construction d'un système de recommandation sensible au contexte dédié au domaine de l'apprentissage mobile.
Pour répondre au problème de combinaison de contenus pédagogiques adaptés au contexte, nous avons étudié le domaine de recherche de l'optimisation combinatoire. Ce domaine nécessite de définir précisément le problème d'optimisation qui doit être étudié. Nous avons défini le problème d'optimisation d'offres d'apprentissage, puis nous avons étudié les différentes métaheuristiques qui pouvaient résoudre ce problème.
Pour répondre aussi au problème de la modélisation du savoir-faire des experts métiers et au verrou de l'hétérogénéité des contenus, nous avons étudié le domaine du Web sémantique. L'ontologie est un concept central du Web sémantique. Une des définitions de l'ontologie qui fait autorité dans le domaine est celle de (Gruber, 1993) : une ontologie est une spécification explicite d'une conceptualisation. Le caractère formel d'une ontologie implique que celle-ci doit être exploitable par des machines tandis que le caractère partagé de sa conceptualisation indique que les créateurs de l'ontologie doivent avoir une même vision du domaine représenté. Une ontologie est ainsi vue comme une structure de données composée de concepts, de relations entre ces dernières et représentant les connaissances d'un domaine. Pour résumer, l'utilisation d'une ontologie dans notre système permet de standardiser le vocabulaire, d'uniformiser le langage d'échange entre les différentes plateformes d'apprentissage et de structurer la connaissance pour simplifier l'analyse des contenus pédagogiques. Nous avons ainsi étendu les propositions des systèmes de recommandations utilisant des ontologies en ajoutant des mécanismes de pondération et des règles logiques pour répondre aux besoins de l'entreprise en matière de modélisation de son savoir-faire métier.

STRUCTURE DU DOCUMENT

Ce rapport de thèse est articulé en 6 chapitres :
– Le chapitre 1 est un état de l'art de l'apprentissage mobile. Nous présentons

dans ce chapitre l'historique de développement, les classifications et les caractéristiques de l'apprentissage. Nous présentons et comparons aussi des architectures représentatives de l'apprentissage mobile dans lesquelles nous avons cherché des inspirations. Finalement nous décrivons l'apprentissage mobile comme une activité en contexte.

– Dans le chapitre 2 nous présentons un état de l'art du contexte et de la sensibilité au contexte. Dans la première partie du chapitre nous présentons les différentes visions de la notion du contexte en essayant d'en faire ressortir les caractéristiques que nous regrouperons en dimensions. La combinaison de ces dimensions permet de construire des contextes spécifiques pour des besoins clairement identifiés. Dans la deuxième partie du chapitre nous étudions l'architecture des systèmes sensibles au contexte en essayant d'en extraire les principales fonctionnalités requises pour prendre en charge ce contexte en apprentissage mobile.
– Le chapitre 3 est un état de l'art de différentes méthodes et techniques de recommandations utilisées pour la résolution de notre projet de recherche. La première partie de ce chapitre présente brièvement le domaine des systèmes hypermédia adaptatifs (un état de l'art plus complet sur les systèmes hypermédia adaptatifs est proposé dans l'annexe A). Cette partie met en évidence une architecture suffisante pour la réalisation de systèmes adaptatifs avec une description des couches principales. Le but des systèmes hypermédia adaptatifs est de résoudre le problème de surcharge cognitive (ou surcharge d'informations) et le problème de perte dans l'hyperespace. Nous décrivons ces deux problèmes dans le chapitre. Un sous-domaine des hypermédia adaptatifs, chargé de résoudre uniquement le premier problème, est appelé systèmes de recommandation. Dans ces systèmes, présentés dans la seconde partie de ce chapitre, il existe deux principales méthodes de recommandations : la méthode basée sur le contenu et la méthode par filtrage collaboratif. Nous identifions qu'aucune de ces deux méthodes ne répond à notre besoin de combiner plusieurs items selon des critères sémantiques et contextuels pour en faire une recommandation à l'utilisateur final.
– Dans le chapitre 4 nous introduisons la notions d'objets pédagogiques. Les ressources pédagogiques sont considérées comme la pierre angulaire de l'appentissage électronique. La qualité des ressources pédagogiques détermine la qualité de la formation. L'assurance qualité se définit comme le résultat des dispositions prises par les organismes de normalisation qui visent à établir des règles qui faciliteront le partage et la réutilisation des objets pédagogiques. Nous détaillerons dans ce chapitre le point de vue adopté par les différents organismes de normalisation et les faiblesses de certaines approches. Cette étude nous permet de nous positionner par rapport aux différentes normes proposés et de choisir celle qui est la plus adaptée pour notre système.
– Le chapitre 5 définit formellement un nouveau type de système de recommandation proposant des combinaisons d'objets pédagogiques et permettant la prise en compte,

sous la forme de règles métiers du savoir-faire des fournisseurs des contenus. Cette proposition est une extension des systèmes de recommandation basés sur le contenu utilisant une ontologie associée à des mécanismes de pondération et d'indexation. L'architecture utilisée est issue des systèmes hypermédia adaptatifs et est composée de trois couches : le modèle des objets pédagogiques (qui correspond au modèle de domaine dans l'architecture hypermédia), le modèle de contexte (qui correspond au modèle utilisateur) et le modèle d'adaptation. Les deux premiers modèles forment un serveur sémantique des connaissances. Pour proposer des recommandations sous forme de combinaisons d'objets pédagogiques, le modèle d'adaptation est décrit de manière à pouvoir utiliser des algorithmes d'optimisation combinatoire adaptés au domaine d'application.

– Le chapitre 6 a pour objectif de définir le problème d'optimisation combinatoire en apprentissage mobile et de proposer des pistes pour la résolution de ce problème. Ce chapitre définit formellement le problème de combinaison d'objets pédagogiques et présente les concepts nécessaires à la résolution des problèmes d'optimisation. Comme notre problème n'est pas résolvable par des heuristiques classiques en un temps raisonnable, la deuxième partie du chapitre présente les métaheuristiques les plus connues pour résoudre des problèmes d'optimisation combinatoire classiques. Finalement, la dernière partie du chapitre est dédié à la description et l'implémentation des algorithmes métaheuristiques utilisés pour résoudre notre problème. Quatre algorithmes sont testés pour la génération de combinaisons : un algorithme de type Hill-Climbing, un algorithme basé sur le recuit simulé, un algorithme tabou et un algorithme de recherche à voisinage variable. Des benchmarks de comparaison des performances de ces algorithmes sont présentés. Il en ressort que l'algorithme de recherche à voisinage variable est supérieur en terme de qualité.
– Trois annexes et une bibliographie viennent compléter ce rapport de thèse. L'annexe A présente un état de l'art sur les systèmes hypermédia adaptatifs, l'annexe B porte sur la logique de description et l'annexe C propose une étude comparative entre différents triplestores.

II

ÉTAT DE L'ART

Les dispositifs mobiles offrent aujourd'hui des possibilités de formation que nous ne pouvons pas avoir couramment avec d'autres outils d'apprentissage. Ils permettent notamment d'accéder à des contenus n'importe où et en tout temps, et de vivre de nouvelles situations d'apprentissage dans différents lieux (Droui et al., 2014).
Cette variabilité du contexte en apprentissage mobile nécessite d'une part, de lever des verrous sur la modélisation du contexte, mais aussi de trouver des solutions pour préserver les objectifs pédagogiques du processus d'apprentissage, malgré les variations du contexte de l'apprenant. Pour répondre à ces objectifs, de nombreux travaux de recherche ont été menés ses 15 dernières années.
Tout d'abord, dans le domaine de l'apprentissage mobile, où l'obtention d'un consensus pour une définition précise de l'apprentissage mobile reste difficile. Ensuite, plus spécifiquement dans le domaine de la modélisation contextuelle, qui n'est pas propre à l'apprentissage mobile. La définition d'un contexte est une tâche difficile, composée de nombreux paramètres. L'identification de ses paramètres et leur modélisation dans un tout cohérent à destination de l'apprentissage mobile demande un grand travail de classification. Enfin, dans le domaine des systèmes hypermédia adaptatifs, qui propose des architectures de recommandation dont le but est de limiter la surcharge cognitive de l'utilisateur par une sélection appropriée d'un item ou d'une combinaison d'items.
Notre état de l'art est ainsi composé de trois grands chapitres dédiés à chacun des domaines précisés.

Le premier chapitre est un état de l'art de l'*apprentissage mobile*.
Ce chapitre présente une brève description de l'évolution de l'apprentissage mobile ainsi que les nombreuses définitions associées. L'apprentissage mobile caractérise tout type d'apprentissage qui se produit lorsque l'apprenant n'est pas à un endroit fixe, prédéterminé, et lorsque ce dernier utilise des technologies de l'information et de la communication mobiles pour se former (O'Malley et al., 2005).
Certaines approches considèrent que l'apprentissage mobile est simplement une extension de l'apprentissage électronique. Toutefois, nous verrons que les besoins et les attentes des apprenants dans une situation de mobilité diffèrent de ceux des apprenants dans un contexte d'enseignement électronique.
Afin de mieux adapter l'apprentissage aux besoins des apprenants en mobilité, ce chapitre présentera une étude des caractéristiques des environnements mobiles et du comportement des systèmes d'apprentissage dans ces environnements. La mobilité est traduite par les notions de contexte d'apprentissage et de sensibilité au contexte.

Le second chapitre se concentre naturellement sur la notion de *contexte* et la notion de *sensibilité au contexte*.
Nous avons en tant qu'humains une compréhension intuitive du contexte. Nous réagissons différemment en fonction d'éléments tels que l'heure, la localisation, la météo, le niveau de bruit, etc. Ainsi, en fonction des informations reçues nous adaptons notre comportement aux conditions externes. Il en est de même pour les nouveaux systèmes

informatiques.
Dans les deux dernières décennies, l'évolution de l'informatique mobile, a donné lieu à de nombreux travaux sur la notion de contexte. En apprentissage mobile le but est d'améliorer l'efficacité de l'expérience d'apprentissage, en recommandant les contenus appropriés au contexte actuel de l'apprenant et aux changements de l'environnement.
Vu la complexité du contexte et les différentes informations contextuelles qui pourraient influencer les systèmes mobiles, il est nécessaire de classifier les différents types de contexte et les façons de réagir au contexte. La sensibilité au contexte est un concept indispensable dans la conception d'un système d'apprentissage mobile. L'objectif est de rendre le système capable de réagir de façon pertinente aux changements de l'environnement de l'apprenant. On en arrive ainsi à l'adaptation ou encore la recommandation de contenus d'apprentissage dans un contexte mobile.

Le troisième chapitre de l'état de l'art est une étude des systèmes *hypermédia adaptatifs*, dont l'utilisation est répandue dans le domaine de l'apprentissage électronique.
De nombreuses architectures sont proposées pour construire de tels systèmes. Elles sont toutes construites à partir d'une architecture pivot, nécessaire et suffisante pour la réalisation d'un système adaptatif (Picot-Clémente, 2011) ; cette architecture est articulée en trois couches : une couche qui représente (1) l'utilisateur et le contexte, (2) une couche qui représente le domaine et (3) une couche qui représente les mécanismes d'adaptation.
Le but des systèmes hypermédia adaptatifs est de résoudre deux problèmes : le problème de perte dans l'hyperespace (perte de repères au sein de la structure du réseau hypertexte) et le problème de surcharge cognitive (surcharge par l'abondance d'informations et par l'excès de tâches à réaliser). Dans un environnement d'apprentissage mobile, l'objectif est de proposer à l'apprenant des contenus adaptés, sans que ce dernier se perde dans la multitude de propositions.
Une sous-classe des systèmes hypermédia adaptatifs chargés de résoudre le problème de surcharge cognitive est appelée *systèmes de recommandation*. La seconde partie de ce chapitre présente les différentes catégories de systèmes de recommandation et surtout leurs limites par rapport aux objectifs de recommandation d'un système d'apprentissage en mobilité.

1

MOBILE LEARNING

L'objectif des travaux en mobile learning est de construire un environnement d'apprentissage où les activités doivent s'adapter à la situation de mobilité de l'apprenant à l'aide des nouvelles technologies numériques.
Ce chapitre présente un état de l'art de l'apprentissage en situation de mobilité (apprentissage mobile ou mobile learning ou m-learning) articulé en trois parties. La première partie présente un bref historique de l'évolution de l'apprentissage électronique vers l'apprentissage mobile ainsi qu'une étude des définitions de l'apprentissage mobile et de ses caractéristiques. La seconde partie est une analyse des différentes stratégies et plateformes développées dans le cadre de l'apprentissage mobile. La troisième partie se focalise sur l'impact du contexte sur l'apprentissage mobile.

Sommaire

1.1 Introduction à l'apprentissage en situation de mobilité 13
1.1.1 Du d-learning au m-learning 14
1.1.2 Le m-learning dans les entreprises 18
1.1.3 Définition du m-learning 21
1.1.4 Caractéristiques du m-learning 23
1.2 Plateformes m-learning 26
1.3 Le m-learning, une activité en contexte 29
1.4 Conclusion 30

1.1/ INTRODUCTION À L'APPRENTISSAGE EN SITUATION DE MOBILITÉ

Avec le développement du tout connecté, l'accès à la connaissance et sa transmission se sont démultipliés de façon exponentielle ces 30 dernières années. Propulsées par les nouvelles technologies, les théories et les méthodes se succèdent pour élaborer et mettre en place des outils susceptibles de rendre plus efficaces et pertinents les processus

d'apprentissage. Néanmoins, efficaces et pertinents signifient aussi en adéquation avec leur époque et en prise avec les sociétés qu'ils doivent instruire (Liénard, 2010).

Dans la suite, nous présenterons tout d'abord, un résumé de l'évolution de la transmission du savoir dans le domaine de l'apprentissage à distance. Ensuite, nous focaliserons notre étude sur l'usage du e-learning au sein des entreprises. L'apprentissage dans un environnement professionnel est un enjeu stratégique pour l'économie des entreprises. C'est dans cet environnement que de nombreuses études et avancées ont été réalisées pour proposer des solutions d'apprentissage mobile, chacune des solutions avec ses propres objectifs et sa propre vision de l'apprentissage mobile. Enfin nous profiterons de cette hétérogénéité pour identifier la principale contrainte qui peut différencier l'apprentissage électronique classique (e-learning) de l'apprentissage en situation de mobilité (m-learning) : l'impact du contexte de l'apprenant en situation d'apprentissage.

1.1.1/ DU D-LEARNING AU M-LEARNING

Historiquement, l'apprentissage à distance, ou d-learning, est apparu après l'apprentissage classique il y a plus de deux cent ans. L'apprentissage à distance, est «*l'utilisation de techniques pédagogiques, de ressources et de moyens de communication spécifiques qui facilitent l'apprentissage, impliquant des apprenants et des formateurs séparés dans le temps et dans l'espace*»[1]. Les premières formations à distance ont vu le jour à l'Université de Londres, au 19 ème siècle, avec des cours par correspondance destinés aux citoyens britanniques qui étaient dans l'impossibilité de s'inscrire à cause de la distance. Plus précisément, l'apprentissage à distance par correspondance est né avec la naissance de la poste. Ensuite, avec l'apparition de chaque nouveau média (e.g. la radio, le téléphone, la télévision) qui permettait de faciliter la communication, cette technologie était utilisée pour enrichir l'apprentissage à distance et les approches pédagogiques.
Les approches pédagogiques permettent de « *répondre à l'ambivalence mathétique par la polyvalence didactique* » (Leclercq, 1998). Avec l'émergence des Technologies d'Information et de Communication (TIC), il est possible de répondre à la diversité de styles cognitifs (visuel ou auditif), aux besoins, aux attentes des apprenants en matière d'apprentissage. De plus la variété de ces technologies permet d'entretenir chez les apprenants, la motivation nécessaire pour s'investir dans leur apprentissage. L'apprenant devient acteur et peux choisir lui-même la méthode, le parcours, le moment.

Avec l'apparition des TIC, l'apprentissage à distance a évolué pour devenir un apprentissage rapide et efficace, avec un minimum de problèmes d'organisation et de perte de temps. Ce mode d'apprentissage est basé sur l'accès à des formations en ligne, interactives et parfois personnalisées, diffusées par l'intermédiaire d'un réseau (Internet ou Intranet) ou d'un autre média électronique[2]. Cet accès permet de développer les

1. http ://www.unesco.org/new/fr/unesco/themes/icts/lifelong-learning/open-and-distance-learning/
2. http ://eduscol.education.fr/numerique/dossier/archives/eformation/e-formation-e-learning

compétences des apprenants, tout en rendant le processus d'apprentissage indépendant du temps et du lieu. Ce nouveau mode d'apprentissage est l'apprentissage électronique, plus connu sous l'appellation e-learning.

Il existe une grande diversité de définitions et de manière de nommer l'e-learning (formation en ligne, site web éducatif, télé-formation, e-training, etc.). "Certaines définitions sont centrées sur le support technologique, d'autres sur la distance, d'autres insistent sur les aspects pédagogiques", tandis que d'autres insistent sur les types d'interactions.

La définition la plus répandue de l'e-learning est celle proposée par la Commission Européenne, «*c'est l'utilisation des nouvelles technologies multimédias et de l'Internet pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que les échanges et la collaboration à distance*» [3]. Cette définition présente l'e-learning comme une nouvelle technique d'apprentissage ne consistant simplement qu'à mettre sur internet des ressources pédagogiques ou des applications d'évaluation de connaissances. Même si on peut effectivement classer ce genre d'apprentissage dans le domaine du e-learning, cette vision est très réductrice et nous renvoie aux systèmes d'apprentissage des années 60-70, appelés aussi systèmes d'Enseignement Assisté par Ordinateur (EAO) (Blanchard, 2007).
Le guide de l'e-learning [4] préfère s'appuyer sur la définition proposée par le LabSET [5] (Laboratoire de Soutien à l'Enseignement Télématique) : «*e-learning ou electronic learning : apprentissage en ligne centré sur le développement de compétences par l'apprenant et structuré par les interactions avec le tuteur et les pairs*».

Avec cette nouvelle définition de l'e-learning centrée sur l'apprenant, et depuis le début des années 80, l'EAO a alors évolué vers ce que l'on nomme fréquemment Environnements Informatiques pour l'Apprentissage Humain (EIAH). Alors que l'EAO se préoccupait plutôt de la transmission de contenus, l'EIAH tente d'orienter l'apprentissage en fonction de l'apprenant (ses préférences pédagogiques, ses connaissances, ses objectifs, etc.). L'objectif de l'EIAH est d'adapter l'apprentissage en fonction des particularités propres à l'apprenant.
Ainsi, en e-learning, on peut distinguer deux types d'applications : d'une part les systèmes d'apprentissage ne proposant aucune adaptation, et d'autre part les systèmes d'apprentissage adaptatifs cherchant à mieux comprendre l'apprenant pour lui fournir le meilleur contenu correspondant à ses besoins. Ces derniers systèmes sont communément appelés Systèmes Tutoriels Intelligents (STI).

Les STI sont des systèmes d'apprentissage informatiques, répandus aujourd'hui sous le nom de Learning Management Systems (LMS), qui possèdent (1) des contenus sous forme de base de connaissances, (2) des stratégies d'apprentissage et (3) une connaissance sur l'apprenant pour une meilleure adaptation (Murray, 1999). Il s'agit principale-

3. http ://www.awt.be/web/edu/index.aspx ?page=edu,fr,gui,080,010
4. http ://www.awt.be/
5. http ://www.labset.ulg.ac.be/portail/

ment de concevoir des systèmes d'apprentissage individualisés.

Un LMS est une plateforme e-learning développée pour accompagner tout apprenant dans son processus de formation. Les plateformes LMS sont les outils les plus utilisés pour accéder aux cours tout en permettant une adaptation des contenus pédagogiques. Ils permettenet aussi d'administrer des contenus provenant de différents fournisseurs mais ne fournissentt pas des outils pour la création de contenus d'apprentissage, qui est le rôle des LCMS (Learning Content Management System). Une plateforme LCMS est donc destinée aux experts de la production de formations et intervient en amont de la diffusion des contenus (cette 2e étape est réalisée via une plateforme LMS).

Propriété		**Plateforme**					
		Claroline	**Blackboard**	**Ganesha**	**Moodle**	**Sakai**	**CK LMS**
Adaptation	Personnalisation par les formateurs du menu de navigation	+	+	-	-	+	+
	Personnalisation par l'apprenant de l'interface du cours	-	-	-	+	-	+
	Personnalisation du contenu en fonction du profil de l'apprenant	+/-	-	-	-	-	+/-
Évaluation	QCM, QCU	+	+	+	+	+	+
	Description, réponse courte, mise en correspondance, texte à trous	+	+	-	+	+	+
Collaboration	Forum	+	+	+	+	+	+
	Wiki	+	+	-	+	+	+
	Groupes d'apprenants	+	+	+	+	+	+
	Communauté d'apprentissage	-	-	-	+	+	+

TABLE 1.1 – Tableau comparatif de LMS

Il existe aujourd'hui plusieurs LMS en licence libre, comme par exemple Claroline [6] et Blackboard [7]. Ces plateformes sont personnalisables, adaptables et offrent un environ-

6. http ://www.claroline.net/
7. http ://fr.blackboard.com/sites/international/globalmaster/

nement de travail flexible et sur mesure. Ganesha [8] est une plateforme qui permet à un formateur, dans le cadre d'une formation à distance ou pour enrichir le présentiel, de mettre à la disposition d'un ou plusieurs groupes de stagiaires des modules de formation avec supports de cours, compléments, quiz et tests d'évaluation ainsi que des outils collaboratifs. L'apprenant n'est plus seul face à son cours et aux interrogations qu'il peut se poser, mais fait partie d'un groupe de travail dans lequel il peut communiquer avec les autres participants à l'aide d'un forum ou d'un chat (Dogbe-Semanou et al., 2008). Moodle [9] et Sakai [10] intègrent la notion de communauté d'apprentissage, c'est à dire partager ses connaissances et communiquer avec les formateurs et les apprenants, plutôt que juste mettre en ligne des contenus pédagogiques.
Parmi les LMS en licence payante et vu le contexte de la thèse, nous citons la plateforme d'apprentissage de CrossKnowledge, connu sous le non de CrossKnowledge Learning Suite. La CrossKnowledge Learning Suite prend la forme d'un Learning portal de dernière génération, personnalisé, social et mobile. Il permet de regrouper en un seul espace toutes les activités d'apprentissage auxquelles est inscrit le participant : ses parcours, ses ressources e-learning, ses échanges avec son tuteur, son groupe ou sa communauté, ses flux RSS, etc. Le Learning Portal de CrossKnowledge est assez flexible et paramétrable sans compétence technique particulière. La personnalisation peut se faire avec de multiples possibilités de mise en page de l'environnement de formation, grâce à une large sélection de widgets disponibles.
Le tableau 1.1 résume les caractéristiques des différentes plateformes décrites précédemment. Nous remarquons qu'aucune de ces plateformes ne propose d'adaptation des contenus suivant le profil de l'apprenant, seuls les paramètres d'affichage peuvent être modifiés.

En e-learning, un nouveau phénomène a émergé au cours de ces dernières années, il s'agit des MOOC (Massive Open Online Course). L'apparition des MOOC aux Etats-Unis dans les années 2010 relève de choix économiques de pure rentabilité. L'objectif affiché était de répondre à la crise des universités privées américaines, suite à la hausse des frais de scolarité, à l'augmentation des emprunts étudiants, au chômage croissant des jeunes diplômés et à l'affaiblissement de leur retour sur investissement. En 2011, trois cours en ligne massifs proposés par l'université de Stanford dépassent la barre symbolique des 100 000 étudiants, un an avant que les plateformes Coursera [11], Udacity [12] ou edX [13] fassent réellement leur apparition. Depuis, les MOOC connaissent un essor remarquable, et prennent de plus en plus d'ampleur en matière d'enseignement. Mais alors que les MOOC se donnaient l'ambition de démocratiser l'enseignement et d'attirer un nouveau public vers l'université, il apparaît que la plupart des inscrits qui réussissent

8. http ://www.ganesha.fr/
9. https ://moodle.org/
10. https ://sakaiproject.org/
11. https ://fr.coursera.org/
12. https ://www.udacity.com/
13. https ://www.edx.org/

une formation MOOC sont déjà diplômés. En effet, l'inscription à un MOOC est très facile, il reste ensuite la capacité de l'étudiant à s'autodiriger, à s'autoévaluer et à exploiter les ressources à sa disposition, donc à les décoder. En réalité la plus part des étudiants inscrits n'ont pas, au départ, forcément la volonté de suivre le cours de A à Z, alors qu'un MOOC nécessite un effort de longue durée qui nécessite une grande motivation initiale. Si les inscriptions sont massives, les abandons le sont tout autant (seuls 5% des inscrits valident leur formation) [14].
Suite à ce au taux d'abandon trop élevé, une évolution des MOOC fait apparition. Le SPOC (Small Private Online Courses) s'impose donc comme une solution adéquate : permettre une meilleure proximité formateur/étudiant ainsi qu'un meilleur encadrement des apprenants. Et surtout, l'obtention d'un diplôme à la fin, donne plus de crédibilité à la formule des SPOC. Les SPOC ont été expérimenté par exemple à Harvard et San José. San José, dont l'expérience MOOC avait été peu concluante, a conclu un partenariat SPOC avec la plateforme Edx, dont les premiers résultats sont encourageants : 91 % des étudiants ont obtenu leur examen à la fin du cours basé sur le SPOC, contre 65 % en moyenne les années précédentes [15].

1.1.2/ LE M-LEARNING DANS LES ENTREPRISES

Aujourd'hui, l'e-learning est utilisé principalement par deux secteurs bien distincts : le secteur des grandes entreprises qui l'utilisent pour former et actualiser les connaissances de leurs employés de manière plus rapide, et le secteur des universités qui offrent des alternatives de formation à distance. Vu l'omniprésence d'Internet dans les établissements scolaires et les nouvelles technologies qui font désormais partie de la vie des jeunes générations, l'intégration de l'e-learning devient nécessaire dans le milieu éducatif. Dans le milieu professionnel, la formation des salariés se retrouve être au cœur des préoccupations des services de ressources humaines, avec pour question principale «comment motiver et responsabiliser les salariés dont les formations ne sont pas forcément adaptées à l'évolution de l'entreprise ?» (Chomienne et al., 2008). L'e-learning devient de plus en plus accessible à un plus grand nombre d'entreprises. L'offre est de plus en plus variée et large. Dans le secteur bancaire par exemple, l'exploitation du e-learning s'explique notamment par l'évolution rapide des produits et des services nécessitant des efforts de formation massifs.

Selon le baromètre Européen de l'e-learning mené par CrossKnowledge (Fiehl et al., 2010), nous constatons que le nombre de salariés ayant suivi au moins une formation en e-learning est plus important dans les grandes entreprises et dans le secteur des services, alors que dans le secteur de l'industrie les taux restent assez faibles. Dans ce rapport, CrossKnowledge prédit une forte progression du e-learning à l'horizon 2020. En

14. http ://etudiant.lefigaro.fr/les-news/actu/detail/article/moocs-seuls-5-des-inscrits-valident-leurs-formation-4534/
15. http ://www.slate.fr/story/78916/mooc-cours-internet-spoc

effet, le pourcentage d'entreprises qui n'utilisent pas ou peu l'e-learning (<10% des salariés formés) a nettement reculé entre 2010 et 2014, passant de 37% à 15% en quatre ans (figure 1.1). Le e-learning se généralise en s'étendant à un cercle croissant de salariés dans l'entreprise, avec comme effet attendu un boom du nombre d'entreprises qui vont former entre 10 et 50% de leurs salariés (passage de 30 à 45% de l'échantillon entre 2010 et 2014).

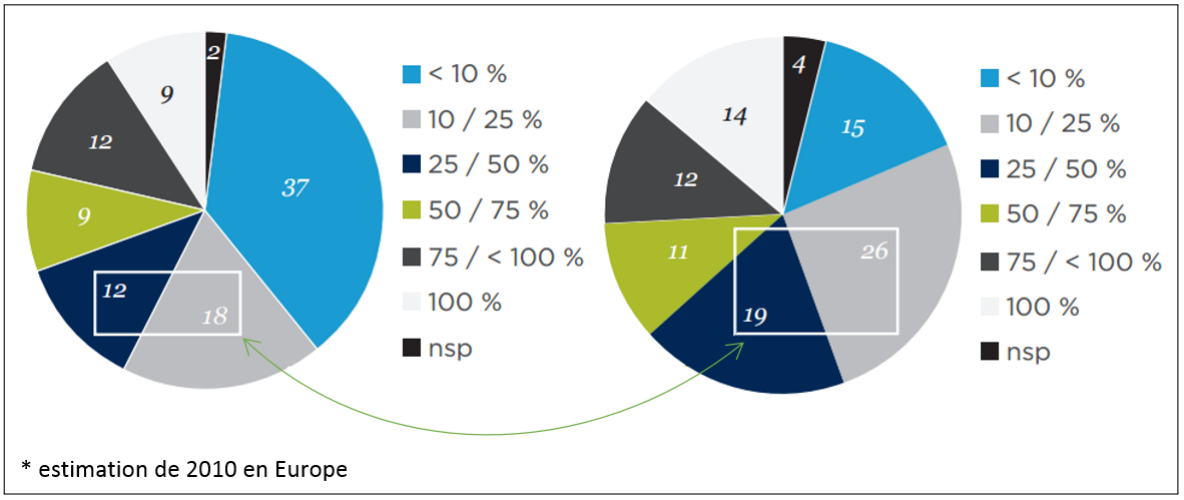

FIGURE 1.1 – Pourcentage des salariés ayant suivi une formation e-learning

Cette rapide évolution du e-learning, aussi bien dans les universités que dans le domaine industriel, est vite rattrapée par l'évolution des nouvelles technologies : l'informatique devient ubiquitaire, pervasive, mobile, et l'utilisation des réseaux sans fil émerge et rend les objets mobiles de plus en plus intelligents. Déjà en 2004, quelques statistiques mettaient en avant l'importance de l'utilisation de ces nouvelles technologies pour l'apprentissage en europe (Georgiev et al., 2004) :

– Plus de 50% des lieux de travail sont mobiles.
– Plus de 1 milliard d'utilisateurs se servent d'une connexion internet sans fil.
– Partout dans le monde sont vendus plus de 500 millions de téléphones portables pouvant se connecter à Internet et le nombre de PDA (Personal Digital Assistant) et de Smartphones vendus a augmenté de plus de 200%.

Quelques années plus tard, l'Union Internationale des Télécommunications (UIT), qui analyse l'évolution des TIC, publie dans une étude portant sur 154 pays en 2009[16], les dernières statistiques récoltées concernant l'usage des différentes technologies de communication y compris les dispositifs mobiles (figure 1.2). L'étude révèle notamment l'abandon petit à petit de l'usage des téléphonies fixes et de la connexion Internet fixe en faveur des nouvelles technologies mobiles.

16. http ://www.itu.int/ITU-D/ict/material/Telecom09_flyer.pdf

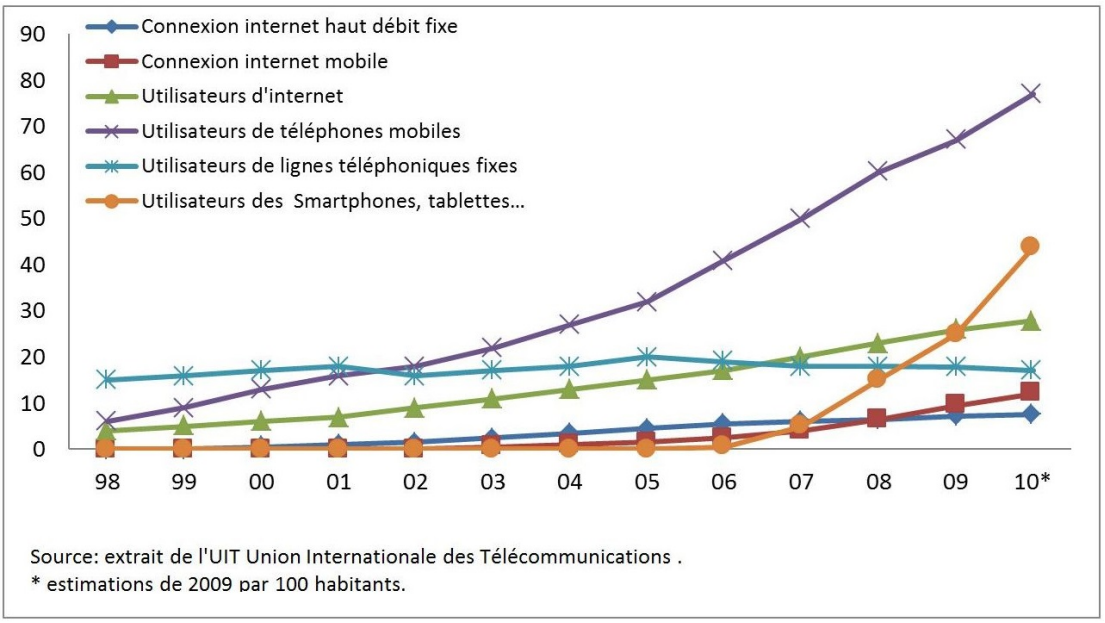

FIGURE 1.2 – Évolution de l'usage des technologies de télécommunication et mobiles

Une étude plus récente présentée en 2013 par Lux Research [17], montre une nette évolution de l'utilisation des technologies mobiles (figure 1.3). Des appareils mobiles tels que les Smartphones et les tablettes ont créé une révolution dans la façon dont les utilisateurs de se connecter, de travailler et de se former.

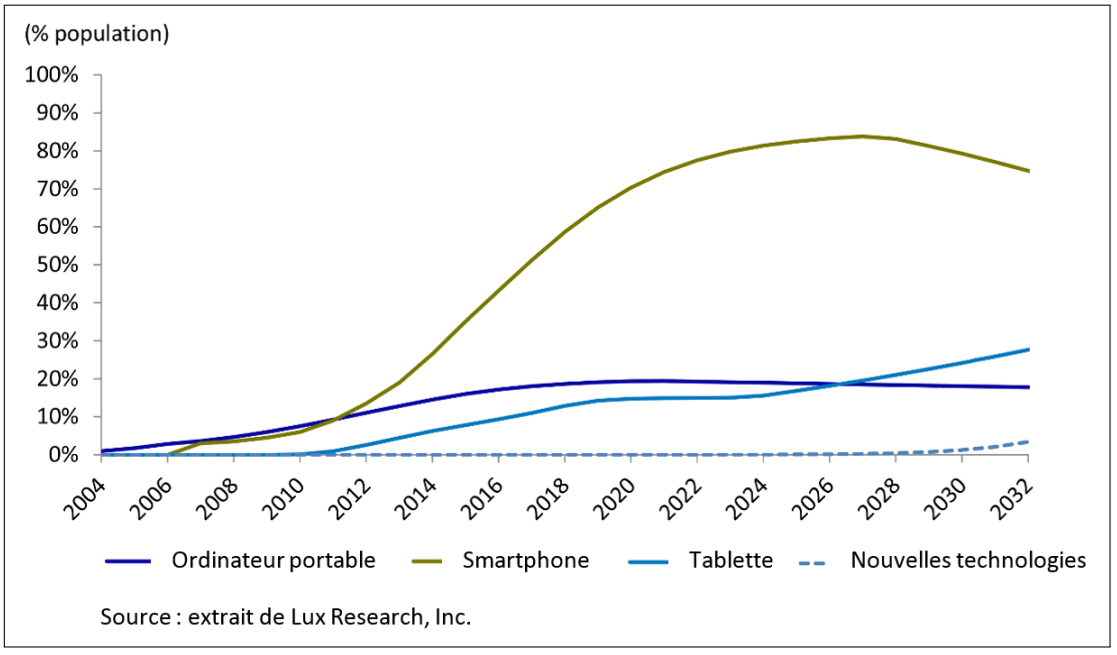

FIGURE 1.3 – Évolution de l'usage des dispositifs mobiles

L'informatique mobile est essentiellement fondée sur la capacité à déplacer physiquement un environnement informatique avec l'utilisateur. De nombreux utilisateurs ne possèdent

17. www.luxresearchinc.com

plus uniquement des ordinateurs de bureau ou mobiles, ils possèdent désormais aussi des PDA, des lecteurs MP3, des Smartphones, etc. Sharples précise : «*we are now entering the mobile age, where phones are carried everywhere, banks are accessed from holes in the wall, cars are becoming travelling offices, airplane seats are entertainment centres, computer games are handheld, and advertising is ubiquitous*» (Sharples, 2006). Le m-learning tire profit de l'informatique omniprésente pour proposer de nouvelles méthodes d'apprentissage.

Dans la section suivante nous discutons de différentes propositions de définitions du m-learning existantes, et nous proposons la nôtre, adaptée à notre cadre de recherche.

1.1.3/ DÉFINITION DU M-LEARNING

Le terme m-learning a différentes définitions selon les communautés scientifiques. Depuis la naissance du terme m-learning, la définition de celui-ci n'a jamais arrêté d'évoluer. Jusqu'aux récentes conférences spécialisées dans le domaine, telle que dans IADIS International Mobile Learning Conference 2013, la définition du m-learning reste un point de discussion pour les chercheurs et les experts du métier.

Sharples a identifié trois directions dans les évolutions de la définition du m-learning : la première est focalisée sur les outils, la deuxième sur l'apprentissage hors des murs (d-learning, e-learning), et la troisième sur la mobilité des apprenants où l'apprentissage se déroule dans un contexte (Sharples, 2006) (Endrizzi, 2011) :

– *Première direction :* Les premières définitions du m-learning apparaissent centrées essentiellement sur les nouvelles technologies électroniques mobiles (Pinkwart et al., 2003). Ces technologies sont considérées comme des outils de stockage et de transport des contenus multimédias. Le m-learning est alors vu comme un prolongement logique du e-learning sur des technologies mobiles (figure 1.4).

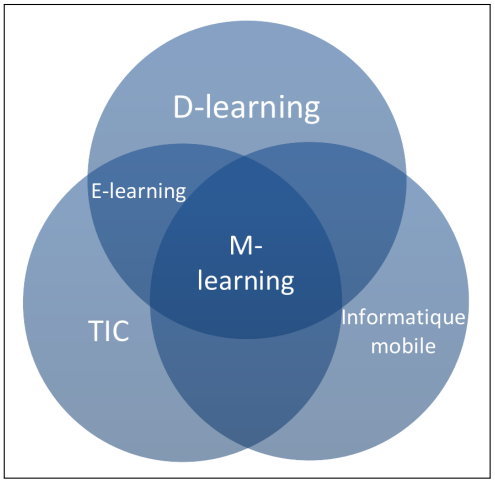

FIGURE 1.4 – Cartographie du m-learning, version 1

Quinn déclare «*mobile learning is e-learning through mobile computational devices : Palms, Windows CE machines, even your digital cell phone*» (Quinn, 2000). Cette définition est considérée exacte mais inutile, car Quinn ne cherche qu'à placer l'apprentissage mobile quelque part sur le spectre de portabilité de l'e-learning (Traxler, 2005). Le projet Learning2go [18] lancé en 2008 illustre bien cette première direction. Les plus récentes définitions de m-learning ont transféré le centre d'intérêt des technologies mobiles à la mobilité des apprenants.

– *Deuxième direction :* Les définitions se concentrent sur les potentialités en termes d'apprentissage à l'extérieur de l'espace physique classique délimité par les murs de la salle de formation (un voyage d'études, une sortie scolaire telle que la visite d'un musée, etc.) (Sharples et al., 2009). En ce sens, O'Malley propose sa définition : «*mobile learning takes place when the learner is not at a fixed, predetermined location, or when the learner takes advantage of the learning opportunities offered by mobile technologies*» (O'Malley et al., 2005). Il s'agit d'une nouvelle méthode, dont les grands principes sont de tirer inspiration de l'environnement de l'apprenant pour proposer des exercices narratifs et ludiques. Représentatif de cette direction, en 2009, un jeu nommé The Island [19] a été développé entre Amsterdam et New York : des étudiants de ces deux villes, connectés entre eux grâce à leurs portables, ont participé à une chasse au trésor dans les rues des deux cités, grâce à une application mêlant géo-localisation, cartes historiques et éléments architecturaux.
– *Troisième direction :* Dans cette dernière direction, les projets ne se focalisent plus ni sur les outils utilisés par les apprenants, ni sur les activités. Cette phase se focalise essentiellement sur la localisation de l'apprenant dans un contexte spécifique grâce aux technologies de capture de localisation, comme les systèmes GPS, les réseaux sans fil, les méthodes de détection de proximité, etc. afin d'analyser ce qui se passe autour de l'apprenant. En d'autres termes, en reliant les activités d'apprentissage à des informations contextuelles, les apprenants peuvent découvrir de nouvelles connaissances et explorer des concepts liés à leur environnement. C'est parce que l'on est dans un contexte particulier qu'une activité utilisant un équipement mobile prend tout son sens et devient pertinente, «*mobile devices give us a unique opportunity to have learners embedded in a realistic context at the same time as having access to supporting tools*» (Naismith et al., 2005). Représentatifs de cette direction, on peut citer les deux projets MOBILearn [20] et MARA [21] (Mobile Augmented Reality Applications). Le projet MOBILearn propose un outil embarquant un ensemble de fonctionnalités combinant capture, stockage et partage de contenus, tout en prenant en compte l'environnement physique de l'apprenant (des fonctionnalités liées à la géo-localisation sont proposées). Dans le projet MARA, il s'agit de superposer des informations graphiques numériques à des images réelles. Le dispositif d'apprentissage devient un outil avec

18. http ://www.learning2go.org/
19. https ://waag.org/en/project/island
20. http ://www.mobilearn.org/
21. https ://research.nokia.com/page/219

lequel l'apprenant accède à un monde réel augmenté d'informations nouvelles.

Dans le cadre de notre projet de recherche, nous proposons une nouvelle définition du m-learning, de manière à intégrer ce dernier dans les trois directions proposées par Sharples. Notre définition est la suivante :
«*Le m-learning est l'adaptation de modules d'apprentissage à des technologies mobiles en ayant une meilleure connaissance de l'apprenant (profil) et une meilleure connaissance de son environnement (contexte)*».
Nous proposons dans la figure 1.5, à partir de notre définition, une nouvelle cartographie du m-learning.

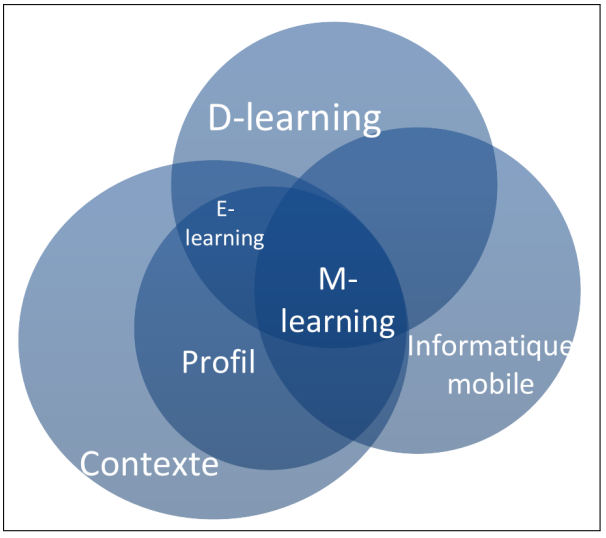

FIGURE 1.5 – Cartographie du m-learning, version 2

A travers ces différentes définitions nous pouvons dégager plusieurs caractéristiques récurrentes du m-learning, telles que l'attractivité, la flexibilité, l'accessibilité et l'informalité. Nous développons plus en détail ces caractéristiques dans la section suivante.

1.1.4/ CARACTÉRISTIQUES DU M-LEARNING

Attractivité, flexibilité, accessibilité et informalité sont les concepts phares qui permettent de synthétiser l'ensemble des avantages du m-learning.

Attractivité :
D'après Traxler, tout tend à montrer que le m-learning est motivant et attrayant, en particulier auprès de ceux que l'apprentissage classique rebutait (Traxler, 2007). En effet, en donnant une formation m-learning à quelqu'un qui se montre d'habitude réticent et sans enthousiasme, le comportement de ce dernier change ; il manifeste une véritable envie et un plaisir à apprendre.

Flexibilité :
Deux grands types de flexibilité offerts par le m-learning peuvent être identifiés [22] " :
– *Organisationnelle :* Dans le temps et dans l'espace (où je veux et quand je veux).
– *Cognitive et pédagogique :* Adaptation, personnalisation, modularisation, just-in-time, polyvalence didactique (ce dont j'ai besoin, comme j'en ai besoin et quand j'en ai besoin)".

Le découpage d'un contenus pédagogique en modules courts permet d'étaler la formation dans le temps et ainsi d'avoir une meilleure assimilation des connaissances. En outre, une formation m-learning permet aux apprenants de consulter les contenus autant de fois qu'ils le souhaitent, contrairement à une formation en présentiel. Tous ces facteurs permettent un assouplissement de l'apprentissage et offrent aux apprenants la possibilité de se former à leur rythme en organisant eux-mêmes leurs formations.

Accessibilité :
Cinq grands types d'accessibilité offerts par le m-learning peuvent être identifiés [23] " :
– *Accessibilité temporelle et géographique :* Permettre l'accès à des personnes soumises à des contraintes diverses, distantes géographiquement, vivant dans des fuseaux horaires différents ou tout simplement possédant des agendas chargés ou une liberté organisationnelle variable.
– *Accessibilité socio-culturelle :* S'adapter facilement à des réalités culturelles et sociales variées. Par exemple, le m-learning permet à des gens issus de milieux défavorisés, comme les habitants du tiers-monde, d'accéder à un enseignement véhiculé par la technologie et de faire fructifier ce nouveau savoir, ce qui était jusqu'ici inenvisageable pour eux (exemple, Dr Maths est un service d'apprentissage en ligne d'Afrique du Sud qui aide les élèves du secondaire en maths).
– *Accessibilité socio-économique :* Permettre une accessibilité à un grand nombre d'utilisateurs, à un coût faible.
– *Accessibilité technique :* Permettre une accessibilité moyennant des compétences techniques basiques et des accès simples au niveau ergonomique.
– *Accessibilité intellectuelle :* Prolonger l'accessibilité en s'adaptant à divers styles d'apprentissage et compétences à développer. Cette nouvelle méthode d'apprentissage permet de rendre accessible la formation à des personnes qui ne pouvaient pas assister à des formations en présentiel (apprenants à mobilité réduite, apprenants distants géographiquement, etc.) et de créer des formations visant des compétences et des besoins particuliers".

Informalité :
L'apprentissage est dit formel lorsqu'il se passe dans le cadre d'un établissement d'en-

22. http ://www.awt.be/web/edu/index.aspx ?page=edu,fr,gui,080,010
23. http ://www.awt.be/web/edu/index.aspx ?page=edu,fr,gui,080,010

seignement. Hors de ce cadre, l'apprentissage est considéré comme informel. Livingstone(Livingstone, 2001) définit l'apprentissage informel comme «*(1) a new significant form of knowledge, understanding or skill acquired outside a prescribed curricular setting, and (2) the process of acquisition, either on your own initiative in the case of self directed informal learning, or with aid of a mentor in the case of informal training*». On déduit de cette définition que l'apprentissage informel a un caractère non intentionnel. Dans un environnement mobile, l'apprenant peut apprendre en situation de mobilité, à la maison, dans le train, au bureau, sur le lieu de travail, au travers des temps, de l'espace, n'importe quand et n'importe où (Pham, 2010). Le m-learning découle de la vie quotidienne (Bomsdorf, 2005). Les résultats acquis de l'apprentissage informel ne conduisent généralement pas à une certification, mais ils peuvent être validés et certifiés dans le cadre de programmes de validation des acquis par un formateur. Le m-learning est alors considéré comme apprentissage informel en complément et renforcement de l'apprentissage formel et vice et versa. Nous pouvons citer comme exemples de projets qui soutiennent l'apprentissage informel, le projet T2T (Text to Teach) [24] et le projet BridgeIT [25]

Les lacunes du m-learning :

Ces différentes caractéristiques attestent de la focalisation sur l'apprentissage davantage que sur le contenu et sur la prise en compte du profil et du contexte de l'apprenant. En effet dans une formation m-learning l'apprenant se trouve souvent confronté à certains problèmes qui peuvent mener à l'échec de la formation. Parmi ces problèmes nous citons [26] " :

- *L'apprenant est submergé d'informations :* La tentation est parfois forte pour le formateur de délivrer dans les modules en ligne un grand nombre de contenus. Si l'information est trop dense ou mal structurée, l'apprenant risque de se perdre dans ces informations. C'est ce qu'on appelle la surcharge cognitive.
- *L'apprenant a peu de contrôle sur les contenus :* "Le contenu des modules d'apprentissage est souvent mis en ligne arbitrairement par les formateurs. Les apprenants n'ont donc pas la possibilité de le manipuler ou de le transformer facilement en fonction de leurs propres besoins.".
- *Les contenus ne s'adaptent pas à l'environnement extérieur :* Certains environnement sont peu adaptés à des apprentissages qui nécessitent calme et concentration. Par exemple, un aéroport ou un train peuvent constituer des environnements peu propices à l'assimilation de certains contenus qui nécessitent un certain niveau de concentration.
- *Confusion entre vie privée et professionnelle :* Les apprenants ayant accès à des formations de type m-learning ne sont pas souvent disposées à consacrer du temps pour se former lorsqu'ils sont chez eux au lieu de se charger de leurs activités quotidiennes.

24. http ://www.text2teach.org.ph/ ?page_id=59
25. http ://bridge-it.communicationproject.eu/The-e-learning-Platform.htm
26. http ://www.formavox.com/elearning-avantages-inconvenients-top-5-flop-5

– *Trop d'outils et manque de pérennité :* Il existe aujourd'hui plusieurs outils à disposition des formateurs pour créer des contenus e-learning et à disposition des apprenants accéder à ces contenus (e.g. plateformes LMS, médias sociaux, podcasts). Difficile donc parfois de s'y retrouver et d'avoir la garantie que ces contenus tiendront dans la durée."
– *Freins à l'usage professionnel :* Lors d'une formation dans le cadre du travail, le m-learning peut être vu comme une tentative de la part de l'entreprise de profiter de périodes où l'employé n'est pas dans les locaux de la société, pour poursuivre la formation, à son domicile.
– *Cadre peu adapté pour l'apprentissage :* Certains environnement sont peu adaptés à l'apprentissage qui nécessite confort, calme et concentration. Un aéroport ou un train peuvent constituer des environnements peu propices à l'assimilation de certains contenus. Le m-learning est donc a priori peu adapté aux contenus théoriques complexes.

Un système m-learning efficace devrait donc adapter son fonctionnement et ses contenus pour la réussite de la formation. Dans la section suivante, nous présentons et comparons quelques travaux qui ont été réalisés dans le cadre du m-learning, afin de déterminer les points forts et les points faibles des différentes propositions existantes.

1.2/ PLATEFORMES M-LEARNING

Il existe de nombreux projets s'intéressant à l'évolution des TIC et leur intégration dans le domaine de l'apprentissage. IAmLearn [27](International Association for Mobile Learning) propose un large catalogue descriptif des différents projets m-learning existants. Dans cette section, nous analysons les projets les plus connus dans le domaine du m-learning afin d'illustrer les caractéristiques décrites dans la section précédente.

Projet MoMaths :
Dans les pays en développement, un grand nombre des nouvelles technologies éducatives font usage de MXit [28], le principal réseau social africain sur plateforme mobile. Lancée en mai 2005, MXit revendique environ 30 millions d'abonnés et est le premier logiciel de messagerie instantanée créé spécifiquement pour être utilisé depuis un téléphone portable.
Cette approche est idéale dans les pays où Internet et les ordinateurs sont encore peu accessibles. C'est l'idée sous-jacente de plusieurs programmes éducatifs mobiles comme le projet MoMaths [29] de Nokia développé par MXit Lifestyle, une plateforme de services facilitant l'apprentissage des mathématiques sur mobiles. Elle propose de nombreuses fonc-

27. http ://www.iamlearn.org/resources/projects
28. http ://www.nextafrique.com/sciencestech/analyses-sciencestech/2182-comment-le-telephone-mobile-pourrait-revolutionner-l-education-en-afrique
29. http ://momaths.nokia.com/

tionnalités basées sur les programmes scolaires de plusieurs niveaux (théorie, exercices, tutorat, entraînement aux examens, etc.). En Afrique du Sud, MoMaths a fait ses preuves dans 200 établissements, auprès de 50.000 élèves dont les résultats en mathématiques sont supérieurs à ceux n'utilisant pas cet outil.
Ce projet est plus axé sur l'accessibilité des ressources d'apprentissage dans des milieux défavorisés que sur l'adaptation des ressources au profil de l'apprenant et à son contexte.

Projet Ensemble :
Le projet Ensemble [30] s'inscrit dans le programme européen Lifelong Learning Program [31] co-financé par la communauté européenne et coordonné par l'Université de Florence [32] dont l'objectif principal est le développement de l'apprentissage en ligne. Le projet a été lancé en 2010 et a développé une stratégie pour l'utilisation des TIC dans le but de promouvoir l'intégration culturelle et sociale des citoyens immigrés.
L'idée est de tirer parti de technologies largement utilisées, telles que les lecteurs MP3 et les téléphones mobiles, et de tester des méthodes d'enseignement et des formats de communication adaptés à ces appareils. Le projet s'adresse à un groupe de collégiens issus de classes au pourcentage élevé d'immigrés ainsi qu'à leurs parents (le projet a été testé en France sur deux classes de 3 ème de deux collèges des Yvelines : à Mantes-la-Jolie et aux Mureaux). Il leur a été diffusé, à un rythme régulier, des contenus sur des thèmes précis, à partir d'une plateforme de diffusion automatique et synchronisée.
L'intérêt de ce projet porte sur les aspects pédagogiques en direction des collégiens en difficulté, mais également sur la construction de liens plus étroits avec les parents, ainsi associés au suivi des études de leurs enfants grâce à une approche de formation innovante et multimodale.

Projet DRIM-AP :
L'objectif du projet DRIM-AP (Mercier et al., 2004) (Dispositifs Radios Interactifs Multiples et Amphis Participatifs) mené à l'école centrale de Lyon est de rendre l'enseignement dans les amphithéâtres plus vivant et participatif.
"Dans un amphithéâtre interactif, chaque étudiant dispose d'un terminal mobile (PDA, PC portable, Smartphones, etc.) communiquant sur réseau wifi. Les enseignants de leur côté disposent d'un ordinateur portable pour superviser les échanges avec les étudiants, ces ordinateurs sont aussi d'outils pour l'évaluation du niveau de connaissance ou de compréhension des étudiants sur les sujets abordés".
Dans ce projet l'apprentissage reste formel. La dimension mobilité du projet consiste à transférer des contenus pédagogiques sur des dispositifs mobiles. L'adaptation au contexte n'est pas prise en compte.

30. http ://www.ensembleproject.org/
31. http ://eacea.ec.europa.eu/llp/
32. http ://www.unifi.it/

Projet MOBILearn :
Le projet MOBILearn lancé en 2002 est un projet européen de recherche et de développement (Beale and Lonsdale, 2004) qui a pour objet d'explorer l'utilisation des environnements mobiles pour favoriser l'apprentissage informel, l'apprentissage par résolution de problèmes et l'apprentissage au travail.
Dans le cadre de ce projet, une architecture pour l'apprentissage mobile a été réalisée. Elle permet de générer des contenus et des services pour accompagner un apprenant pendant ses activités d'apprentissage dans une galerie d'exposition ou dans un musée. L'apprentissage est supporté par un ensemble d'activités lors des visites des musées. Les visiteurs placés devant une œuvre peuvent utiliser un PDA ou un Smartphone pour obtenir des informations pertinentes, tout en observant l'œuvre. De plus, si un apprenant visite un musée pour la seconde fois, il peut obtenir des informations différentes qui prennent en compte sa visite précédente.

Projet MoULe :
Le projet MoULe [33] (Mobile and Ubiquitous Learning) qui a obtenu Le Gold Award à la conférence mLearn 2009, a été développé par le Conseil National Italien de recherche (Institut de Technologies Educatives). MoULe propose un environnement en ligne pour l'apprentissage collaboratif qui fonctionne en intégrant les téléphones portables et les Smartphones, favorisant ainsi les activités pédagogiques basées sur l'exploration de lieux géographiques.
Le système inclut des fonctionnalités spécifiques pour rechercher et accéder à des espaces d'information, pour communiquer et annoter des lieux selon leurs coordonnées géographiques. Les données sont stockées dans un système, y compris les données déposées par les autres utilisateurs, et sont enrichies par les informations concernant les localisations géographiques, pour que le système fournisse à l'utilisateur des informations spécifiques sur les lieux à visiter. Les administrateurs peuvent utiliser MoULe pour concevoir des parcours aussi bien touristiques que pédagogiques, animer les activités de l'utilisateur en temps réel, et évaluer la quantité et la qualité des interactions entre les utilisateurs.

Projet AMULETS :
Le projet AMULETS [34] (Advanced Mobile and Ubiquitous Learning Environments for Teachers and Students) développé à l'université de Linnaeus a été lancé en 2012. Il a pour objectif de concevoir un système permettant aux enseignants d'élaborer et de mettre en œuvre de nouveaux scénarios d'enseignement.
AMULETS utilise comme outil mobile un Smartphone de type NoKia 6630 muni d'un GPS. Il est utilisé pour la génération et l'affichage de contenus, l'envoi de messages et le décodage d'étiquettes de code-barres 2D. De plus AMULETS propose un ensemble d'ac-

33. http ://moule.pa.itd.cnr.it/
34. http ://amulets.blogg.se/

tivités d'apprentissage afin d'encourager la collaboration des différents groupes d'élèves avec les enseignants. L'apprentissage prend la forme d'un jeu collaboratif. Ce dernier est constitué d'un ensemble de missions que les élèves doivent découvrir et accomplir. Ces missions ont lieu dans différents endroits et à différentes périodes de temps. Les activités ainsi conçues contiennent des défis, à découvrir par les élèves, dans lesquels ils ont besoin de collaborer et d'appliquer des stratégies de résolution de problème.

Projet InTouch :
Le projet InTouch [35] est un projet européen, en cours de réalisation, qui a pour objectif de définir une approche innovante dans le cadre des entreprises : permettre de développer les principales compétences inhabituelles des employés.
Le projet vante une approche flexible, qui utilise les possibilités offertes par les nouvelles technologies, et qui répond efficacement aux nouvelles attentes du marché du travail. Afin de couvrir les besoins des employés, les partenaires du projet InTouch sélectionnent les supports pédagogiques les plus flexibles, à savoir les appareils mobiles, ainsi qu'une solution d'apprentissage intéressante, les jeux sérieux (serious games). Ces jeux présentent des exemples d'apprentissage situationnels liés à des compétences inhabituelles identifiées dans le cadre d'un projet, et emploient différents types d'interactions en fonction du thème abordé. Pour élargir leurs connaissances, les utilisateurs peuvent partager leurs points de vue personnels et leurs expériences au sein de la communauté [36].

Synthèse sur les plateformes m-learning :
Bien que ces différents projets respectent les caractéristiques d'attractivité, de flexibilité, d'accessibilité et d'informalité, ils ne proposent pas d'adaptation de contenus en fonction du profil et du contexte de l'apprenant. Nous décrivons dans la section suivante le m-learning comme une activité en contexte.

1.3/ LE M-LEARNING, UNE ACTIVITÉ EN CONTEXTE

Le m-learning est souvent considéré comme une spécialisation du e-learning. C'est plutôt une extension à de nouvelles formes d'environnements d'apprentissage. Le m-learning permet d'apprendre à travers des contextes : «*mobile learning is not just about learning using portable devices, but learning across contexts*» (Sharples, 2006). Ici l'accent est mis sur la manière dont les apprenants se forment au travers des lieux et des transitions entre différents contextes.
Il ne doit pas y avoir de rupture dans l'apprentissage entre le présentiel et l'extérieur. L'apprentissage est fondé sur la continuité de l'activité au travers de l'espace et du temps

35. hhttp ://www.intouch-ict.eu/
36. https ://www.adam-europe.eu/prj/8980/prd/14/1/D%207.5%20Article%20for%20sensitisation_FR.pdf

en interagissant avec des technologies mobiles et fixes. L'apprentissage ne doit pas être limité à certains environnements, mais doit accroitre la mobilité de l'apprenant au travers ces derniers.
Avec l'apparition et l'évolution des nouvelles technologies mobiles, l'adaptation au contexte est devenue un caractère indispensable des nouveaux systèmes informatiques pour une utilisation nomade. Il faut donc dans le cas d'une adaptation au contexte en apprentissage, déterminer selon le contexte de l'apprenant quelles sont les contenus à envoyer, de quelle manière, sur quel outil, etc. Tout le processus d'apprentissage doit s'adapter à ces changements de contexte. D'un autre côté, l'apprentissage au travers des contextes nécessite une suite organisée d'activités, c'est-à-dire que l'apprentissage ayant lieu dans un contexte précis dépend de ceux qui ont eu lieu avant. Il faut donc que le système prenne en compte l'historique de l'apprentissage pour proposer à l'apprenant des activités d'apprentissage pertinentes et ainsi poursuivre ses activités au travers de contextes.

1.4/ CONCLUSION

Ce chapitre avait pour objectif de mieux comprendre le domaine de l'apprentissage en situation de mobilité. L'étude des différentes définitions du m-learning nous montre que l'apprentissage mobile est caractérisé essentiellement par l'attractivité, la flexibilité, l'accessibilité, l'informalité. L'apprentissage mobile peut avoir lieu au travers de contexte différents et dynamiques, grâce aux technologies mobiles. A travers l'analyse des travaux existants, nous avons un aperçu des possibilités offertes par les environnements m-learning en fonction de besoins particuliers des apprenants. La plupart de ces travaux permettent un apprentissage dans des localisations différentes : en classe, en entreprise, à l'extérieur, dans un musée, etc. Cependant le m-learning reste limité car il n'est pas capable de s'adapter à son environnement. Le contexte n'est pas explicitement représenté dans le système, mais requis par celui-ci pour un apprentissage en situation. L'adaptation au contexte doit permettre de dépasser cette limite.
Le chapitre 2 introduit la notion de contexte et de sensibilité de contexte pour l'adaptation de parcours pédagogiques en situation de mobilité.

2

CONTEXTE ET SENSIBILITÉ AU CONTEXTE

Dans le domaine de l'informatique, la prise en compte du contexte dans une analyse ou une réponse programmée n'est pas nouvelle. Dès les années soixante, systèmes d'exploitation, théorie des langages et intelligence artificielle exploitent déjà cette notion pour adapter leur comportement et leur réponse à l'environnement d'exécution. Ces 10 dernières années, avec l'émergence massive de l'informatique mobile, ubiquitaire et ambiante, elle est placée au cœur des débats. De nombreux travaux de recherche ont été menés sur la modélisation du contexte. Bien que cette notion soit universelle, chacun possède sa propre perception du contexte. Un contexte est généralement décrit comme un ensemble d'éléments qui peuvent influencer la compréhension d'une information.
Dans la première partie de ce chapitre, nous présenterons les différentes visions de la notion du contexte en essayant d'en faire ressortir des traits de caractérisation que nous appellerons dimensions. La combinaison de ces dimensions permet de construire des contextes spécifiques pour des besoins clairement identifiés. Au-delà de la capacité de représenter un contexte, il faut être capable d'exploiter ce contexte. La construction d'un système dont les réponses sont influencées par le contexte, un système sensible au contexte, est un travail complexe. L'étude de ce domaine à aussi donné lieu a de nombreux travaux. Dans la seconde partie de ce chapitre, nous présenterons ce domaine en essayant, là encore, d'extraire les principales fonctionnalités requises pour prendre en charge ce contexte.

Sommaire

2.1 Contexte 32
2.1.1 Définition du contexte 32
2.1.2 Dimensions du contexte 35
2.1.3 Acquisition des informations de contexte 38
2.1.4 Modélisation du contexte 39
2.2 Systèmes sensibles au contexte 43
2.2.1 Définition de la sensibilité au contexte 43

2.2.2 Architecture générale d'un système sensible au contexte 44
2.2.3 Exemples d'architectures de systèmes sensibles au contexte . . 46
2.3 Conclusion 54

2.1/ CONTEXTE

La notion de contexte est assez universelle. Elle désigne l'ensemble des éléments qui peuvent influencer la compréhension d'une situation particulière. Cette description est ensuite utilisée pour mieux comprendre l'environnement.
Il existe de nombreuses définitions du contexte. Dans cette section nous proposerons d'organiser ces définitions selon un ordre chronologique, montrant une progression dans la compréhension de la notion de contexte. Ensuite nous proposerons une classification des différentes informations contextuelles en dimensions, en termes de caractéristiques, propriétés et utilisations. La prise en compte du contexte nécessite une représentation explicite de ce dernier dans le système. Nous présenterons et analyserons en dernier lieu les méthodes de modélisation du contexte, en nous basant sur l'organisation en dimensions et les caractéristiques du contexte.

2.1.1/ DÉFINITION DU CONTEXTE

Avant de présenter les définitions du contexte proposées dans la littérature scientifique, nous commençons par introduire et analyser des définitions proposées dans différents dictionnaires :

- *Techno-Science*[1], encyclopédie scientifique en ligne : « *Le contexte d'un évènement inclut les circonstances et conditions qui l'entourent* ». En informatique, « *le contexte est l'ensemble des conditions sous lesquelles un dispositif est en train d'être utilisé, par exemple l'occupation actuelle de l'utilisateur* ».
- *Grand Dictionnaire Technologique*[2] *(GDT)* : « *Texte entourant un terme, ou situation dans laquelle ce terme est employé* ». En informatique, « *ensemble d'informations concernant l'action du stylet, en rapport principalement avec sa localisation à l'écran, qui permet au système d'exploitation de l'ordinateur à stylet de différencier les commandes et l'entrée des données, et de fonctionner en conséquence* ».
- *Centre National de Ressources Textuelles et Lexicales*[3] : « *Ensemble de circonstances liées, situation où un phénomène apparaît, un événement se produit* ». En informatique, « *le contexte est l'ensemble d'informations concernant un évènement en rapport principalement avec sa localisation, qui permet au système informatique de fonctionner et de s'adapter en conséquence* ».

1. http ://www.techno-science.net/
2. http ://gdt.oqlf.gouv.qc.ca/
3. http ://www.cnrtl.fr/

Un point commun émerge de ces définitions. Elles décrivent toutes le contexte comme un ensemble d'informations associées à quelque chose « ensemble de ... qui entoure/constitue/concernant/liés/etc. quelque chose ». La nature de ce quelque chose (un évènement, un texte, une action, un phénomène, une situation, etc.) dépend de l'usage dans un domaine d'application (linguistique, informatique, etc.). Cet usage permet de spécifier comment le contexte est utilisé (modifier la valeur des éléments, différencier les commandes, ou fonctionner en conséquence) (Pham, 2010).

En informatique, selon Schilit dans (Schilit and Theimer, 1994), le contexte se réfère à la localisation et l'identité des personnes et des objets à proximité ainsi que les modifications pouvant intervenir sur ces objets. Dans un autre travail, Schilit reprend sa première définition et propose une méthode pour déterminer le contexte en répondant à trois questions : *où êtes-vous ?*, *avec qui êtes-vous ?* et *quelles sont les ressources à proximité de vous ?* (Schilit et al., 1994). Quelques années plus tard, (Brown et al., 1997) introduit l'heure, la saison, la température, l'identité et la localisation de l'utilisateur comme nouveaux éléments du contexte.
En fait, dans ces premières définitions du contexte, les auteurs limitent le contexte à l'utilisateur et son environnement. Les travaux de recherche plus récents essayent alors d'établir une définition plus générale et plus claire du contexte.

Abowd fournit une définition plus générale du contexte, en le considérant comme des informations qui entourent l'entité ou l'élément sur lequel un système se focalise, « *any information that can be used to characterize the situation of an entity* » (Abowd et al., 1999). Une entité ou un élément peut être interprété différemment par différents objectifs d'utilisation : (1) Une entité, dans un domaine centré application logicielle, est une fonction, un événement d'un logiciel ou le logiciel lui-même. Le contexte est alors constitué d'un ensemble de paramètres et d'états de l'environnement, utilisés pour déterminer cette entité (Chen and Kotz, 2000). (2) Une entité peut être un dispositif. Dans ce cas, le contexte est un environnement physique et social dans lequel ce dispositif est utilisé, « *physical and social situation in which computational devices are embedded* » (Moran and Dourish, 2001). (3) Dans un système centré utilisateur, les entités impliquées dans des interactions sont divisées en trois catégories : les lieux (bâtiment, pièce, etc.), les personnes (individus ou groupes d'individus) et les objets (objets physiques, ordinateurs, etc.) (Dey et al., 2001).
Pour résumer, une information fait partie du contexte si elle influe sur une interaction entre deux entités : « *context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves* » (Dey, 2000).
Cette dernière définition concerne particulièrement la conception des systèmes sensibles au contexte, que nous introduirons dans la deuxième partie du chapitre. Chaari considère que cette définition est incomplète car elle ne précise pas ce qui compose le contexte. Il

propose alors une nouvelle définition pour apporter plus de précision à celle de Dey : « *le contexte est l'ensemble des paramètres externes à l'application qui peuvent influer sur son comportement en définissant de nouvelles vues sur ses données et ses fonctionnalités. Ces paramètres ont un aspect dynamique qui leur permet d'évoluer durant le temps d'exécution* » (Chaari et al., 2005).

Nous considérons que cette définition est la plus adaptée pour la conception de notre système m-learning sensible au contexte, car elle apporte plus de précisions sur le contexte. Elle permet de voir le contexte comme un ensemble d'éléments susceptibles d'influencer le comportement d'un système et de modifier le résultat produit en conséquence.

A partir de cette définition nous constatons aussi que le contexte n'est pas stable, il est dynamique et continuellement en évolution. Syvanen développe cette propriété du contexte en déclarant « *context is continually constructed through negotiation between communicating partners (including humains and interactive technology) and the interplay of activities and artifacts* » (Syvanen et al., 2005).

Les caractéristiques qui décrivent le contexte sont essentiellement les suivantes :

– *L'imprévisibilité :* Les changements de l'environnement et les interactions de l'utilisateur sont imprévisibles. En conséquence certains éléments du contexte peuvent ne pas être connus et décrits à l'avance dans le système. L'une des principales questions qui se pose est de savoir comment un système doit prendre en compte ces nouveaux éléments et les mettre en œuvre dans le système.
– *L'hétérogénéité :* Les éléments du contexte proviennent de sources variées, ce qui amène à une large hétérogénéité en termes de modélisation, de traitement et de qualité. La complexité et l'hétérogénéité des informations du contexte nécessitent des moyens de modélisation et d'interprétation spécifiques. Par exemple, en informatique mobile, les dispositifs tels que les ordinateurs portables, les Smartphones, les tablettes, etc. sont différents à tous niveaux (matériel, logiciel, communication, etc.). Le développement d'applications devra proposer des solutions d'interopérabilité.
– *La variabilité :* Dans un contexte dynamique et continuellement en évolution, le fonctionnement des systèmes mobiles peut être perturbé par les modifications du contexte. Il faut alors pouvoir prendre en compte ces évolutions du contexte pour pouvoir agir de manière à maintenir la qualité du service rendu et satisfaire l'utilisateur.
– *La continuité :* Le contexte évolue pendant que l'utilisateur se déplace d'une localisation à une autre ou pendant qu'il obtient de nouvelles ressources, etc. Cette continuité dans l'évolution du contexte doit s'accompagner d'un fonctionnement continu des systèmes malgré les changements perpétuels du contexte.

Pour conclure cette section, nous remarquons que la notion de contexte n'est pas nouvelle en informatique et que malgré la multitude d'études réalisées sur le sujet, il n'existe pas de définition consensuelle. Toutefois l'analyse des différentes définitions proposées fait ressortir les trois principes suivants : (1) Le contexte émerge ou se définit pour une

finalité, une utilité précise (Brézillon, 2002). Pour notre part, la finalité recherchée est l'adaptation dynamique des contenus d'apprentissage dans un système m-learning. (2) Le contexte est un ensemble d'informations structuré, il est partagé, il évolue et sert l'interprétation (Winograd, 2001) et dépend d'une finalité. Il faut donc, premièrement, définir la finalité et ensuite déterminer les informations nécessaires pour servir cette finalité. (3) Le contexte est un espace d'informations évolutif et hétérogène.

Nous présentons dans la section suivante les éléments et les dimensions potentielles du contexte qui seront utilisés dans notre système.

2.1.2/ DIMENSIONS DU CONTEXTE

À des fins d'adaptation, les informations contextuelles doivent être collectées et présentées au système. Vu l'hétérogénéité, la diversité et la qualité variable de ces informations, il est préférable de faire une catégorisation pour faciliter l'opération d'adaptation. Les différentes catégories qui constituent le contexte sont dépendantes des objectifs du système. Nous présentons ici la plupart des dimensions couramment utilisées dans différents modèles de contexte et les dimensions nécessaires pour la conception de notre système.

(Schilit et al., 1994) proposent de catégoriser le contexte en deux dimensions : le contexte primaire qui contient les informations sur la localisation, l'identité, le temps et l'activité de l'utilisateur ; et le contexte secondaire qui contient des informations pouvant être déduites du contexte primaire (par exemple, de la localisation, on peut déduire les personnes à proximité). (Chen and Kotz, 2000) proposent : le contexte actif qui influence le comportement d'une application ; et le contexte passif qui est nécessaire mais pas critique pour l'application. (Petrelli et al., 2000), se basent sur les travaux de Schilit et Chen, et introduisent deux nouvelles dimensions du contexte : le contexte matériel (device, plateforme existante) ; et le contexte social (être seul ou pas, qui sont les autres, les relations entre ces individus, etc.). Enfin (Dey, 2000) propose de classer toute variable contextuelle dans des dimensions répondant chacune aux questions, *quand ?*, *où ?*, *quoi ?* et *qui ?*.

Dans le domaine de l'apprentissage, temps, identité, activité et localisation sont des catégorisations primaires et nécessaires du contexte pour expliciter une situation. Dans les travaux de (Derntl and Hummel, 2005), le contexte est composé de cinq dimensions : la dimension physique ; la dimension numérique (ressources numériques) ; la dimension dispositifs (logiciels, matériels, réseau, dispositifs portables) ; la dimension apprenant ; et le contexte spécifique du domaine d'application.
Par rapport à Derntl, (Azouaou and Desmoulins, 2006) introduisent une nouvelle dimension pour décrire le profil de l'environnement physique et le temps.
(Pham, 2010) synthétise les propositions précédentes de catégorisation et propose les dimensions suivantes : la dimension temporelle ; la dimension spatiale ; la dimension dispositifs (caractérise les dispositifs d'apprentissage mobiles ou pas, les ressources

informatiques et les outils pédagogiques) ; la dimension environnement (décrit les caractéristiques de l'environnement autour de l'utilisateur) ; la dimension utilisateur (décrit les participant humains du contexte dans le système et les relations entre eux) ; la dimension du scénario/activités (détermine l'intention de l'apprenant, les informations, les connaissances, etc. nécessaires à l'accomplissement d'une activité).

Dans ce qui suit, nous décrivons en détail les dimensions du contexte les plus utilisées dans le domaine de l'apprentissage mobile.

2.1.2.1/ DIMENSION SPATIALE

Cette dimension définit selon (Soylu et al., 2009) la localisation. La localisation de l'apprenant mobile est un facteur d'une grande importance afin de proposer des contenus adaptés aux propriétés de l'environnement dans lequel il se situe.

Souvent, la localisation est considérée comme un concept seulement caractérisé par les coordonnées physiques par rapport à un système de coordonnées géographiques. Cependant, les coordonnées physiques ne sont pas les seules données pertinentes pour définir une localisation. Supposons que nous essayons de nous localiser sans l'utilisation de technologies de détection de position, tout simplement en essayant de répondre à la question *où suis-je ?*. Nous pouvons imaginer une multitude de réponses à cette simple question. En effet, tel que discuté dans (Dobson, 2005), il existe différentes façons plausibles pour caractériser la localisation d'un utilisateur mobile : relative (à côté de, loin de, etc.), nom de la localisation (place de la République, dans un restaurant, dans un train, etc.), type de la localisation (dynamique, fixe, public, privé, etc.), propriétés de la localisation (niveau de bruit, niveau de luminosité, confort, etc.).

L'utilisation d'un modèle sémantique, par exemple un modèle ontologique, permettrait de décrire les localisations et leurs relations, afin d'accroître la compréhension sémantique de l'environnement. Comme dans (Chen et al., 2004b) ou dans le projet SPIRIT [4] (Spatially-Aware Information Retrieval on the Internet), nous proposons de nous baser sur un modèle sémantique décrivant les localisations et leurs relations.

2.1.2.2/ DIMENSION TEMPORELLE

Cette dimension caractérise, selon (Soylu et al., 2009), des éléments de contexte dont l'existence ou l'importance dans la perception du contexte est liée à un moment donné. Nous considérons que le contexte temporel a une influence majeure sur l'activité de l'apprenant en mobilité. Par exemple, imaginons qu'un apprenant émet la requête, *apprendre dans le métro avec un trajet d'une durée de 15 min*. Le système dans ce cas doit suggérer des vidéos dont la durée est légèrement inférieure à 15 min. Afin de définir les aspects

4. http ://www.geo-spirit.org/index.html

temporels liés à la situation de l'apprenant (lundi, weekend, matin, soir, etc.), mais aussi liés aux documents (heures et dates de disponibilité), un modèle pour la représentation du temps est nécessaire.

Les formes les plus courantes pour décrire la dimension temporelle sont : l'heure (le fuseau horaire, l'heure actuelle d'apprentissage, etc.), la date (date d'une activité, jour de la semaine, mois de l'année, etc.), la période (le début et la fin d'une activité, etc.), la durée d'une activité ou la durée d'une ressource (15min, 1h, etc.), et les relations temporelles (après, avant, pendant, etc.).

La représentation du temps par une ontologie accroît la compréhension commune par les systèmes de cette dimension et permet des raisonnements temporels. L'ontologie OWL-Time (Pan, 2007) est aujourd'hui une référence pour la représentation et le raisonnement sur le temps. Nous proposons de baser notre travail sur cette ontologie et de l'étendre avec quelques classes spéciales du temps, comme le temps du jour (matin, après midi, soir, etc.), le jour de la semaine, les saisons, etc.

2.1.2.3/ DIMENSION UTILISATEUR

Il est nécessaire d'identifier les utilisateurs du système avant de commencer à collecter des informations pour construire les différents profils (Brusilovsky et al., 2007). Dans notre système m-learning par exemple, plusieurs profils d'utilisateurs interagissent : les apprenants, les formateurs, les tuteurs, etc.
Une gestion des profils joue un rôle important dans un système d'apprentissage pour adapter les ressources aux spécificités de chaque utilisateur. La gestion et la diffusion de ressources personnalisées nécessite de pouvoir modéliser, de gérer l'évolution et de mettre à jour le profil utilisateur. Un profil utilisateur est une collection de données qui caractérisent un utilisateur dans un système. Ces données sont collectées par le système et classées essentiellement en deux ensembles : un ensemble de données statiques qui regroupe des informations générales sur l'utilisateur, telles que le nom, le prénom, la date de naissance, la langue maternelle, etc. ; et un ensemble de données dynamiques qui varie en fonction de différents domaines d'application, telles que les objectifs, les préférences, les connaissances, les compétences, les centres d'intérêts, etc.

La modélisation du profil repose sur des techniques permettant non seulement de représenter et construire le profil de l'utilisateur, mais aussi de gérer son évolution de manière dynamique au cours du temps. Une représentation sémantique du profil utilisateur, généralement basée sur une ontologie, permet de mieux représenter les différents acteurs du système en décrivant les relations sémantiques entre ces derniers, mais aussi avec les données collectées sur eux (Gauch et al., 2002) (Heckmann et al., 2007).

2.1.2.4/ DIMENSION DISPOSITIF

Cette dimension permet de mesurer les caractéristiques des dispositifs (de Groot and van Welie, 2002). Afin d'adapter un contenu d'apprentissage à la technologie mobile utilisé par l'apprenant il est nécessaire de connaitre les propriétés caractérisant cette technologie. Par exemple, si une formation nécessite l'utilisation d'une souris, un Smartphone n'est alors pas adapté, de même si une formation contient des vidéos elle ne sera pas adaptée à des appareils mobiles sans écran tels que les lecteurs MP3.

Des travaux sur la construction de modèles sémantiques pour la description des dispositifs sont proposés, comme la FIPA Device Ontology Specification [5].

Toute adaptation au contexte nécessite d'abord l'acquisition des informations de contexte qui seront classées, traitées et analysées afin de réaliser l'adaptation. Nous présentons dans la section suivante les différentes techniques utilisées pour les acquérir.

2.1.3/ ACQUISITION DES INFORMATIONS DE CONTEXTE

Diverses méthodes sont utilisées pour collecter les informations de contexte. Dans les travaux de (Mostéfaoui et al., 2004) trois types de contextes sont identifiés du point de vue de leur acquisition :

Le contexte capturé :
Ce type de contexte est acquis par le moyen de capteurs matériels ou logiciels tels que les capteurs de localisation, de température, de niveau d'éclairage, etc. La localisation de l'utilisateur est généralement capturée à l'aide d'un système de suivi de positionnement GPS (Global Positioning System), grâce aux réseaux sans fil, ou encore à l'aide de méthodes de détection de proximité (systèmes RDIF (Radio Frequency IDentification), Blutooth, Ultrasons, etc.).

Le contexte explicite :
Ce type de contexte contient des données collectées via l'action de l'utilisateur, par exemple en remplissant un formulaire d'inscription. Ainsi l'utilisateur communique explicitement au système son nom, son prénom, sa date de naissance, ses préférences, ses connaissances, etc.

Le contexte implicite :
Ce type de contexte contient des données collectées de manière non-intrusive par le système. Contrairement au contexte capturé et au contexte explicite, qui représentent des informations de contexte de bas niveau, le contexte implicite représente un contexte

5. http ://www.fipa.org/specs/fipa00091/PC00091A.html

de haut niveau. En d'autres termes, le contexte implicite est calculé à partir d'autres informations de contexte. Par exemple, le pays où se trouve l'utilisateur est un contexte de haut niveau déduit à partir du contexte de bas niveau, à savoir ses coordonnées collectées à partir d'un GPS.

Ces trois types de données sont par la suite stockés et classés selon les différentes dimensions du contexte décrites précédemment.
L'hétérogénéité et la dynamicité des informations contextuelles nécessitent des modèles abstraits de description de contexte. Plusieurs travaux ont défini différentes méthodes pour modéliser le contexte, nous les décrivons dans la section suivante.

2.1.4/ MODÉLISATION DU CONTEXTE

Un modèle de contexte est une représentation des informations contextuelles dans le système. Ces informations contextuelles proviennent de plusieurs sources de nature diverses et hétérogènes, telles que les données provenant des capteurs, des entrées utilisateurs ou des données systèmes. Le contexte doit être construit de manière structurée et ordonnée pour qu'il puisse être exploité de façon efficace et afin que le système puisse raisonner de façon pertinente. De nombreuses recherches proposent différents modèles de contexte fondés sur différents langages. Un aperçu de ces modèles de contexte est présenté ci-dessous (Behlouli, 2006) (Chaari, 2007) (Makssoud, 2008) (Achiri, 2012) :

Modélisation par paire attribut/valeur :
Un des premiers travaux sur la modélisation du contexte est publié par Schilit (Schilit et al., 1993). Les informations du contexte sont modélisées par une paire attribut/valeur. L'attribut représente le nom d'une information contextuelle et la valeur représente la valeur actuelle de cette information (Makssoud, 2008). Par exemple, on peut décrire un contexte *context1* comme suit : *(Name=context1, User=x, Localisation=y, Time=t)*. L'avantage de cette méthode est la facilité d'implantation. En effet, la gestion du contexte revient à parcourir la liste des contextes disponibles.
Cependant, ce modèle manque d'expressivité et de complétude (Chaari, 2007). Par exemple, si nous définissons un nouveau contexte *context2* : *(Name=context2, User=x, Localisation=y', Time=t)*, nous ne pouvons pas dire que *context2* est un sous-contexte de *context1*, par conséquent les propriétés offertes par l'application dans *context1* ne sont pas appliqués dans le *context2* .

Modélisation par mots clés :
Dans ce modèle il s'agit d'associer à un utilisateur un ensemble de mots clés pondérés. Ces mots clés sont extraits du domaine de l'application. Les poids associés à chaque mot clé sont une représentation numérique de l'intérêt de l'utilisateur par rapport au mot concerné. Par exemple on peut décrire les préférences d'un utilisateur inscrit à une

plateforme de recommandation de films par un ensemble de mots clés comme suit : (action=10, romantique= 8, drame=3, aventure=10). Ces mots clés représentent des préférences ou des centres d'intérêt qui constituent la dimension profil de l'utilisateur. Cette représentation du profil utilisateur est parmi les premières techniques développées et elle a été utilisée pour les systèmes de recherche d'informations (Lieberman, 1995) ou encore pour les systèmes de recommandation (Balabanovic and Shoham, 1997).
Seulement cette technique de modélisation présente plusieurs inconvénients. Premièrement, elle ne permet pas de représenter des types d'informations contextuelles autres que les préférences et les intérêts. De plus, elle ne permet pas de capturer les relations et les dépendances entre les informations du contexte.

Modélisation par langages à balises :
Ici, le contexte est modélisé par une structure de données hiérarchique constituée de balises avec des attributs et des contenus. Des langages dérivés du SGML (Standard Generic Markup Language) et en particulier le XML (eXtended Markup Language) sont utilisés pour cette modélisation. Par exemple, ConteXtML (Contexte Markup Language) est un protocole basé sur XML, utilisé pour échanger des informations du contexte entre un utilisateur mobile et un serveur. Les messages ConteXtML sont regroupés dans des balises *context*, et entre ces balises un utilisateur peut par exemple indiquer ses coordonnées géographiques par une balise *spatial* (Ryan, 1999). Dans les travaux de Klyne, le standard du W3C pour la description des systèmes mobiles, CC/PP (Composite Capabilities/ Preferences Profiles) est utilisé. Ce langage est basé sur RDF (Resource Description Framework). Il est utilisé afin de décrire les capacités des dispositifs et les préférences de l'utilisateur (Klyne et al., 2005).
L'avantage du langage de balises est qu'il permet la mise en forme et l'échange de modèles entre plusieurs plateformes différentes facilement. Cependant, les auteurs des travaux décrits précédemment soulignent aussi que cette méthode de modélisation reste non intuitive et difficile à utiliser pour décrire des informations plus complexes.

Modélisation par graphes :
Cette méthode consiste à modéliser les informations contextuelles selon un graphe conceptuel. Le modèle graphique le plus répandu est l'UML (Unified Modeling Language). Les diagrammes de classes d'UML sont utilisés pour modéliser de façon statique les concepts qui constituent le contexte sous forme classe/association. Les classes sont utilisées pour représenter des éléments du contexte avec leurs attributs et les associations représentent les liens entre ces éléments. Sheng propose un modèle basé sur une extension d'UML appelé ContextUML (Sheng and Benatallah, 2005). Ce modèle est composé de plusieurs classes qui permettent de créer des services sensibles au contexte. La classe *Context* permet de décrire un contexte de bas niveau représenté par la classe *AtomicContext*, ou bien un contexte implicite représenté par la classe *CompositeContext*. Pour chaque contexte de bas niveau, le modèle permet de spécifier la source à partir de

laquelle il a été collecté. Henricksen propose un langage graphique appelé CML (Context Modeling Language)(Henricksen and Indulska, 2004). Ce langage permet de capturer les dépendances entre les types de données du contexte et de qualifier les informations du contexte (capturées, implicites, explicites). Il permet en plus de sauvegarder l'historique de certaines données contextuelles et de définir les contraintes sur les données historisées.
Malgré la simplicité de ces modèles au niveau de la représentation, ils restent moins formels que les autres méthodes (exemple, CML n'est pas standardisé, et donc il n'existe pas d'outil support).

Modélisation orientée objet :
L'approche orientée objet est utilisable pour pouvoir intégrer facilement la représentation du contexte dans des systèmes complexes. Le but de la modélisation orientée objet est de profiter des mécanismes d'encapsulation, d'héritage et de réutilisation. Parmi les travaux qui ont abordé cette méthode, (Hofer et al., 2003) introduit une approche nommée HYDROGEN, où chaque type de contexte utilisé est composé de plusieurs objets contexte. Chaque type de contexte constitue une super-classe de plusieurs éléments du contexte, notamment le temps, le réseau, la localisation, l'utilisateur, la machine, etc. Les autres éléments peuvent être ajoutés par héritage de la super-classe.
Cette approche de modélisation est efficace en termes de distribution et d'abstraction. Cependant, elle reste propre à une application spécifique et ne permet pas de partager le contexte entre applications.

Modélisation basée sur la logique :
La première approche de modélisation du contexte en utilisant la logique a été publiée par (McCarthy and Buvac, 1994). Le contexte est défini par un ensemble de formules du type *c' : ist(p,c)* qui signifie que l'assertion *p* est vraie dans le contexte *c*, confirmant ainsi le contexte *c'*. Par exemple, *c0 : ist(context-of("Histoire de la Bourgogne"),"Philippe III le Bon est un Duc de Bourgogne")* considère que Philippe III le Bon est un Duc dans l'histoire de la Bourgogne. Ce type de modèle est caractérisé par un haut degré de formalisation et il est utilisé avec des règles, formant ainsi des mécanismes de déduction de nouvelles informations sur le contexte et de détection d'inconsistances dans le modèle.
Cependant, l'inférence dans un tel modèle est une opération qui nécessite un grand nombre de ressources systèmes. Or l'informatique mobile considère généralement des environnements dans lesquels les terminaux mobiles sont limités en ressources. Cette modélisation n'est donc pas adaptée pour notre problème.

Modélisation basée sur les ontologies :
Plusieurs modèles d'ontologies ont été proposés pour représenter le contexte. Parmi eux, les travaux de (Chen et al., 2004a) proposent un système nommé CoBrA (Context Bro-

ker Architecture) basée sur l'idée d'un courtier de contexte. Dans le système CoBrA une collection d'ontologies appelée COBRA-ONT est définit pour la modélisation du contexte dans un environnement d'une salle de rencontre intelligente. Cette collection d'ontologies aide le courtier de contexte à partager les informations contextuelles avec d'autres agents du système. Dans un autre travail de (Wang et al., 2004), une ontologie du contexte appelée CONON (CONtext ONtology) est proposée pour la modélisation du contexte dans un environnement ubiquitaire et pour le support de raisonnement logiques. CONON propose deux ontologies : La première ontologie capte les concepts généraux du contexte de base d'une manière abstraite (localisation, utilisateur, activité, etc.). La deuxième ontologie est constituée d'un ensemble de sous-classes qui viennent compléter l'ontologie de base afin de détailler la modélisation des concepts généraux avec un ensemble de termes concrets, suivant un environnement (maison, bureau, tablette, etc.).
L'avantage de la modélisation basée sur les ontologies réside dans les caractéristiques mêmes de l'ontologie qui offrent le moyen de faire des descriptions sémantiques et d'effectuer un raisonnement logique grâce aux moteurs d'inférence associés.

Selon (Strang and Linnhoff-Popien, 2004) la modélisation du contexte doit répondre à six exigences. Dans son travail, (Yin, 2010) décrit ces différents exigences et propose une évaluation des différentes approches de modélisation par rapport à ces exigences :

- *Composition distribuée (dc) :* Les systèmes ubiquitaires sont des systèmes distribués, qui n'ont pas une instance centrale responsable de la création, du déploiement et de la maintenance des données, et en particulier de la description du contexte. Surtout que l'administration d'un modèle de contexte et ses données varient selon la topologie du réseau, de la source, etc.
- *Validation partielle (pv) :* La validation des informations du contexte en structure ou en niveau des instances est nécessaire, quand un modèle de contexte est en cours d'utilisation, à cause de la complexité des interactions contextuelles.
- *Richesse et qualité de l'information (qua) :* La qualité de l'information contextuelle capturée grâce aux senseurs varie en fonction du temps. De même, la richesse de l'information contextuelle captée grâce aux différents types de senseurs peut différer dans un environnement. Un modèle de contexte doit donc supporter ces deux critères.
- *Incomplétude et ambiguïté (inc) :* Les informations contextuelles sont parfois incomplètes ou ambiguës, en particulier si ces informations sont recueillies à partir des capteurs en réseau. Un modèle de contexte doit couvrir ce critère, par exemple par l'interpolation des informations incomplètes au niveau des instances.
- *Degré de formalité (for) :* Afin de décrire des faits contextuels et leurs interrelations d'une manière précise (après, avant, à côté, près de, etc.), il est souhaitable d'utiliser un modèle formel, capable d'être interprété de la même manière par différents systèmes.
- *Applicabilité aux environnements existants (app) :* Il est important que le modèle de contexte puisse être applicable et adaptable dans les infrastructures déjà existantes.

Une évaluation des approches de modélisation du contexte par rapport à ces différentes exigences est proposée dans la table 2.1.

Critère	dc	pv	qua	inc	for	app
Modélisation par attribut/valeur	-	-	-	-	-	+
Modélisation par mots clés	-	-	-	-	-	++
Modélisation par langages de balises	+	++	-	-	+	++
Modélisation par graphes	-	-	+	-	+	+
Modélisation orientée objet	++	+	+	+	+	+
Modélisation basée sur la logique	++	-	-	-	++	-
Modélisation basée sur les ontologies	++	++	+	+	++	+

TABLE 2.1 – Évaluation des approches de modélisation du contexte par rapport aux critères de (Strang and Linnhoff-Popien, 2004), (Yin, 2010)

La modélisation par attribut/valeur et par mots clés ne répondent qu'à un seul des six critères. La modélisation par langage de balises, par graphes ou basée sur la logique, répondent à plus de deux critères, mais ne satisfont pas les six critères. Les deux modèles orienté objet et basé sur les ontologies répondent à tous les critères, seulement le modèle ontologique permet une meilleure validation partielle des données et une meilleure formalité du modèle. D'après l'analyse des différentes méthodes de modélisation du contexte, nous concluons que l'approche par ontologies correspond le mieux à l'implémentation de notre système sensible au contexte.

Nous présentons la notion de sensibilité au contexte dans la section suivante.

2.2/ SYSTÈMES SENSIBLES AU CONTEXTE

Le terme sensibilité au contexte est devenu aujourd'hui une technologie embarquée dans une grande variété de systèmes informatiques. Dans cette partie du chapitre, nous commençons par définir la notion de sensibilité. Ensuite nous présentons l'architecture générale d'un système sensible au contexte et quelques architectures connues dans le domaine.

2.2.1/ DÉFINITION DE LA SENSIBILITÉ AU CONTEXTE

La notion de sensibilité au contexte a été tout d'abord introduite dans le domaine de l'IHM (Interactions Homme-Machine) par Weiser en 1991 pour réconcilier le monde virtuel et le monde physique (Weiser, 1991). Il déclare que les technologies les plus réussies sont celles qui s'associent à la vie de tous les jours, jusqu'à ce qu'il devienne difficile de distinguer les deux. L'informatique ubiquitaire consiste en l'intégration du monde informatique

partout dans l'environnement et de manière invisible.

La première définition de la sensibilité au contexte a été proposée par (Schilit et al., 1994), comme étant la capacité d'un système à s'adapter au contexte de son exécution en fonction de sa localisation, de l'ensemble des personnes à proximité, des équipements accessibles, etc. (Brown, 1996) annonce qu'une application sensible au contexte doit automatiquement extraire de l'information ou effectuer des actions en fonction du contexte utilisateur détecté par les capteurs. (Pascoe, 1998) définit la sensibilité au contexte comme étant la capacité des dispositifs informatiques à détecter, interpréter et répondre aux aspects de l'environnement local d'un utilisateur et des dispositifs informatiques eux-mêmes. (Dey, 2000) affine ces définitions en une autre plus générale. Il annonce qu'un système est sensible au contexte lorsqu'il utilise le contexte pour fournir des informations pertinentes aux utilisateurs : «*a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*». Ces définitions s'articulent toutes autour de l'aptitude d'un dispositif à adapter son comportement en se basant sur le contexte de l'utilisateur. Un système sensible au contexte est un ensemble de mécanismes destinés pour la collection et la gestion des informations de contexte, et le contrôle du comportement du système en fonction de ces informations.

2.2.2/ ARCHITECTURE GÉNÉRALE D'UN SYSTÈME SENSIBLE AU CONTEXTE

Pour réaliser un système sensible au contexte dans une activité d'apprentissage, nous avons étudié les différents cycles de vie des systèmes sensibles au contexte.

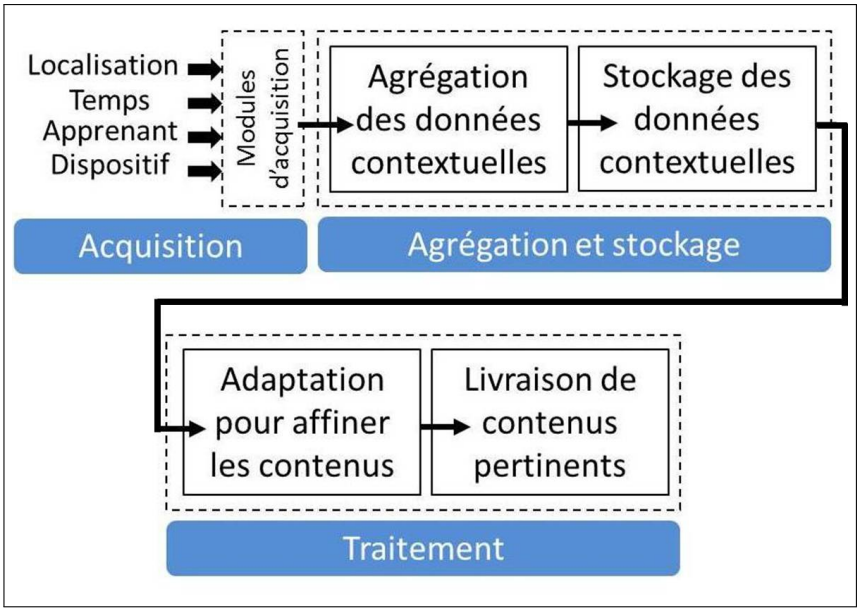

FIGURE 2.1 – Architecture minimale d'un système sensible au contexte

Dans la littérature, plusieurs travaux ont proposé différentes architectures de ces systèmes, comme dans (Dey et al., 2001), (Schilit et al., 2002) et (Chaari, 2007). A par-

tir de ces travaux on peut extraire une architecture minimale applicable pour tous les systèmes sensibles au contexte. Cette architecture s'articule essentiellement autour de trois modèles pour : l'acquisition, l'agrégation et le stockage, et le traitement des informations contextuelles (Ghayam, 2011). La figure 2.1 illustre l'architecture minimale d'un système sensible au contexte.

Modèle d'acquisition du contexte :
Dans ce modèle il s'agit de capturer toutes les informations qui sont disponibles dans un contexte utilisateur, à partir de divers modules d'acquisition (capteurs, utilisateurs, services tiers, etc.). Ces modules d'acquisition peuvent être physiques ou logiques. Les modules physiques sont des dispositifs matériels qui sont capables de fournir des données de contexte, tels que les GPS, pour déterminer les coordonnées d'un utilisateur. Les modules logiques fournissent des informations contextuelles à partir d'applications ou services logiciels. Par exemple, certaines nouvelles applications mobiles permettent de déterminer la position ou l'activité courante de l'utilisateur.

Modèle d'agrégation et stockage du contexte :
Dans ce modèle il s'agit d'interpréter les informations contextuelles fournies par les modules d'acquisition. Cette étape sert à analyser et à transformer les données brutes fournies par l'étape d'acquisition dans d'autres formats plus faciles à manipuler et à utiliser par le système. Par exemple, les coordonnées géographiques peuvent être transformées en adresses physiques (numéro de rue, code postal, nom de ville, etc.) afin qu'elles soient plus compréhensibles par l'utilisateur. Une fois les informations contextuelles capturées et interprétées, elles doivent être bien gérées pour faciliter leur utilisation. La gestion du contexte contient l'organisation (en dimensions) et la représentation formelle des informations contextuelles selon un modèle. Le choix du modèle dépend fortement des mécanismes choisis pour adapter le comportement du système au contexte. Plusieurs méthodes de modélisation ont été proposées dans la littérature, et ont été présentées dans la section 2.1.4.

Modèle de traitement selon un contexte :
Dans ce modèle le contexte est exploité pour fournir finalement une application sensible au contexte. C'est au niveau de cette étape que s'effectue l'implémentation d'un ensemble de mécanismes d'adaptation prévues suite aux changements de contexte. Ces mécanismes peuvent être, par exemple, un ensemble de règles d'adaptation, implémentées en utilisant la logique de prédicats, ou des algorithmes d'adaptation.

Nous présentons dans la section suivante l'étude de quelques architectures dans le domaine de la sensibilité au contexte.

2.2.3/ EXEMPLES D'ARCHITECTURES DE SYSTÈMES SENSIBLES AU CONTEXTE

Dans cette section, nous présentons des architectures de référence, de plateformes sensibles au contexte. Nous analysons les approches abordées par chaque plateforme pour implémenter l'architecture générale, décrite précédemment, d'un système sensible au contexte. Nous nous intéressons spécifiquement aux plateformes suivantes décrites et analysées dans l'état de l'art des travaux de (Chaari, 2007) et de (Pham, 2010) :

Architecture Context Toolkit :
Context Toolkit (Dey et al., 2001) offre une boite à outils et des composants logiciels permettant le développement et le déploiement d'applications sensibles au contexte et réutilisables pour d'autres applications. Il a une architecture en couche permettant la séparation des processus d'acquisition et de représentation de contexte, de l'adaptation. La figure 2.2 présente l'architecture de base de Context Toolkit.

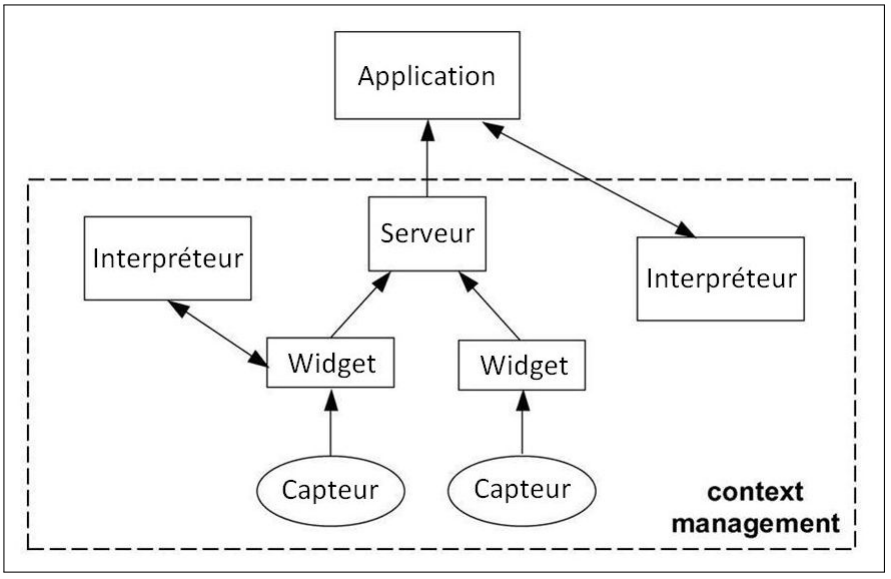

FIGURE 2.2 – Architecture de Context Toolkit

Les *capteurs* détectent le contexte physique. Les *widgets* sont des composants autonomes dont le rôle est de communiquer les informations capturées aux serveurs ou aux interpréteurs, en fonction des données reçues par les capteurs. Ils ont aussi comme rôle de constituer un historique des informations du contexte. Les *interpréteurs* transforment l'information du contexte dans le but d'élever le niveau d'abstraction, en leur donnant une sémantique. Finalement, les *serveurs* ont pour tâche de collecter l'ensemble des signaux émis par les autres composants et de faire le lien entre les applications et les widgets.
Pour résumer, dans Context Toolkit nous retrouvons les différentes étapes de l'architecture générique des systèmes sensibles au contexte. Les principaux inconvénients de l'architecture Context Toolkit sont :
– Les widgets ne sont que de simples pilotes logiciels pour les capteurs.

– Les widgets ne fournissent pas de sens aux informations qu'ils transmettent.
– Les serveurs regroupent la majeure partie du travail.

CMF :
CMF (Context Management Framework) (Korpipaa et al., 2003) est une plateforme analogue à context toolkit de Dey. Cette plateforme est composée de cinq principales entités, comme le montre la figure 2.3.

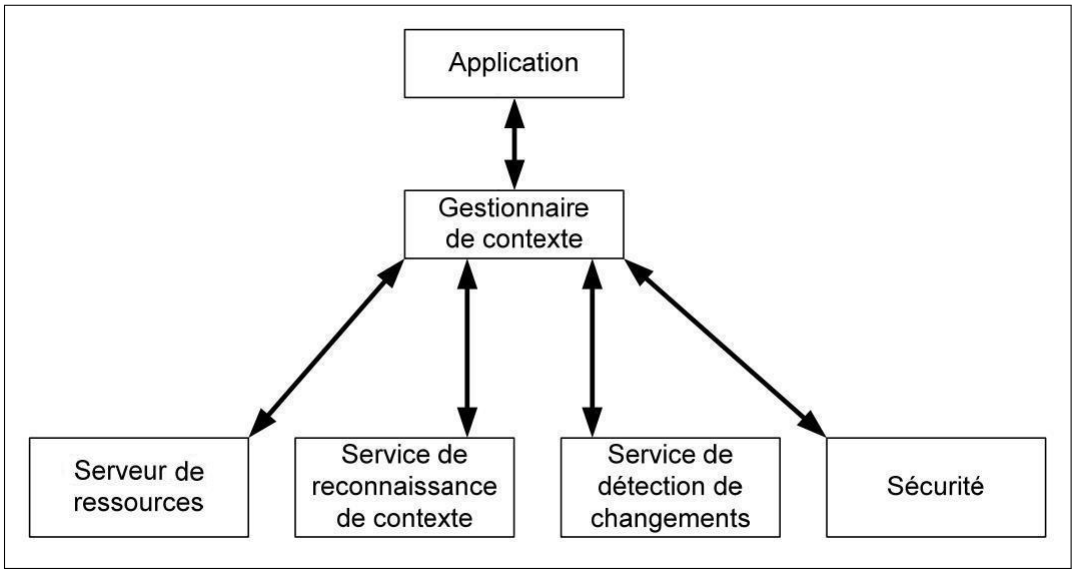

FIGURE 2.3 – Architecture de CMF

CMF est composé de (1) un *serveur de ressources* pour la capture du contexte et leur interprétation selon un schéma spécifique avant de les envoyer au gestionnaire de contexte, (2) un *service de reconnaissance de contexte* pour l'interprétation du contexte et la représentation dans un modèle ontologique, (3) un *service de détection de changements* pour détecter les changements d'un service suite à un changement de contexte, (4) une composante *sécurité* pour vérifier et contrôler les informations de contexte, et finalement (5) un *gestionnaire de contexte* responsable du stockage des informations contextuelles dans un serveur et de la livraison aux clients selon différents mécanismes. CMF utilise les ontologies pour la représentation du contexte, mais n'offre pas un mécanisme de raisonnement. Il possède un bon mécanisme d'interprétation, ce qui favorise l'abstraction du contexte et la réutilisation. Le module de sécurité est un point en plus. Le principal inconvénient de l'architecture CMF est qu'il utilise un serveur pour une gestion centralisé du contexte, ce qui présente un problème en cas de défaillance du serveur.

CoBrA :
Context broker ou CoBrA à été développé par (Chen et al., 2004a). Son architecture est présenté dans la figure 2.4.

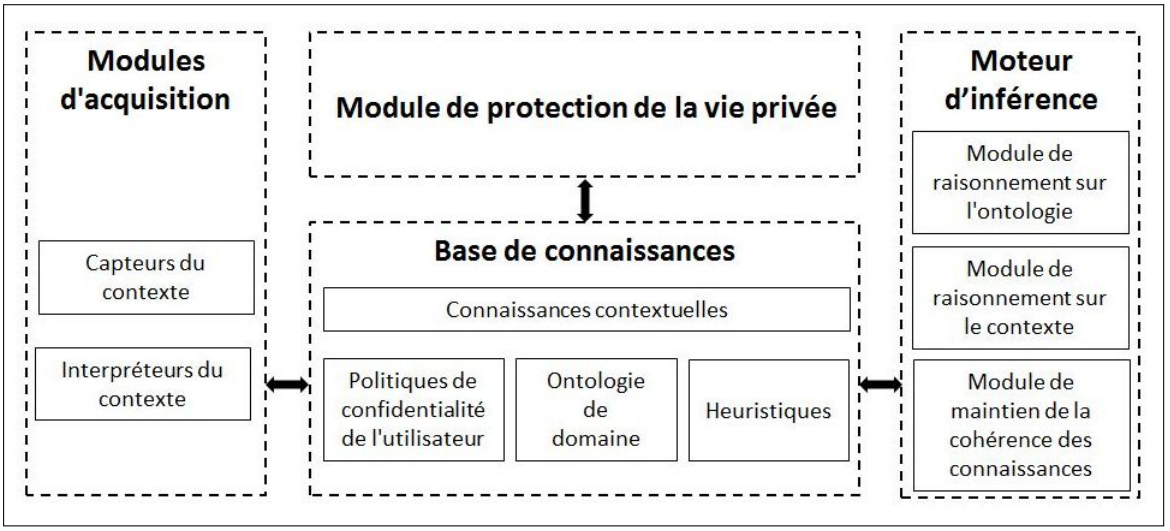

FIGURE 2.4 – Architecture de CoBrA

Dans la vision de Chen, l'environnement intelligent est muni de plusieurs *context brokers*, chacun étant responsable d'une partie de l'environnement (exemple, une salle de réunions intelligente particulière). Le rôle d'un context broker est de maintenir un modèle du contexte dont il est responsable et de partager ce modèle avec d'autres agents ou services [6]. Le context broker est un agent central qui reçoit des informations de différentes sources (des capteurs, d'autres agents, des dispositifs, des serveurs d'information, etc.) et doit fusionner ces informations afin de fournir un modèle toujours cohérent.
Les *modules d'acquisition* des informations de contexte ont le double rôle de collecter les données brutes et d'effectuer les premières transformations avant la transmission vers la base des connaissances. Ainsi, les informations de contexte transmises à la base de connaissances sont pré-traitées et dirigées vers le domaine correspondant à la catégorie du capteur source.
La *base de connaissances* utilise des ontologies pour décrire les informations contextuelles et partager ses connaissances. Cette base est couplée avec un *moteur d'inférence* composé de trois parties : (1) le *module de raisonnement sur l'ontologie* qui déduit des faits à partir des informations de la base de connaissances et des modèles définis dans les ontologies, (2) le *module de raisonnement sur le contexte* qui est un moteur d'inférence hybride basé sur la logique (déduction, logique flue, etc.) et l'analyse statistique (arbre de décision, réseau Bayésien, etc.) et (3) le *module de maintien de la cohérence des connaissances* qui sert à supprimer le bruit dans les informations de contexte en provenance de l'environnement physique et à réconcilier la base de connaissances avec les changements trop rapides de l'environnement.
CoBrA possède en plus un *module de protection de la vie privée*. Les utilisateurs peuvent ainsi définir des règles dans un langage propre à CoBrA. L'inférence sur ces règles permet au context broker de déterminer s'il peut partager une information donnée ou pas.

6. http ://www-prima.imag.fr/prima/people/zaidenberg/These/node25.html

En conclusion, l'architecture CoBrA utilise les ontologies pour la description du contexte, ce qui permet un raisonnement des informations du contexte. Elle utilise un modèle centralisé pour le stockage et le traitement du contexte. Le modèle étant centralisé, l'architecture ne tolère pas la survenue de défaillances de l'hôte hébergeant la base de connaissances. Pour cela, CoBrA nécessite un serveur dédié.

SOCAM :
SOCAM (Service-Oriented Context-Aware Middleware) (Gu et al., 2004) est un projet qui vise à assurer le développement et le prototypage rapide de services sensibles au contexte dans des environnements intelligents. Ce projet propose un système distribué qui convertit les divers espaces physiques, d'où le contexte est capturé, en un espace sémantique, où le contexte peut être partagé entre des services sensibles au contexte. Les composantes de l'architecture SOCAM sont présentées dans la figure 2.5.

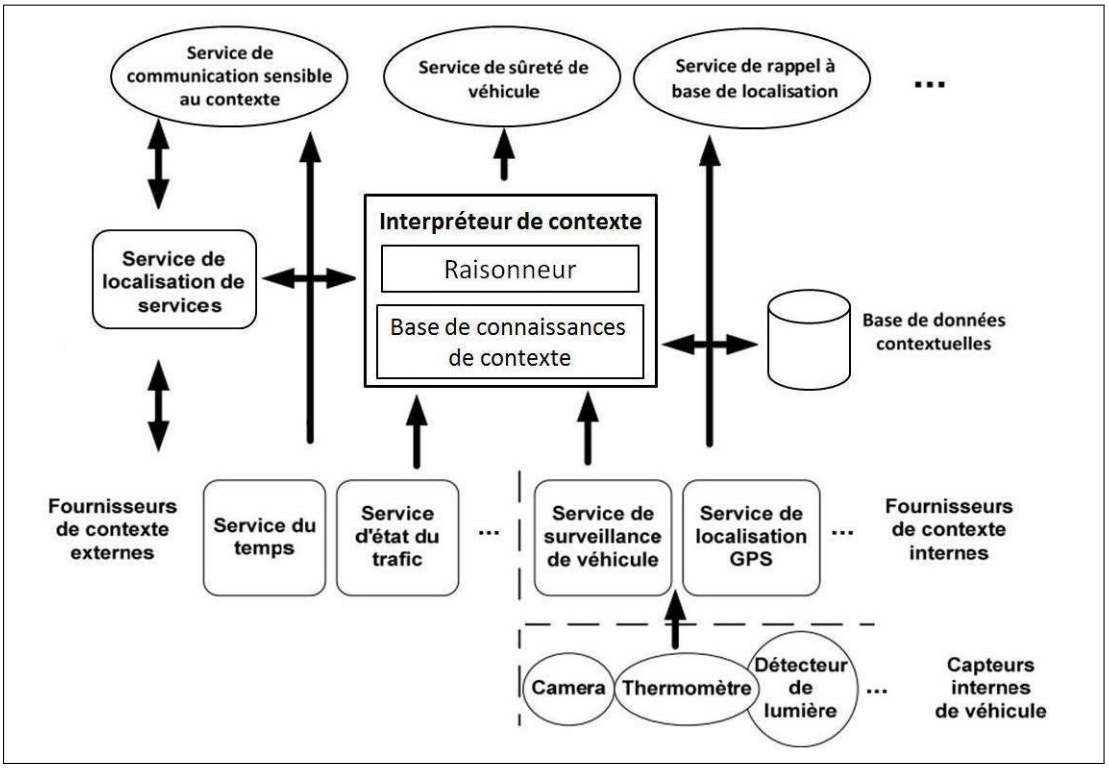

FIGURE 2.5 – Architecture de SOCAM

L'architecture SOCAM se base sur des ontologies pour la modélisation du contexte. Il utilise une première ontologie de haut niveau qui décrit les informations caractérisant l'environnement d'exécution de l'application d'une façon générale. Ensuite, une deuxième ontologie spécifique au domaine de l'application doit être définie (par exemple une ontologie décrivant un utilisateur dans une maison intelligente).
Les *fournisseurs de contexte* capturent des informations contextuelles à partir de sources hétérogènes, concernant l'environnement de l'application et l'environnement de l'utilisa-

teur. Ensuite, ils les convertissent et les stockent dans un modèle ontologique afin que le contexte puisse être partagé et réutilisé par d'autres composants de l'architecture. L'*interpréteur de contexte* fournit des services de raisonnement logique sur le modèle ontologique de contexte en appliquant des enchaînements de règles d'interprétation. La *base de données contextuelles* stocke les différents éléments de l'ontologie décrivant l'environnement de l'application et les instances des ontologies spécifiques décrivant l'environnement de l'utilisateur. Les *services sensibles au contexte* utilisent les différentes informations stockées dans la base de données de contexte pour modifier leur comportement selon le contexte courant. Finalement, le *service de localisation de services* fournit un mécanisme avec lequel des utilisateurs ou des applications peuvent localiser les fournisseurs et les interpréteurs de contexte.
L'utilisation des ontologies dans le modèle SOCAM offre une grande richesse sémantique et plus de possibilités d'inférence et d'interprétation du contexte. L'interpréteur du contexte dans ce modèle est chargé par une quantité importante de données sous forme de plusieurs ontologies de différents domaines. Ce choix affecte la performance globale du système vu les difficultés d'implémentation et de raisonnements sur plusieurs ontologies. De plus, le modèle étant centralisé, l'architecture ne peut pas tolérer de défaillances de l'hébergeur de la base de connaissances.

CASS :
CASS (Context-Awareness Sub-Structure) (Fahy and Clarke, 2004) est un middleware créé pour appuyer le développement d'applications sensibles au contexte. La figure 2.6 présente l'architecture de ce middleware.

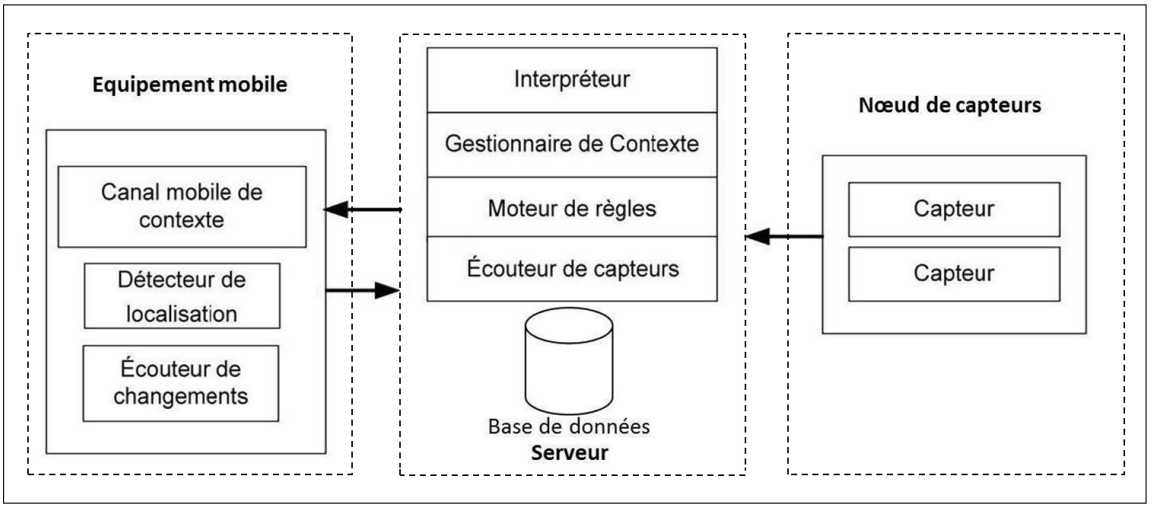

FIGURE 2.6 – Architecture de CASS

CASS offre une bonne abstraction des informations contextuelles et utilise un modèle orienté objet pour la représentation du contexte. Cette architecture est basée sur un

serveur contenant une *base de données* des informations contextuelles ainsi qu'une *base de connaissances*. La base de connaissance permet à un *moteur d'inférence* de déduire d'autres informations de contexte en utilisant la technique de chaînage arrière. Les équipements mobiles sont munis de différents types de *capteurs*. Ceux-ci perçoivent les changements du contexte et les envoient au serveur sans aucun traitement local. La communication entre les équipements mobiles et le serveur est assurée par des connexions sans fil.
Cette architecture offre une bonne modularité permettant à un utilisateur de modifier facilement les composants du serveur, en particulier le mécanisme d'inférence. Les équipements mobiles dans CASS ne font aucun traitement local, tout se fait au niveau du serveur. Cela limite leur autonomie, en revanche cela favorise l'extensibilité du système.

Architecture en couches :
(Daruwala, 2008) propose une architecture d'un système sensible au contexte en couches, présentée par la figure 2.7.

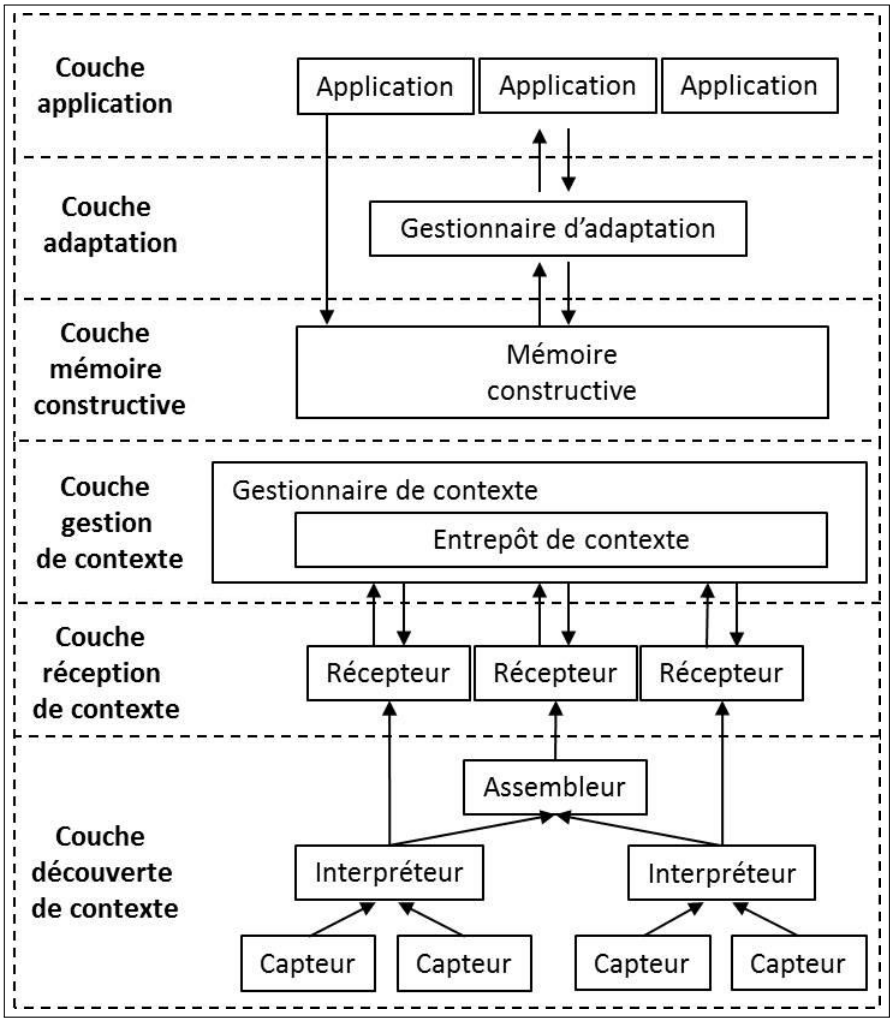

FIGURE 2.7 – Architecture en couches (Daruwala, 2008)

La *couche découverte de contexte* est responsable de l'acquisition d'informations contextuelles à partir des *capteurs* physiques et logiques. Les données capturées traversent une chaîne de composants comme l'*interpréteur*, l'*assembleur*, etc., avant d'arriver à la *couche réception de contexte*. La couche réception de contexte représente une interface entre la couche découverte de contexte et la couche gestion de contexte. Elle est conçue pour assurer la traduction entre les composants de la couche inférieure et les composants de la réception de contexte. La *couche gestion de contexte* consiste en un dépôt des informations contextuelles qui servent aux applications sensibles au contexte. Chaque application peut avoir son propre modèle de contexte. Un système de gestion est donc exigé pour maintenir des modèles différents. Le système de gestion stocke une base commune d'informations contextuelles, aussi bien que les métadonnées correspondant à chaque modèle de contexte, pour réduire les exigences de stockage et accélérer le processus requête-réponse. La *couche mémoire constructive* raisonne sur le contexte en effectuant des recherches dans la base de données du système pour obtenir un meilleur appariement des expériences passées au contexte en cours. Pour réaliser cette couche, un processus capable de stocker les représentations de contextes passées et leurs associations avec d'autres contextes est exigé. Ainsi, quand survient une nouvelle situation, le système peut non seulement rappeler un contexte existant dans le répertoire des contextes, mais aussi construire un nouveau contexte à partir des contextes existants similaires. Finalement, la *couche adaptation* permet d'adapter les services pour la *couche application* en fonction du contexte actuel.
Cette architecture en couches est actuellement considérée comme modèle pour les concepteurs de systèmes sensibles au contexte. Cependant, elle manque de solutions pour supporter la vie privée et la tolérance aux défaillances.

m-LOCO :
(Siadaty et al., 2008) présentent une architecture sensible au contexte pour le m-learning. L'architecture est basée sur une ontologie, appelée m-LOCO (mobile Learning Object Context Ontologies), dans le but de capturer des informations contextuelles dans des environnements d'apprentissage mobiles. L'objectif est de personnaliser la navigation pour guider l'apprenant en lui proposant des ressources selon ses préférences et ses besoins pour mieux réussir son apprentissage. (Siadaty et al., 2008) considèrent que les ontologies sont l'un des moyens les plus fonctionnels pour représenter des données contextuelles. En outre, les ontologies peuvent utiliser des règles pour raisonner sur le contexte, rendant la représentation à la fois flexible et extensible.
Les données contextuelles de m-LOCO sont intégrées dans un modèle flexible reposant sur une ontologie, plus précisément un ensemble d'ontologies inter-reliées nommé LOCO framework (Learning Object Context Ontologies) (Jovanovic et al., 2007). Ces ontologies représentent la couche fondatrice pour le développement de l'architecture m-LOCO. LOCO est composé de trois principales ontologies et est étendu grâce à un ensemble supplémentaire de propriétés à d'autres ontologies pertinentes, tels que la modélisation

de l'utilisateur (user model ontology) et la modélisation du domaine en question (domain ontology). Les trois principales ontologies sont : (1) l'ontologie des objets d'apprentissage (content structure ontology + content type ontology), (2) l'ontologie des activités d'apprentissage, inspirée du modèle IMS-LD [7] (learning design ontology) et (3) l'ontologie LOCO-Cite, qui est en fait le point de liaison les autres ontologies.
Pour être en mesure de traiter l'hétérogénéité des dispositifs dans un environnement ubiquitaire, une ontologie des dispositifs mobiles (mobile device ontology) vient compléter le modèle LOCO framework.

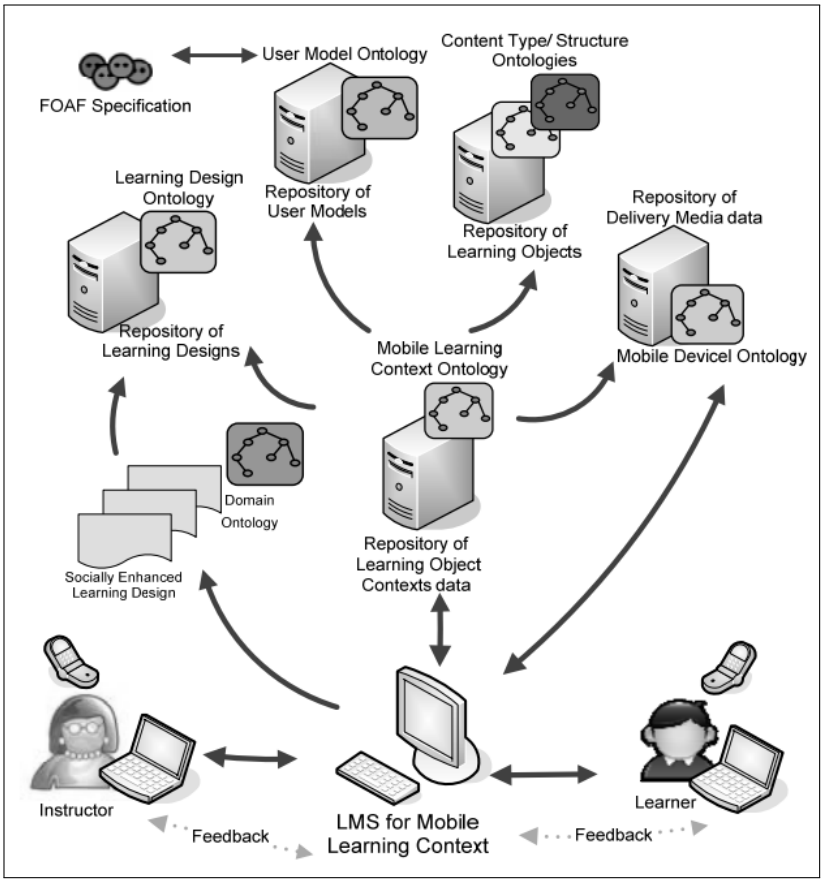

FIGURE 2.8 – Architecture générale de m-LOCO

La figure 2.8 illustre l'architecture globale de l'environnement d'apprentissage sensible au contexte m-LOCO comme un ensemble d'entrepôts ontologiques pour localiser, chercher et proposer des objets d'apprentissage personnalisés aux apprenants.

Synthèse sur les architectures sensibles au contexte :
Selon (Daruwala, 2008), un système sensible au contexte doit répondre à certaines exigences qui sont les suivantes :

1. *Support de l'hétérogénéité :* Le système doit supporter la diversité des capteurs

7. http ://en.wikipedia.org/wiki/IMS_Learning_Design

physiques, des dispositifs mobiles, des langages de modélisation, etc.

2. *Support de la mobilité :* Les protocoles de communication doivent supporter la mobilité des dispositifs et des capteurs.
3. *Protection de la vie privée :* Les flux d'informations contextuelles entre les composants du système doivent être contrôlés selon les besoins et les exigences de protection de la vie privée des utilisateurs.
4. *Traçabilité et contrôle :* L'état des composants du système et les flux d'informations entre les composants doivent être ouverts à l'inspection pour fournir la compréhension adéquate et le contrôle du système pour les utilisateurs.
5. *Tolérance aux échecs :* Les composants du système peuvent éventuellement rencontrer des défaillances. Le système doit continuer les opérations sans exiger des ressources excessives, et détecter des échecs.
6. *Déploiement et configuration :* Les matériels et les logiciels du système doivent être facilement déployés et configurés pour atteindre les exigences des utilisateurs.

Le tableau 2.2 synthétise les principaux aspects de chaque architecture présentée précédemment :

Critère	Toolkit	CoBra	CMF	SOCAM	CASS	Architecture en couches	m-LOCO
(1)	-	++	++	-	+	+	++
(2)	-	+	+	+	++	+	+
(3)	+	+	+	-	-	-	-
(4)	+	+	+	-	+	+	+
(5)	-	-	-	-	-	-	+
(6)	++	+	+	+	+	++	+

TABLE 2.2 – Exigences pour les systèmes sensibles au contexte (Daruwala, 2008)

En nous basant sur le tableau 2.2 nous constatons qu'aucune des architectures présentées ne remplit tous les critères demandés pour l'implémentation d'un système sensible au contexte. L'architecture en couches, l'architecture CASS et l'architecture m-LOCO peuvent être considérées comme modèles pour les concepteurs de systèmes, mais elles manquent encore de solutions pour supporter la vie privée et la tolérance aux échecs. Il en est de même pour l'architecture SOCAM, qui en plus ne supporte pas l'hétérogénéité et la traçabilité. L'architecture de CMF et l'architecture CoBra (qui se base sur Toolkit) sont celles qui répondent à presque tous les critères.

2.3/ CONCLUSION

Dans ce chapitre, nous avons présenté un état de l'art sur le contexte et la sensibilité au contexte. Nous avons mis en avant la nécessité de prendre en compte le contexte dans

la conception d'un système d'apprentissage mobile, et d'intégrer ce contexte tout au long du cycle de conception. A travers différentes définitions du contexte, nous avons étudié les éléments significatifs du contexte. Le contexte est vu comme un ensemble d'informations qui concernent l'utilisateur, son environnement et ses activités, qui pourront être utilisées pour faciliter les tâches de l'utilisateur. Cet ensemble est structuré, partagé et évolue au cours du temps. Une organisation du contexte selon différentes dimensions a été proposée. L'utilisation des ontologies pour la modélisation du contexte devient de plus en plus répandue. Ceci est principalement dû aux propriétés formelles des ontologies qui permettent d'assurer l'interopérabilité au niveau sémantique. La sensibilité au contexte est un concept indispensable dans la conception de systèmes d'apprentissage mobile. L'étude des architectures de systèmes sensibles au contexte nous a permis de dégager des principes pour concevoir un système de contextualisation pour l'apprentissage mobile. L'architecture en couches constitue un cadre général pour la conception, l'acquisition d'informations contextuelles et l'adaptation des contenus aux utilisateurs. L'objectif est de rendre le système capable de réagir de façon pertinente aux changements de l'environnement dans lequel se situe l'apprenant.
On en arrive ainsi à l'adaptation ou encore la recommandation de contenus d'apprentissage dans un contexte mobile. Ainsi le chapitre suivant, est dédié à l'étude des systèmes adaptatifs et des systèmes de recommandation pour l'apprentissage mobile.

3

SYSTÈMES DE RECOMMANDATION

La majorité des solutions mises en pratique ces dernières années dans le domaine de l'apprentissage électronique privilégient une approche centrée sur la mise à disposition de ressources pédagogiques de qualité, souvent au détriment de leur exploitation au sein de contextes pédagogiques diversifiés. De même, en apprentissage mobile, l'exploitation des contenus pédagogiques par les apprenants mobiles est complexe lorsque ces derniers doivent rechercher des contenus en rapport avec leur situation réelle. Le problème en apprentissage mobile n'est pas tant la disponibilité ni la qualité de l'information, mais sa pertinence relative à un contexte d'utilisation spécifique. La qualité d'un système d'apprentissage mobile dépend, dès lors, de sa capacité à fournir aux apprenants, d'une part des contenus pédagogiques adaptés à leur contexte et d'autre part, des processus qui les guident véritablement dans leur processus d'apprentissage. Le développement de systèmes d'apprentissage adaptatifs répond à cet objectif.

Un système d'apprentissage adaptatif utilise trois types de connaissances : les connaissances sur les contenus à apprendre, les connaissances sur les apprenants (profil et environnement), et finalement les connaissances sur les processus d'apprentissage. Un système d'apprentissage adaptatif met aussi en œuvre des méthodes et des techniques d'adaptation pour construire des parcours personnalisés en utilisant ces trois types de connaissances. Cette architecture minimale est proposée à la base, par les travaux menés sur les systèmes hypermédia adaptatifs (pour plus de détails voir l'annexe A).

Seulement ces systèmes posent encore des problèmes relatifs à la gestion de l'information (surcharge cognitive). Une réponse à la surcharge informationnelle est les systèmes de recommandation. Dans une deuxième partie, ce chapitre présente une étude des systèmes de recommandations. Nous exposerons les différentes techniques de recommandation, et leurs apports pour la résolution du problème de surcharge informationnelle en apprentissage électronique. Ce chapitre reprend certaines idées développées par Romain Picot-Clémente lors de sa thèse au sein de l'équipe Checksem (Picot-Clémente, 2011).

Sommaire

3.1 Introduction aux systèmes de recommandations 58
3.2 Approches de recommandation 60
3.2.1 Recommandations basées sur le contenu 60
3.2.2 Recommandations par filtrage collaboratif 65
3.2.3 Avantages et inconvénients des approches de recommandation 67
3.2.4 Approches hybrides 68
3.2.5 Systèmes de recommandation sensibles au contexte 70
3.3 Conclusion 70

3.1/ INTRODUCTION AUX SYSTÈMES DE RECOMMANDATIONS

Les systèmes de recommandation sont devenus un domaine de recherche indépendant dans le milieu des années 1990 (Goldberg et al., 1992). Ces dernières années, avec le développement du Web et plus particulièrement des plateformes e-commerce, l'intérêt pour les systèmes de recommandation a considérablement augmenté. A l'origine, les systèmes de recommandations sont apparus pour essayer de résoudre les problèmes liés à la surcharge informationnelle (surcharge cognitive).

Ce sont des outils capables de fournir des recommandations personnalisées ou permettant de guider l'utilisateur vers des ressources intéressantes ou utiles au sein d'un espace de données important (Burke, 2002). Les systèmes de recommandation sont essentiellement orientés vers les individus qui n'ont pas suffisamment d'expérience personnelle ou de compétences pour évaluer la quantité potentiellement immense d'informations qu'un site Web, par exemple, peut offrir (Resnick and H.R.Varian, 1997).
Le développement des systèmes de recommandation s'est initié à partir d'une observation assez simple : les individus s'appuient souvent sur les recommandations des autres pour la prise de décisions quotidiennes (McSherry and Mironov, 2009). Par exemple, il est commun de s'appuyer sur ce que nos semblables recommandent lors du choix d'un livre à lire ; les employeurs comptent sur les lettres de recommandation pour leurs décisions de recrutement ; et pour la sélection de films à regarder, les individus tendent à lire et se fier aux critiques de films.
Aujourd'hui, le domaine d'application de ces systèmes de recommandation est très large. Ils font partie intégrante des sites de e-commerce. Les articles les plus recommandés sont les films, les livres, la musique, les photos, les jeux, etc. Les services les plus recommandés sont les voyages, les restaurants, les hôtels, les pages Web, etc. Un bon exemple de système de recommandation, est celui utilisé par le site Web Amazon. Le site a en effet accru significativement le nombre de ses visites et de ses ventes grâce à cet outil. Certains utilisateurs vont même maintenant sur ce site principalement dans le

but de bénéficier de ces recommandations, même s'ils n'ont pas l'intention d'y acheter de produit. Et il en est de même pour des sites tels que Allociné pour les cinéphiles, Deezer pour la recommandation de musiques, Gameloft pour la recommandation de jeux vidéos, lasminute.com pour la recommandation de séjours touristiques ou encore Tripadvisor pour la recommandation d'hôtels et de restaurants.

Les deux entités de base qui apparaissent dans tous les systèmes de recommandation sont l'*item* et l'*utilisateur* : Item est le terme général utilisé pour dénoter ce que le système recommande aux utilisateurs. Afin de proposer des suggestions utiles et pertinentes pour un utilisateur en particulier, un système de recommandation va se focaliser sur un type spécifique d'item (par exemple, des fragments de pages Web, des livres, des images, etc.) et en conséquence, va personnaliser son modèle de navigation, son interface, ses techniques de recommandation, etc. Les utilisateurs doivent être modélisés dans le système de recommandation afin que le système puisse exploiter leurs profils et préférences. Par ailleurs, une description précise et claire des contenus est également nécessaire pour obtenir de bons résultats au moment des recommandations (Vozalis and Margaritis, 2003).

Les systèmes de recommandation ont prouvé ces dernières années qu'ils sont un bon moyen pour faire face au problème de surcharge cognitive. En effet pour résoudre ce problème, un système de recommandation met en avant des items inconnus qui peuvent être pertinents pour les utilisateurs. Ce niveau de pertinence est déterminé par le système en fonction de connaissances variées (profil de l'utilisateur, contexte de consommation, items disponibles, historique des transactions, feedbacks d'autres utilisateurs sur l'item, etc.). L'utilisateur peut alors parcourir les recommandations et peut fournir un feedback implicite ou explicite. Par exemple, dans une plateforme e-learning un feedback peut être des notes ou des avis que les apprenants peuvent attribuer au contenus. Toutes les actions et les feedbacks des utilisateurs peuvent être enregistrés dans la base de données du système de recommandations, et par la suite utilisés pour générer de nouvelles recommandations.

Il est possible de classer les systèmes de recommandation de différentes manières. La classification la plus connue est une classification selon deux approches : la recommandation basée sur le contenu et la recommandations par filtrage collaboratif (Shahabi et al., 2001) (Picot-Clémente, 2011).

– *La recommandation basée sur le contenu :* Le système recommande des items qui sont similaires à ceux que l'utilisateur a aimé dans le passé. La similarité des items est calculée en se basant sur les caractéristiques associées aux items comparés. Par exemple, si l'utilisateur a noté positivement un film qui appartient au genre comédie, alors le système peut fournir des recommandations de films de ce genre.
– *La recommandation par filtrage collaboratif* : Le système demande aux utilisateurs d'évaluer des ressources, de sorte qu'il sache ce qu'ils aiment le plus. Puis, quand une recommandation est demandée pour l'utilisateur courant, lui seront alors proposées

des ressources que des utilisateurs semblables à lui ont aimé. Le filtrage collaboratif est la technique la plus populaire et la plus répandue dans les systèmes de recommandation.

Nous décrivons brièvement ces différentes approches de recommandations dans les sections suivantes.

3.2/ APPROCHES DE RECOMMANDATION

3.2.1/ RECOMMANDATIONS BASÉES SUR LE CONTENU

Les systèmes de recommandation basés sur le contenu sont à l'intersection des domaines des systèmes de recherche d'informations et de l'intelligence artificielle (Ricardo and Ribeiro-Neto, 1999).
La technique de recommandation basée sur le contenu repose sur l'hypothèse que des items ayant des contenus similaires seront appréciés pareillement (Schafer et al., 2007). Cette technique est fondée sur l'analyse des similarités de contenu entre les items précédemment consultés par les utilisateurs et ceux qui n'ont pas été encore consultés (Burke, 2002).
Les items qui peuvent être recommandés aux utilisateurs sont représentés par un ensemble de caractéristiques, aussi appelées attributs, variables ou propriétés dans la littérature. Par exemple, dans une application de recommandation de films, les attributs adoptés pour décrire un film sont : acteurs, directeurs, genres, sujet, etc., ou encore dans une plateforme e-learning les attributs adoptés pour décrire un contenu sont : titre, auteur, thème, etc. Un item est représenté dans le système au moyen d'une donnée structurée. Plus formellement, cette donnée structurée est un vecteur $X = (x_1, x_2, \cdots, x_n)$ de n composantes. Chaque composante représente un attribut et peut contenir des valeurs binaires, numériques ou encore nominales.

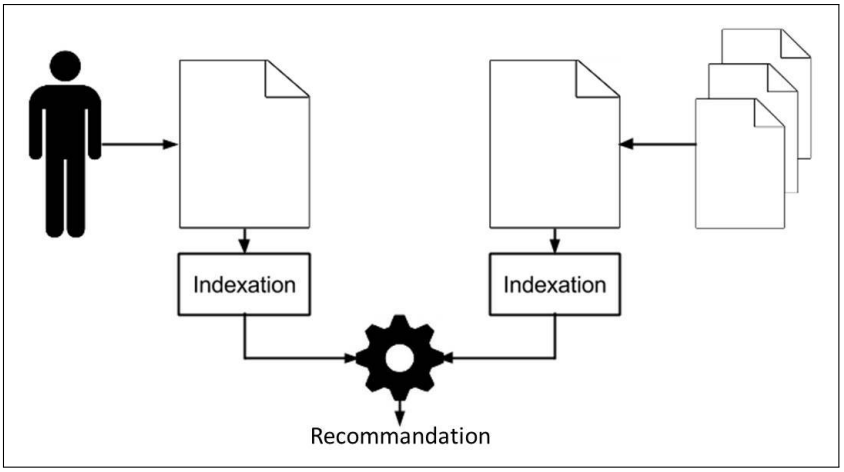

FIGURE 3.1 – Recommandation basée sur le contenu

Ainsi, le processus principal réalisé par un système de recommandation basé sur le contenu consiste à faire correspondre les attributs d'un profil utilisateur (où les préférences et intérêts sont stockés) avec les attributs des items d'un contenu, dans le but de recommander à l'utilisateur de nouveaux objets intéressants (figure 3.1). Cette démarche est souvent passive, c'est le système qui déduit, qui analyse, qui apprend le profil utilisateur et non l'utilisateur qui le fournit.
Dans la plupart des systèmes de recommandations basés sur le contenu, les descriptions des items sont des caractéristiques textuelles extraites à partir de pages Web, emails, articles de news ou descriptions de produits. Les items correspondent souvent à des mots-clés. Exemple, pour un système de recommandation d'articles scientifiques basé sur le contenu, lorsqu'un utilisateur a tendance à consulter souvent des articles portant sur le domaine du Web sémantique, le système lui proposera alors des articles liées à ce domaine. En effet, ces articles disposent de mots-clés communs (extraits sur la base d'une indexation automatique, ou attribués manuellement) tels qu'owl, sémantique ou ontologie. La précision de cette approche est donc hautement dépendante de la nature des contenus : elle est beaucoup plus élevée pour des contenus textuels que pour des contenus tels que les images, les vidéos ou les ressources audio, dont il est difficile d'extraire des attributs. En général, quand cette approche est employée pour des ressources non textuelles, des méta-données sont utilisées. On peut citer par exemple, les métadonnées du Dublin Core [1] pour la description de contenus (titre, auteur, sujet, etc.). Nous pouvons également citer les travaux de (Lainé-Cruzel, 1999) qui permettent de définir des propriétés liées à l'ensemble d'un document (profession de l'auteur, type du contenu, etc.) ainsi que celles relatives à des parties du contenu (forme, style, etc.) afin de restreindre les contenus pertinents aux seuls contenus exploitables et réellement utilisables. De même, une liste non exhaustive de métadonnées pour l'annotation qualitative de documents est donnée par (Berti-Equille, 2002) dans le contexte de la recommandation multi-critères. Cependant aujourd'hui, la plupart des recherches sur la recommandation basée sur le contenu portent sur des données textuelles (Adomavicius et al., 2005). Dans ce cadre, la mesure la plus populaire pour l'analyse du contenu est la mesure TF-IDF(Term Frequency-Inverse Document Frequency) (Salton, 1989) que nous présentons dans la section suivante.

Recommandation basée sur les mots-clefs :
La plupart des systèmes de recommandation basés sur le contenu utilisent de simples modèles de recherche initialement utilisés en recherche d'informations, comme la correspondance de mots-clefs ou le Modèle d'Espace Vectoriel (MEV), et les couples avec la pondération basique TF-IDF. MEV est une représentation spatiale des documents textuels et la mesure TF-IDF est une mesure statistique qui permet d'évaluer l'importance d'un mot dans un document ou dans un item faisant partie d'une collection de documents (Pazzani and Billsus, 2007). Dans ce modèle, chaque document est représenté par un

1. http ://dublincore.org/documents/dces/

vecteur de dimension n, où chaque dimension correspond à un terme de l'ensemble du vocabulaire d'une collection de documents. Formellement, tout document est représenté par un vecteur poids sur des termes, où chaque poids indique le degré d'association entre le document et le terme :
Soit $D = \{d_1, d_2, \cdots, d_N\}$ dénotant un ensemble de N documents ou corpus, et $T = \{t_1, t_2, \ldots, t_n\}$ le dictionnaire, ou l'ensemble des mots du corpus. T est obtenu en appliquant des opérations de traitement du langage naturel, comme l'atomisation (tokenization), l'élimination des mots vides de sens, et la troncature (stemming) (Baeza-Yates and Ribeiro-Neto, 1999). Chaque document d_j est représenté par un vecteur dans un espace vectoriel à n dimensions, tel que $d_j = \{w_{1j}, w_{2j}, \ldots, w_{nj}\}$, où w_{kj} est le poids du terme t_k dans le document d_j, obtenu avec le calcul TF-IDF :

$$TF(t_k, d_j) = \frac{f_{k,j}}{max_z.f_{z,j}} \tag{3.1}$$

où le maximum est calculé sur la fréquence $f_{z,j}$ de tous les termes t_z qui apparaissent dans le document d_j,

$$TFIDF(t_k, d_j) = TF(t_k, d_j).log\frac{N}{n_k} \tag{3.2}$$

où N dénote le nombre de documents dans le corpus, et n_k représente le nombre de documents de la collection dans lesquels le terme t_k apparait au moins une fois.
L'étape suivante consiste à normaliser les poids obtenus par la fonction TF-IDF. Afin que les documents soient représentés par des vecteurs de même longueur les poids obtenus sont généralement normalisés par la fonction cosinus :

$$w_{j,k} = \frac{TFIDF(t_k, d_j)}{\sum_{s=1}^{|T|} TFIDF(t_k, d_j)^2} \tag{3.3}$$

Ensuite, pour déterminer la proximité entre deux documents il faut définir une mesure de similarité des vecteurs caractéristiques. Il existe de nombreuses mesures de similarité, mais la plus largement utilisée dans le domaine de la recherche d'information est la similarité cosinus qui consiste à calculer la similarité entre deux vecteurs en déterminant le cosinus de l'angle entre eux (Salton and McGill, 1983) :

$$sim(d_i, d_j) = \frac{\sum_k w_{k,i}.w_{k,j}}{\sqrt{\sum_k w_{k,i}^2}.\sqrt{\sum_k w_{k,j}^2}} \tag{3.4}$$

De la même manière, les profils utilisateurs sont représentés par des vecteurs de termes pondérés. La prédiction de l'intérêt d'un utilisateur sur un item donné est ensuite effectuée

par calcul de similarité entre le vecteur de profil de l'utilisateur et le vecteur de contenu en utilisant ici aussi la similarité cosinus.

Les systèmes de recommandation basés sur les vecteurs mots-clefs ont été développés et utilisés dans de multiples domaines d'applications. Chaque domaine présente différents problèmes qui requièrent différentes solutions : Exemple 1, dans le domaine des systèmes de recommandation Web, on peut citer comme exemple le système Letizia (Lieberman, 1995) qui est implémenté comme une extension de navigateur Web traquant le comportement de l'utilisateur et qui construit un modèle personnalisé constitué des mots-clefs liés aux intérêts de l'utilisateur. Il s'appuie sur un feedback implicite pour inférer sur les préférences de l'utilisateur. Par exemple, ajouter une page en favoris est interprété comme une preuve évidente de l'intérêt de l'utilisateur pour cette page.
Exemple 2, dans le domaine de la recommandation d'actualités, on peut citer yourNews(Ahn et al., 2007) ou encore NewT (NewsTailor) (Sheth and Maes, 1993). yourNews contient un système classique fonctionnant à l'aide de vecteurs pondérées avec la mesure TF-IDF et comparés à l'aide de la similarité cosinus. Il es est de même pour NewT. Les articles recommandés par les systèmes en flux en fonction de le leur thème. Ensuite, un agent de recommandation est utilisé indépendamment pour chaque thème. Ainsi plusieurs agents de recommandation sont formés dans le même système pour différents types d'information, par exemple, un pour les actualités politiques, un pour le sport, etc.

La recommandation basée sur les mots-clefs présente plusieurs avantages :
– *Démarrage à froid :* La recommandation basée sur les mots-clefs permet de générer des recommandations malgré une situation de démarrage à froid qui se produit par l'introduction d'un nouvel item dans le système de recommandation. En effet quand de nouveaux items sont introduits dans le système, ils peuvent être recommandés directement en faisant une analyse de similarité, sans que cela ne nécessite un temps d'intégration.
– *Précision des recommandations :* La recommandation utilisant une représentation par mots-clefs à la fois pour les items et pour les profils permet de donner des résultats précis. Mais pour cela il est nécessaire d'avoir un nombre suffisant de preuves d'intérêts des utilisateurs dans le système.

La plupart des systèmes basés sur le contenu sont conçus comme des classificateurs de textes, construits à partir d'un ensemble de documents qui sont soit des exemples positifs, soit des exemples négatifs des intérêts de l'utilisateur. Le problème avec cette approche est le manque d'intelligence. De plus dans ce type de système de recommandations les représentations des profils utilisateurs ainsi que des items sont souvent hétérogènes, ce qui peut empêcher leurs utilisation et exploitation dans d'autres processus de recommandation. Des stratégies de recommandation plus avancées prenant en compte la sémantique associée aux profils utilisateur et aux items sont nécessaires. Nous présentons dans la section suivante un nouveau type de systèmes de recommandations

basés sur le contenu ; les systèmes de recommandations basés sur la sémantique.

Recommandation basée sur la sémantique :
Les systèmes basés sur la sémantique sont un cas particulier des systèmes basés sur le contenu. Ils tendent à intégrer les nouvelles technologies du Web sémantique, afin de remédier à certains manques des systèmes basés sur le contenu classiques :
SiteIF a été le premier système de recommandation à adopter une représentation basée sur le sens des documents pour construire un modèle des intérêts de l'utilisateur (Magnini and Strapparava, 2000). SiteIF est un agent personnel pour un site Web de nouvelles multi-lingue. La source externe de connaissance impliquée dans le processus de représentation est MultiWordNet, une base de données lexicale multi-lingues. Chaque news est automatiquement associée à une liste d'ensembles de synonymes (dits synsets) de MultiWordNet. Le profil utilisateur est un réseau sémantique où les nœuds représentent les synsets trouvés dans les documents lus par l'utilisateur. Durant la phase de correspondance, le système reçoit en entrée la représentation sous forme de synsets d'un document et le modèle utilisateur courant, et il produit en sortie une estimation de la pertinence du document en utilisant la Semantic Network Value Technique proposé par (Stefani and Strappavara, 1998).
ITR (ITerm Recommender) est un système capable de fournir des recommandations d'items dans plusieurs domaines (films, musique, livres), à condition que les descriptions d'articles soient disponibles sous forme de documents textes (Degemmis et al., 2007). Similairement à SiteIF, ITR intègre la connaissance linguistique au processus d'apprentissage des profils utilisateurs. La connaissance linguistique vient exclusivement de l'ontologie lexicale WordNet. Les items sont représentés selon des vecteurs basés sur des synsets, appelés bag-of-synsets (BOS). Dans le modèle BOS, un vecteur de synsets, plutôt qu'un vecteur de mots, correspond à un document. Le profil utilisateur est construit comme un classificateur binaire naïf bayésien de texte, capable de catégoriser un item comme intéressant ou non. Il comprend les synsets qui s'avèrent être plus révélateurs des préférences de l'utilisateur, selon la valeur des probabilités conditionnelles estimées dans la phase d'apprentissage. La correspondance item-profil consiste à calculer la probabilité qu'un item appartienne à la classe *intéressant*, par utilisation des probabilités des synsets dans le profil utilisateur.
Le système JUMP (Basile et al., 2007) est capable de délivrer intelligemment une information contextuelle et personnalisée à des travailleurs dans un environnement de travail non routinier. Les besoins des utilisateurs sont représentés sous la forme de requêtes complexes, plutôt que sous la forme d'un profil utilisateur. Un exemple de requête complexe est *Je dois préparer un rapport technique pour le projet m-learning*. L'analyse sémantique des documents et l'analyse des besoins de l'utilisateur sont basées sur une ontologie de domaine où chaque concept est manuellement annoté en utilisant des synsets de WordNet. La correspondance entre les documents et les concepts du domaine est réalisée automatiquement au moyen des procédures qui exploitent les annotations lexi-

cales dans l'ontologie de domaine. Pour le traitement de la requête donnée en exemple, toutes les instances des concepts rapport technique et projet, et les relations entre ces instances sont extraites de l'ontologie.
L'intérêt majeur pour les connaissances linguistiques est mis en valeur par la large utilisation de WordNet, qui est principalement utilisé pour l'interprétation du contenu en utilisant la désambiguïsation. D'un autre côté, les études décrites précédemment montrent que WordNet n'est pas suffisant pour la compréhension totale des intérêts des utilisateurs et pour leur contextualisation. Une connaissance spécifique au domaine et au contexte est nécessaire.

3.2.2/ RECOMMANDATIONS PAR FILTRAGE COLLABORATIF

Le filtrage collaboratif est parmi les technologies les plus populaires dans le domaine des systèmes de recommandation (Herlocker et al., 2000). Le filtrage collaboratif se base sur l'idée que les personnes à la recherche d'information devraient se servir de ce que d'autres ont déjà trouvé et évalué. Dans la vie quotidienne, si quelqu'un a besoin d'une information, il essaye de s'informer généralement auprès de ses amis, ses collègues, qui vont à leur tour lui recommander des articles, des films, des livres, etc. C'est à partir de ce constat que l'idée de filtrage collaboratif est née (figure 3.2). Selon (Goldberg et al., 2001) le filtrage collaboratif est l'automatisation des processus sociaux.

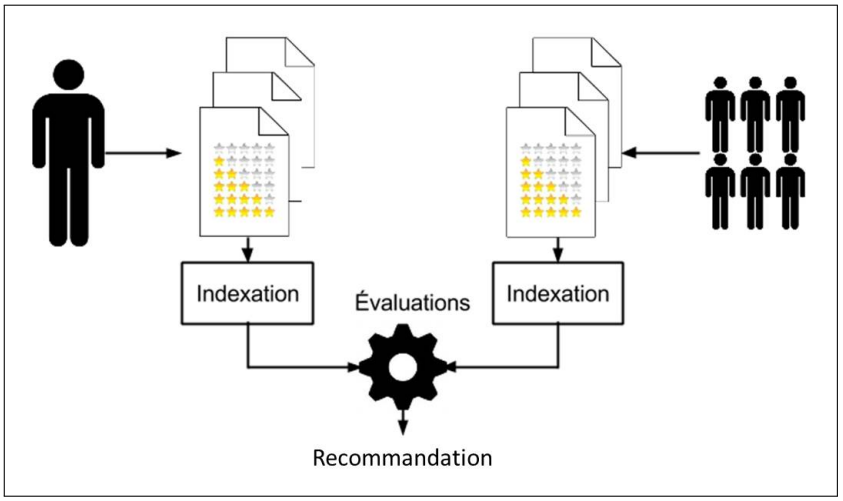

FIGURE 3.2 – Filtrage collaboratif

Le filtrage collaboratif utilise des méthodes statistiques pour faire des prévisions basées sur l'évaluation des intérêts des utilisateurs. Ces prévisions sont utilisées pour faire des propositions en se basant sur la corrélation entre son profil personnel et les profils des autres utilisateurs (qui présentent des intérêts semblables).
Le premier système de recommandation ayant été désigné comme étant un système de filtrage collaboratif est le système Tapestry (Goldberg et al., 1992). En réalité, ce système

était à la fois un système de recommandation et un système de recherche d'information puisque les utilisateurs pouvaient accéder à des messages électroniques à la fois en fonction des appréciations des autres utilisateurs et en formulant des requêtes. Les auteurs avaient appelé cette approche filtrage collaboratif car les utilisateurs pouvaient collaborer afin de mettre de côté les messages électroniques indésirables.
Cette expression a depuis beaucoup été reprise au sein de la communauté des chercheurs de ce domaine. Si elle est aujourd'hui encore communément utilisée, les systèmes dits de filtrage collaboratif d'aujourd'hui sont bien différents du système Tapestry. En effet, dans la représentation commune de la littérature actuelle, le filtrage collaboratif consiste à fournir des recommandations en exploitant exclusivement les notes attribuées par les utilisateurs pour regrouper ces derniers en fonction de leurs goûts communs. Ces systèmes de recommandation regroupent donc des ressources plutôt qu'ils n'en filtrent (filtrer consiste davantage à mettre de côté les ressources indésirables). Des utilisateurs soumettent des appréciations indépendamment les uns des autres et ne collaborent pas directement. Cette terminologie peut donc sembler inappropriée. Cette imprécision était déjà mise en avant en 1997 par (Resnick and H.R.Varian, 1997). Cependant, elle est à ce point répandue qu'il ne serait pas judicieux de proposer une alternative dans cette thèse, cette approche se situant au contour de notre problématique.
Plus techniquement, les méthodes de filtrage collaboratif produisent des recommandations d'items à des utilisateurs donnés, en se basant sur des schémas de notations ou d'usage (par exemple, les achats) sans besoin d'information sur les items et les utilisateurs. À la différence des approches basées sur le contenu, qui utilisent les items précédemment notés par un seul utilisateur u, les approches de filtrage collaboratif s'appuient sur les notes de tous les utilisateurs du système. L'idée clef est que la note de u pour un nouvel item i est susceptible d'être similaire à celle donnée par un autre utilisateur v, si u et v ont noté d'autres items d'une manière similaire. De même, u est susceptible de noter deux items i et j de la même façon, si d'autres utilisateurs ont donné des notes similaires à ces deux items.
Formellement, l'ensemble des items recommandables est représenté par le vecteur $X = \{x_1, x_2, \cdots, x_n\}$, de même l'ensemble des utilisateurs est représenté par le vecteur $U = \{u_1, u_2, \cdots, u_m\}$. Soit u_a un utilisateur actif pour lequel une recommandation doit être calculée. Chaque note de u_i pour chaque item x_j est une valeur numérique désignée par $\alpha_{i,j}$. La matrice de notes est alors la matrice suivante :

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & . & \alpha_{mn} \end{pmatrix}$$

Les approches collaboratives dépassent certaines limitations des approches basées sur le contenu. Par exemple, des items dont le contenu n'est pas défini, ou difficilement définissable peuvent quand même être recommandés aux utilisateurs grâce aux feedbacks des autres utilisateurs. De plus, les recommandations collaboratives sont basées sur la qualité des items évalués par les utilisateurs, au lieu de s'appuyer sur le contenu qui peut être un mauvais indicateur de qualité. Enfin, le fait que ce type de systèmes ne soit pas basé sur le contenu, permet de filtrer n'importe quel type de ressources qu'elles soient textuelle, image, vidéo, etc.
Mais contrairement aux approches basées sur le contenu, le filtrage collaboratif souffre du problème du démarrage à froid : Avant que le système puisse fournir des recommandations pertinentes à un utilisateur, il faut que ce dernier ait fourni, implicitement ou explicitement, des appréciations pour un nombre suffisant de ressources. Un problème supplémentaire par rapport aux recommandations basées sur le contenu est que cette limitation s'applique également aux nouvelles ressources introduites dans le système.

3.2.3/ AVANTAGES ET INCONVÉNIENTS DES APPROCHES DE RECOMMANDATION

Les différents types de recommandation ont leurs avantages et inconvénients. Dans cette section nous répertorions et expliquons les caractéristiques (les problèmes les plus récurrents, ainsi que les apports) des différentes stratégies de recommandation selon (Burke, 2002) :

1. *Cross-genre niches :* Le filtrage collaboratif se distingue par sa capacité à recommander à un utilisateur ce qui est hors du familier, c'est ce que Burke appelle : cross-genre niches. En effet, un utilisateur peut se voir recommander des items de genres différents. Par exemple, un utilisateur qui a des voisins similaires du point de vue des sports peut se voir recommander des recettes de cuisine si ces voisins aiment la cuisine et les recettes, même si cet utilisateur n'a jamais exprimé ce genre de favoris.
2. *Connaissance du domaine :* La connaissance du domaine n'est pas requise, le processus de recommandation se base uniquement sur les évaluations des items.
3. *Adaptabilité :* Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise.
4. *Feed-back implicite suffisant :* Un système de recommandation peut utiliser les habitudes d'écoute, sans avoir besoin d'une évaluation explicite.
5. *Problème de démarrage à froid, cas du nouvel utilisateur :* Un problème commun au filtrage basé sur le contenu et au filtrage collaboratif, est qu'un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
6. *Problème de démarrage à froid, cas du nouvel item :* C'est un problème qui concerne le filtrage collaboratif, et non pas le filtrage à base de contenu. Dans

le cas du filtrage à base de contenu, il suffit d'introduire l'item dans le système pour que celui-ci soit analysé et rentré dans le processus de recommandation. Dans le cas du filtrage collaboratif il doit avoir suffisamment d'évaluations pour que celui-ci soit pris en considération dans le processus de recommandation.

7. *Problème de démarrage à froid, cas du système débutant :* Le cas du système débutant provient lors du lancement d'un nouveau service de recommandation. Le système ne possède alors aucune information sur les utilisateurs et sur les items. Les méthodes de filtrage collaboratif ne peuvent pas fonctionner sur une matrice vide. La solution consiste en général à trouver des informations descriptives des items afin d'organiser le catalogue et inciter les utilisateurs à le parcourir jusqu'à ce que la matrice soit assez remplie et permette de passer en mode collaboratif.
8. *Le gray Sheep :* Quand des utilisateurs ont des goûts atypiques (qui varient de la norme), ils n'auront pas beaucoup d'utilisateurs en tant que voisins. Cela mènera à des recommandations pauvres. Ce problème est également connu comme gray sheep. La popularité est un problème qui se produit fréquemment dans le filtrage collaboratif.
9. *Le shilling* : C'est l'action malveillante d'influencer la recommandation, en créant de faux profils pour voter et favoriser/défavoriser certains items.

Le tableau 3.1, résume les forces et faiblesses des méthodes traditionnelles de recommandations.

Techniques de recommandation	**Forces**	**Faiblesses**
Recommandation basée sur le contenu	(2) (3) (4)	(5) (8) (9)
Recommandation par filtrage collaboratif	(1) (2) (3)	(6) (7) (8) (9)

TABLE 3.1 – Forces et faiblesses des approches de recommandation

3.2.4/ APPROCHES HYBRIDES

Les systèmes de recommandation représentent aujourd'hui un thème de recherche fondamental en plein essor, comme on l'a montré dans les sections précédentes. Pour résumer, actuellement, il existe deux grandes approches : la recommandation basée sur le contenu, et la recommandation basée sur le filtrage collaboratif. Dans la recommandation basée sur le contenu le profil décrit les caractéristiques de contenu que les documents intéressants sont censés présenter. Le système de recommandation compare les nouveaux documents au profil de chaque utilisateur et recommande ceux qui sont les plus proches. Dans la recommandation basée basée sur le filtrage collaboratif le profil est constitué de l'historique des évaluations émises par l'utilisateur sur des documents, sans qu'aucun élément de contenu ne soit rendu explicite. Le système de recommandation compare les utilisateurs entre eux sur la base de leurs évaluations passées pour créer

les communautés, et chaque utilisateur reçoit les documents jugés intéressants par sa communauté.
Chacune de ces approches, comme le montre le tableau 3.1 présente des avantages et des inconvénients. Pour pallier les inconvénients de chaque méthode de recommandation, de nombreux systèmes, dits hybrides , combinent les deux méthodes de recommandation de sorte que l'incapacité du filtrage collaboratif de recommander de nouveaux éléments soit réduite en le combinant avec des méthodes de recommandation basée sur le contenu (Li et al., 2010).
Plusieurs travaux de recherche ont proposé de combiner ou d'hybrider des techniques de recommandation. Parmi ces propositions de systèmes, celles qui nous semblent les plus intéressantes au regard de notre domaine sont :
(Andronico et al., 2003) propose un système de recommandation nommé InLinx, qui est un système hybride intégré dans un environnement d'apprentissage mobile. InLinx combine les techniques de recommandation basée sur le contenu et les techniques de recommandation par filtrage collaboratif, pour classer les différentes informations retrouvées sur le Web, recommander ces informations à des utilisateurs ayant des intérêts similaires, et les informer régulièrement des nouvelles informations potentiellement intéressantes. Les systèmes multi-agent ont aussi été intégrés dans son système. InLinx inclut ainsi, premièrement un processus de recommandation par filtrage collaboratif pour la classification de signets (informations marquées comme importantes), deuxièmement un processus de recommandation collaboratif pour le partage des signets. En entrée, le système reçoit différents documents potentiellement intéressants, puis tous les documents sont lexicalement analysés pour en extraire un ensemble de termes significatifs (signets), utilisés par la suite dans la phase de filtrage. Les termes sont classés suivant leur fréquence d'apparition dans un document, et le nombre de documents dans lesquels ils apparaissent. Finalement un processus de recommandation basée sur le contenu, est utilisé pour la recommandation de nouvelles sources. Chaque utilisateur du système correspond à un agent. Les agents utilisateurs apprennent le comportement des apprenants et s'adaptent aux changements d'intérêts de l'utilisateur, ce qui donne aux agents la possibilité d'être plus précis dans les classements suivants et dans le processus de recommandation. Plus les apprenants utilisent le système, plus les actions prochaines des agents sont précises.
(Liu, 2011) présente un travail portant sur le développement d'un système de recommandation basé sur le contexte hybride, et appliqué à l'apprentissage mobile. Sa proposition est un système qui mélange deux approches complémentaires pour atteindre de meilleures performances : Pour commencer, une recommandation initiale basée sur le contenu, qui tient compte des similitudes des modules des ressources et des profils utilisateurs, est appliquée. Ensuite une recommandation finale sur les résultats obtenus lors de la première étape sera effectuée, avec une approche de recommandation basée sur les règles prenant en compte le contexte de l'apprenant. Les règles se déclenchent lorsqu'une condition contextuelle spécifique se produit. Afin d'obtenir une recommandation

précise, les règles devraient envisager toutes les situations possibles progressives d'apprentissage. Le contexte actuel est décidé par la recherche dans un ensemble prédéfini de contextes avec une méthode de pattern matching. Ensuite, une recommandation est délivrée à l'apprenant, qui correspond à un ensemble d'informations utiles sur un sujet selon la base de règles contextuelles créées.

3.2.5/ Systèmes de recommandation sensibles au contexte

L'objectif des systèmes de recommandation est essentiellement d'améliorer les interactions entre les utilisateurs et les systèmes de recherche et d'accès à l'information. Seulement, les systèmes de recommandation discutés précédemment opèrent dans un espace bidimensionnel : (utilisateur, contenu). Or, dans les environnements mobiles, d'autres aspects sont à considérer dans la recommandation. Le profil de l'utilisateur est insuffisant pour produire des recommandations. Il y a plusieurs éléments dépendants du contexte qui devraient être considérés (quand ?, où ?, pourquoi ?, etc.). En effet, le contexte mobile et dynamique de l'utilisateur influence le choix de l'information pertinente.
La recommandation basée sur le contenu et le filtrage collaboratif ne sont pas adaptés (Chu and Park, 2009). Une nouvelle famille de systèmes de recommandation appelés systèmes de recommandation sensibles au contexte se développe de plus en plus dans la littérature ces dernières années, comme les travaux de (Camara et al., 2010), (Bader et al., 2011), (Baltrunas et al., 2012), (Djallel, 2014) qui se basent sur le contexte de l'utilisateur en modélisant des données comme sa localisation, l'heure et les personnes aux alentours. Ces systèmes sont à la base, des systèmes de recommandations classiques (filtrage basée sur le contenu ou filtrage collaboratif) mais qui intègrent de nouvelles composantes pour l'acquisition et le traitement du contexte. Le filtrage consiste ensuite à analyser le contexte courant de l'utilisateur et à sélectionner, parmi les contenus disponibles, ceux qui correspondent le mieux à la situation décrite.

3.3/ Conclusion

Dans ce chapitre, nous avons décrit les systèmes de recommandation, qui sont considérés comme un sous-ensemble des systèmes hypermédia adaptatifs, proposant une solution au problème de surcharge d'information par proposition de recommandations d'items. Le problème de recommandation d'items à un utilisateur a été formalisé et les différents types de systèmes de recommandation et leurs méthodes ont été exposées. Nous avons synthétisé cette étude dans la deuxième partie du chapitre, où nous avons identifié deux grandes catégories de systèmes de recommandation : les systèmes de recommandation basés sur le contenu et les systèmes de recommandation basés sur le filtrage collaboratif. Ces deux approches présentent néanmoins des caractéristiques complémentaires. Par conséquent les nouveaux travaux s'intéressent à différentes tech-

niques d'hybridation, qui en plus de profiter des avantages respectifs de ces approches, s'avèrent fournir des recommandations plus précises.
Dans le cadre du m-learning, l'adoption de la technique de recommandation basée sur le contenu présente de nombreux avantages par rapport à celle par filtrage collaboratif. En effet, un des avantages des systèmes de recommandation basés sur le contenu est qu'ils ne souffrent pas du problème de démarrage à froid, contrairement aux autres types de systèmes de recommandation. Il est alors préférable d'envisager de la recommandation basée sur le contenu, vu la situation dans laquelle se trouve l'apprenant, qui exige souvent une réponse et une réaction assez rapide du système. De plus, pour pouvoir exprimer pleinement le savoir-faire du formateur, il est nécessaire d'avoir une base de connaissance bien formalisée et complète. Enfin, dans notre contexte d'étude, nous devrons construire une solution de recommandation d'une combinaison pertinente d'items, où les items sont des objets pédagogiques atomiques. Pour atteindre cet objectif, nous devrons étendre les propositions actuelles de recommandations basées sur le contenu en intégrant à notre proposition des règles d'adaptation dans un environnement mobile et des algorithmes d'optimisation combinatoire.
Nous proposons une approche pour un système de recommandation pour le m-learning basé sur le contenu et sensible au contextuel intégrant des stratégies de recommandation de scénarios de formations sans risque de rupture.

III

APPROCHE SÉMANTIQUE ET COMBINATOIRE POUR UN SYSTÈME DE RECOMMANDATION SENSIBLE AU CONTEXTE

Les trois premiers chapitres de l'état de l'art nous ont permis d'identifier les éléments essentiels pour notre projet :
Dans le premier chapitre de l'état de l'art et à travers l'analyse des travaux existants dans le domaine du m-learning, nous avons dressé un aperçu des possibilités offertes par les environnements d'apprentissage mobiles. Nous avons pu constater que l'apprentissage reste limité car il n'est pas capable de s'adapter à son contexte.
Dans le deuxième chapitre nous avons étudié les caractéristiques des informations de contexte et les mécanismes utilisés dans les différentes phases d'adaptation au contexte qui sont l'acquisition des informations contextuelles, leur agrégation et stockage, et l'adaptation au contexte. L'adaptation au contexte nécessite la modélisation des informations contextuelles d'une façon abstraite. Ces informations se caractérisent par leur hétérogénéité. L'utilisation des ontologies pour la modélisation du contexte devient de plus en plus répandue grâce aux propriétés formelles des ontologies qui permettent d'assurer l'interopérabilité au niveau sémantique. Nous avons ensuite étudié différents systèmes sensibles au contexte existants. L'étude des architectures de ces systèmes nous a permis de dégager des principes pour concevoir un système de contextualisation pour l'apprentissage mobile.
Dans l'objectif de rendre un système m-learning capable de réagir de façon pertinente aux changements de l'environnement dans lequel se situe l'apprenant, nous avons étudié dans le troisième chapitre, en première partie les systèmes hypermédia adaptatifs, et en deuxième partie les systèmes de recommandation. Au niveau des systèmes hypermédia, une architecture formée de trois couches minimum est nécessaire pour la création d'un système hypermédia adaptatif. Des couches supplémentaires peuvent y être ajoutées selon les besoins d'adaptation et d'évolutivité désirés. Les systèmes de recommandation qui sont considérés comme un sous-ensemble des systèmes hypermédia adaptatifs, proposent une solution au problème de surcharge d'information. Suite à l'étude des différents types de recommandations, nous avons pu conclure que l'adoption de la technique de recommandation basée sur le contenu présente de nombreux avantages par rapport à celle par filtrage collaboratif. En effet, un des avantages des systèmes de recommandation basés sur le contenu est qu'ils ne souffrent pas du problème de démarrage à froid, contrairement aux autres types de systèmes de recommandation.

Dans cette partie du rapport, nous définissons un nouveau type de système de recommandation apparenté à la famille des approches basées sur le contenu, qui utilise une ontologie, mais aussi une modélisation du savoir-faire métier sous la forme de règles et des algorithmes d'optimisation combinatoire. Ce nouveau type de systèmes de recommandation se différencie de ceux présentés dans l'état de l'art, car il ne se limite pas à l'extraction d'items pertinents, mais il a pour objectif la construction de combinaisons pertinentes d'items en fonction du contexte actuel de l'apprenant. La combinaison d'items nécessite l'intégration d'algorithmes d'optimisation combinatoire. Néanmoins, nous verrons que chaque application possède son propre problème d'optimisation. Notre architecture devra donc être conçue de manière à accueillir des algorithmes adaptés aux

problèmes d'optimisation du domaine étudié.
Nous organisons ainsi cette partie en trois chapitres : Dans le premier chapitre nous introduisons la notion d'objet pédagogique (item) et nous nous positionnons par rapport aux différents modèles de description de ces objets pédagogiques pour la construction d'un contenu d'apprentissage. Dans le deuxième chapitre nous définissons de manière formelle un nouveau type de système de recommandation sensible au contexte basé sur des technologies du Web sémantique et des techniques d'optimisation combinatoire, proposant des combinaisons d'objets pédagogiques adaptées au contexte de l'apprenant. Dans le troisième chapitre nous décrivons formellement le problème d'optimisation combinatoire en m-learning et nous présentons et comparons l'efficacité de nos algorithmes de recommandation.

4

AUTOUR DES OBJETS PÉDAGOGIQUES

Les objets pédagogiques sont aujourd'hui au cœur de nombreux travaux au sein des institutions d'enseignement et des organismes de standardisation. Ces travaux visent principalement à permettre l'échange et la réutilisation des objets pédagogiques.
Plusieurs organismes de normalisation ont proposé de structurer un document pédagogique autour d'un ensemble d'objets pédagogiques. Généralement la description du contenu de ces objets pédagogiques est basée sur des métadonnées. Plusieurs différents modèles ont été mis au point au cours des dernières années. Ces modèles poursuivent généralement des buts distincts, mais s'accordent sur l'idée d'aboutir à des composants pédagogiques réutilisables.
Ce chapitre s'attache à étudier et classer les différentes approches et modèles proposés de représentation des objets pédagogiques. Tout d'abord, nous donnerons une définition de la notion d'objet pédagogique. Ensuite, nous décrirons les caractéristiques et le cycle de vie d'un objet pédagogique. Puis, nous présentons les principaux organismes de normalisation dans le domaine du e-learning et nous proposons une classification des modèles de métadonnées décrivant des objets pédagogiques. Cette classification nous permettra d'identifier le modèle le plus adaptée à notre système d'apprentissage mobile.

Sommaire

4.1 Objet pédagogique, définition et caractéristiques 78
4.1.1 Définition d'un objet pédagogique 78
4.1.2 Caractéristiques d'un objet pédagogique 79
4.2 Cycle de vie d'un objet pédagogique 80
4.3 Normes pour la description d'un objet pédagogique 83
4.3.1 Dublin Core 84
4.3.2 LOM 85
4.3.3 IMS-LD 87
4.3.4 SCORM 89

4.3.5 TIN CAN 92
4.3.6 LRMI 93
4.3.7 Que choisir pour notre modèle ? 93
4.4 Conclusion 95

4.1/ OBJET PÉDAGOGIQUE, DÉFINITION ET CARACTÉRISTIQUES

Les objets pédagogiques reposent sur un principe de base : celui de structurer le matériel pédagogique autour de petites unités réutilisables. Dans cette partie du chapitre nous analyserons les différentes définitions d'un objet pédagogique proposées dans la littérature, puis nous en déduirons ses caractéristiques.

4.1.1/ DÉFINITION D'UN OBJET PÉDAGOGIQUE

Le terme objet pédagogique, en anglais Learning Object (LO), est devenu d'usage à partir de 1994 lorsque Wayne Hodgins a baptisé les groupes de travail de l'association CEdMA [1] (Computer Education Managers Association). Dans cet ensemble, nous nous intéressons plus particulièrement au groupe de travail LTSC [2] (Learning Technology Standards Committee) de l'IEEE qui définit un objet pédagogique en 2002 comme « *toute entité numérique ou non, qui peut être utilisée pour l'enseignement ou l'apprentissage* » (IEEE-LTSC, 2002). Selon cette définition un objet pédagogique peut être un livre, tout document imprimé, un cours, un exercice, une étude de cas, une présentation, mais également une salle de cours, un rétroprojecteur, etc. Cette définition est donc trop générale, non utilisable en pratique et montre la difficulté à définir clairement ce qu'est un objet pédagogique. Cette définition nous amène à nous poser la question suivante : dans le cadre du e-learning, qu'est exactement un objet pédagogique ?

D'autres définitions plus restrictives sont proposées dans la littérature, dont celle de (Wiley et al., 2000). Dans sa définition il introduit la notion de réutilisabilité et définit un objet pédagogique comme « *n'importe quelle ressource numérique réutilisable dans un cadre d'apprentissage* », et exclut ainsi les entités non numériques telles que les livres.

Un regroupement d'universités du Wisconsin dont le but est de développer des ressources en ligne réutilisables définit les objets pédagogiques comme des « *petites unités d'apprentissage autonomes en ligne. Ils sont suffisamment petits pour être intégrés à une activité pédagogique, une leçon, un module ou un cours* ». Les auteurs rajoutent que les objets pédagogiques sont « *des petites unités d'une durée comprise entre 2 et 5 minutes* ». L'ajout d'une dimension temporelle est intéressante. De ce fait cette définition est encore plus restrictive car elle exclut tout ce qui n'est pas *petit*.

1. http ://www.cedma.org/
2. https ://ieee-sa.centraldesktop.com/ltsc/

D'autres définitions sont proposées afin de répondre à certains objectifs :
– Au niveau économique, l'objectif est de diminuer les coûts de production sans pour autant diminuer la qualité. Cela implique d'avoir des composants réutilisables et partageables.
– Au niveau pédagogique, l'objectif est de développer un processus de formation tout au long de la vie et de mettre en place des parcours de formation personnalisés à la carte. Cela implique d'avoir de petites unités réutilisables et adaptables.

Pour atteindre ces objectifs l'idée est d'utiliser des techniques rendant possible la réutilisation de ces petites unités pédagogiques dans de multiples contextes, tels que l'approche par objet ou encore les technologies du Web sémantique.

C'est en observant ses enfants jouer au LEGO^{TM} que Wayne Hodgins eu l'idée d'organiser les contenus pédagogiques autour de petits morceaux/unités tout comme les briques du jeu. Ces unités peuvent être des morceaux de texte, des figures, des graphes, des diaporamas, des séquences vidéo, etc. Ces unités peuvent être combinées pour créer par exemple une page en ligne. Ces mêmes unités peuvent être elles-mêmes combinées pour créer un cours, jusqu'à construire un cursus complet. La métaphore avec les briques de LEGO^{TM} sous-entend que chaque brique peut être combinée avec n'importe quel autre brique, de n'importe quelle façon. Afin d'éviter une mauvaise combinaison des unités pédagogiques nous distinguons deux classes d'objets pédagogiques. D'une part, les objets pédagogiques qui permettent de structurer un parcours de formation et l'organiser. D'autre part, les objets pédagogiques qui correspondent aux ressources physiques ou numériques nécessaires à la réalisation des activités d'apprentissage.

4.1.2/ CARACTÉRISTIQUES D'UN OBJET PÉDAGOGIQUE

A partir des différentes définitions présentées dans la section précédente, il est possible d'identifier plusieurs caractéristiques décrivant les objets pédagogiques, telles que la réutilisabilité, l'agrégation, l'accessibilité, etc. (O.Catteau, 2008) décrit ces principales caractéristiques :
– *La réutilisabilité :* Dès 1965, Nelson établit un cadre permettant de construire du contenu à partir d'objets réutilisables issus de différentes librairies électroniques interconnectées (Nelson, 1965). L'objectif de la réutilisabilité est d'avoir des objets pédagogiques élémentaires pouvant être utilisés dans des contextes et dans des buts multiples. Ces composants doivent par conséquent être autonomes. Ils peuvent être produits séparément, mais doivent pouvoir être modifiés pour correspondre aux besoins des utilisateurs. Par exemple, un auteur qui conçoit un objet pédagogique qui explique le fonctionnement d'un moteur doit idéalement éviter de faire référence à d'autres objets pédagogiques, car il pourrait être utilisé séparément dans un autre contenu et dans un autre contexte.
– *L'agrégation :* Un objet pédagogique peut certes être réutilisé, mais il peut aussi être

créé par agrégation d'autres objets pédagogiques de granularité plus fines.
- *L'accessibilité :* L'accès à un objet pédagogique est efficace lorsque le coût engendré par sa recherche en vue de sa réutilisation est inférieur au coût nécessaire pour créer un objet pédagogique équivalent. Afin de retrouver un objet pédagogique facilement il est nécessaire de le munir d'une description. Ce processus est appelé indexation. La technique la plus utilisée est l'étiquetage par des métadonnées afin de stocker et référencer un objet pédagogique dans une base de données.
- *La personnalisation* : Cette caractéristique dicte de créer des objets pédagogiques personnalisables pour faciliter l'intégration à des contextes particuliers permettant l'adaptation aux besoins de l'apprenant.
- *L'interopérabilité* : Les contenus pédagogiques sont conçus et développés par des organisations et des formateurs différents, constituant généralement des sources de données sémantiquement hétérogènes. De ce fait, l'interopérabilité entre ces contenus est complexe puisque chaque système peut possèder son propre modèle de description et d'encodage des objets pédagogiques et son propre langage d'interrogation. La recherche et la combinaison des résultats devient alors un travail long et fastidieux. Les objets pédagogiques doivent être autonomes, c'est-à-dire indépendants du support de diffusion et de la plateforme d'apprentissage. Cela soulève la notion d'interopérabilité, d'où l'apparition de plusieurs travaux pour la mise en place de standards et de normes pour la description d'un objet d'apprentissage.

La création d'objets pédagogiques respectant ces caractéristiques permet un apprentissage :
- *Sans superflu :* Si un apprenant a besoin d'une fraction de cours, ce dernier sera limité aux objets pédagogiques correspondants.
- *Sur mesure :* Les objets pédagogiques permettent une personnalisation des cours à l'échelle d'une organisation ou de chaque apprenant.
- *Au moment voulu :* Comme les objets pédagogiques sont indexés, on peut les retrouver instantanément.

La conception et la mise à disposition des objets pédagogiques nécessite un travail minutieux. La recherche et la combinaison d'objets pédagogiques requiert l'usage de systèmes homogènes. Cet ensemble forme le cycle de vie de l'objet pédagogique.

4.2/ CYCLE DE VIE D'UN OBJET PÉDAGOGIQUE

Le cycle de vie d'un objet pédagogique et ses métadonnées est une succession d'évènements liés à son évolution au cours du temps. Il s'agit surtout de formaliser une structuration méthodique des différents processus mis en œuvre. Comme le montre la figure 4.1, le cycle de vie d'un objet pédagogique est constitué de cinq étapes : (1) la création d'unités d'information (2) la description et l'indexation par des métadonnées (3)

l'agrégation(4) l'intégration des objets pédagogiques dans un scénario et finalement (5) la livraison aux apprenants (Fournier-Viger, 2005) :

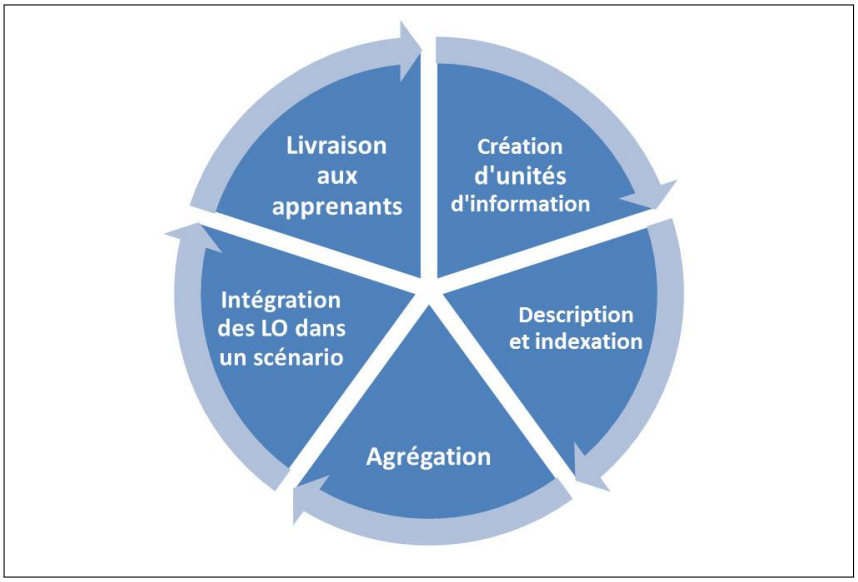

FIGURE 4.1 – Cycle de vie d'un LO

Création des unités d'information :
La première étape du cycle de vie d'un objet pédagogique consiste à créer une unité d'information ; une unité d'information peut avoir le format de pages Web, d'un applets Java, etc. et peut-être de type séquence vidéo, figure, texte, etc. De manière générale, l'auteur des unités d'information choisit le format en fonction de l'ensemble des logiciels avec lesquels il souhaite une compatibilité. Dans le domaine du e-learning, les institutions préfèrent généralement les documents Web comme unités d'information, afin de pouvoir les présenter à l'apprenant par l'intermédiaire de navigateurs Web.
Les auteurs doivent créer des unités d'informations (1) autonomes, (2) personnalisables, (3) de faible granularité et (4) qui ne sont pas neutres pédagogiquement (avec apport pédagogique). Les trois premiers principes visent à maximiser la capacité de réutilisation des unités d'informations, un objectif fondamental de l'approche des objets pédagogiques. Le quatrième principe dicte qu'un auteur d'une unité d'information ne doit pas viser la neutralité pédagogique, car cela mène souvent à une dégradation de la pertinence des objets pédagogiques.

L'indexation par des métadonnées :
La deuxième étape du cycle de vie d'un objet pédagogique consiste à ajouter des métadonnées aux unités d'information (objet pédagogique = unité d'information + métadonnées).
Les métadonnées sont des données représentant d'autres données (IEEE-LTSC, 2002), ici des informations décrivant un objet pédagogique. Elles sont représentées par un en-

semble de descripteurs et peuvent être structurées à l'aide d'un schéma. Par exemple, dans le cas d'un fichier informatique, le nom, la taille, la date de création représentent des métadonnées de ce fichier. Les métadonnées assurent (1) l'identification de manière unique d'une ressource (2) la recherche et l'organisation de ressources à partir de critères (3) et l'interopérabilité grâce à des descriptions pouvant être comprises aussi bien par un humain que par une machine, grâce aux processus de standardisation qui y sont associés (NISO, 2004).
Plusieurs initiatives de standardisation ont vu le jour au cours de ces dernières années. Les efforts entrepris par des consortiums comme ARIADNE [3] ou IMS [4] se sont orientés vers la description des unités d'information à l'aide de schémas de métadonnées. Ces travaux, sont basés en partie sur la description normalisée des documents proposée par le Dublin Core [5].

L'agrégation :
La troisième étape consiste à construire des environnements d'apprentissage à base des objets pédagogiques. Les objets pédagogiques sont assemblés pour construire de nouveaux objets pédagogiques sous forme d'une leçon, d'un cours, etc. Ces derniers peuvent à leur tour être agrégés pour former des ensembles plus vastes, comme un module, un parcours de formation ou un programme.
Les objets pédagogiques sont modulables et réutilisables sous forme de *paquetages de contenus* (appelés en anglais content packages dans la plupart des standards). Un paquetage regroupe le référencement des unités d'information par des métadonnées, la description de sa structure et les ressources physiques. Puisqu'un paquetage est aussi une unité d'information, si l'auteur ajoute les métadonnées appropriées, l'agrégat sera un objet pédagogique. Une spécification d'agrégation populaire est IMS, pour laquelle un paquetage est un fichier au format zip qui contient tous les objets pédagogiques et un fichier imsmanifest.xml qui agit comme une table des matières.

Création de scénarios pédagogiques :
La quatrième étape consiste à déterminer l'ordre de présentation (sequencing) des objets pédagogiques pour créer un scénario. Un scénario pédagogique définit une activité d'apprentissage par un ensemble de contraintes qui régissent l'ordre de présentation des objets pédagogiques en fonction des objectifs de l'apprenant.
Dans le cadre d'un apprentissage mobile nous remarquons que la création d'un scénario pédagogique passe par deux étapes :
– Premièrement un scénario initial est créé et permet de définir de manière générale la structure d'un scénario abstrait qui ne tient pas forcément compte du profil et du contexte de l'apprenant. Cette première étape permet de définir a priori et en termes

3. http ://www.ariadne-eu.org/
4. http ://www.imsglobal.org/
5. http ://dublincore.org/

généraux l'organisation et le déroulement d'un scénario d'apprentissage. Cette tâche peut être confiée à un auteur, un enseignant, un formateur, dans le cadre de la mise au point de ses propres séquences pédagogiques. Elle peut aussi être réalisée par un ingénieur spécialisé dans l'ingénierie pédagogique dans le cadre d'un processus industrialisé de formation. Dans notre cas cette étape est réalisée par les auteurs des contenus pédagogiques proposés par CrossKnowledge, ainsi que par des ingénieurs spécialisés de l'entreprise. Nous reprenons donc l'organisation (sequencing) proposée par CrossKnowledge et l'intégrons dans notre modèle.

– La deuxième étape consiste à contextualiser le scénario initial, c'est-à-dire déterminer les conditions d'exploitation d'un scénario initial dans un contexte précis en termes de profil apprenant, de support d'apprentissage, de localisation, etc. Cette étape est réalisée dans nos travaux en utilisant des règles métiers appliquées à notre modèle de description des objets pédagogiques.

Livraison à l'apprenant :
La dernière étape du cycle de vie consiste à délivrer les objets pédagogiques à l'apprenant. Les acteurs dans le domaine du e-learning emploient généralement un système logiciel Web développé pour accompagner toute personne impliquée dans un processus d'apprentissage. Ce système est appelé LMS pour Learning Management System ou MLE (Managed Learning Environment) ou VLE (Virtual Learning Environment) ou CMS (Course Management System) ou encore LSS (Learning Support System).
Chaque LMS se conforme à un certain nombre de standards. Habituellement les LMS proposent aussi une gamme de logiciels de gestion de comptes, de développement de cours, d'administration et des outils de communication (synchrones/asynchrones). Il existe un grand nombre de plateformes LMS sur le marché international, environ 300, dont certaines sous licence libre. Parmi ces plateformes nous pouvons citer : Claroline [6], Ganesha [7] , Moodle [8] , BlackBoard [9] , myTeacher [10], Stellar [11].

Dans la section suivante, nous nous intéressons plus particulièrement à la deuxième, troisième et quatrième étape du cycle de vie d'un objet pédagogique.

4.3/ NORMES POUR LA DESCRIPTION D'UN OBJET PÉDAGOGIQUE

En e-learning les métadonnées permettent une recherche sémantique efficace des ressources pédagogiques sur le Web et leur traitement automatique. Différentes normes ont été définies pour aider à l'élaboration des LMS et à la représentation des objets

6. http ://www.claroline.net/
7. http ://www.ganesha.fr/
8. http ://moodle.org/
9. http ://www.blackboard.com/International/EMEA/Overview.aspx ?lang=en-us
10. http ://www.cerfi.ch/fr/myTeacher/myTeacher.html
11. http ://stellar.mit.edu/

pédagogiques associés dans un scénario. L'application de ces normes garanti non seulement l'interopérabilité mais également la qualité du système.
Parmi les normes les plus connues en e-learning on peut citer le Dublin Core, LOM, IMS-LD, SCORM, TIN CAN et LRMI.

4.3.1/ DUBLIN CORE

Le Dublin Core tire son nom d'un groupe de travail qui s'est réuni en 1995 dans la ville de Dublin pour définir un tronc commun d'éléments utilisables par le gouvernement américain, pour la description de ressources numériques dans les registres de métadonnées officiels (par exemple, dans les domaines de la défense ou de la justice). Au départ, le standard visait essentiellement les bibliothèques numérisées, ensuite ont été ajoutées des extensions à des fins pédagogiques. Aujourd'hui le Dublin Core a un statut officiel au sein du W3C et de la norme ISO 15836.
Le Dublin Core est un schéma de métadonnées qui permet de décrire des ressources numériques ou physiques et d'établir des relations avec d'autres ressources. Il comprend officiellement 15 éléments de description formels (titre, créateur, éditeur, etc.), intellectuels (sujet, description, langue, etc.) et relatifs à la propriété intellectuelle [12]. Le tableau 4.1 regroupe les différents éléments du Dublin Core.

Propriété	**Description**
Titre	Titre principal de la ressource.
Créateur	Nom de la personne, de l'organisation ou du service à l'origine de la rédaction de la ressource.
Sujet	Mots-clefs, phrases de résumé, ou codes de classement. Il est préférable d'utiliser des mots-clefs choisis dans le cadre d'une politique de classement.
Description	Résumé, table des matières, ou texte libre.
Editeur	Nom de la personne, de l'organisation ou du service à l'origine de la publication de la ressource.
Contributeur	Nom d'une personne, d'une organisation ou d'un service qui a contribué à l'élaboration de la ressource
Date	Date d'un événement dans le cycle de vie de la ressource. Par exemple, date de création, date de modification, date de mise à disposition.
Format	Format physique ou électronique du document. Par exemple, taille, durée, matériel et logiciel nécessaires.
Identifiant	Identifiant de la ressource. Par exemple, URI, numéro ISBN.
Type	Genre du contenu.
Source	Origine dont dérive la ressource.

12. http ://fr.wikipedia.org/wiki/Dublin_Core

Langage	Langue utilisé par la ressource.
Relation	Lien vers une ressource liée. Par exemple, en utilisant les URIs.
Portée	Couverture spatiale (par exemple, coordonnées géographiques, pays) et couverture temporelle (par exemple, laps de temps, heure).
Droits	Statut des droits de la ressource (par exemple, copyright, un lien vers le détenteur des droits).

TABLE 4.1 – Liste des éléments du Dublin Core

Tous les éléments de Dublin Core sont optionnels et peuvent être répétés et présentés dans n'importe quel ordre. Des vocabulaires contrôlés sont suggérés mais pas imposés. Grâce à sa simplicité, le Dublin Core est très utilisé. Il peut servir à décrire des objets pédagogiques bien que son champ d'application soit plus vaste. Seulement sa simplicité devient une faiblesse car il ne permet pas de détailler suffisamment les propriétés d'un objet pédagogique, tel que nous le souhaitons dans le cadre d'un apprentissage mobile personnalisé. Néanmoins, il a largement inspiré un autre standard, cette fois ci spécifique aux objets pédagogiques. Il s'agit du standard LOM.

4.3.2/ LOM

LOM (Learning Object Metadata) est un schéma de description des objets pédagogiques par les métadonnées. Il peut être utilisé pour décrire des ressources aussi bien numériques que non numériques. Techniquement, son nom est IEEE 1484.12.1-2002. Il a été conçu, en 2002, par le groupe de travail IEEE-LTSC avec la participation de nombreux consortiums dont ARIADNE et IMS. LOM a été proposé à l'ISO pour devenir une norme internationale, mais malgré sa simplicité d'expression il a été refusé par la France et la Chine qui se sont opposés à sa publication en tant que norme ISO. Quelques années plus tard, en France, a été publiée par l'AFNOR en 2006 une adaptation de LOM à la norme française NF, le LOMFR2.
L'objectif de LOM est de définir des objets pédagogiques pouvant être utilisés de façon indépendante d'un contexte précis, et pouvant également être réutilisés. Cette réutilisation est rendue possible par une description normalisée à l'aide d'un vocabulaire prédéfini regroupant un ensemble précis de métadonnées.
La version actuelle de la norme propose une structure hiérarchique contenant 78 éléments sous forme d'un arbre à quatre niveaux : 8 éléments descriptifs de quatrième niveau, 13 éléments descriptifs de troisième niveau et 45 éléments descriptifs de deuxième niveau, tous regroupés en 9 catégories (figure 4.2).

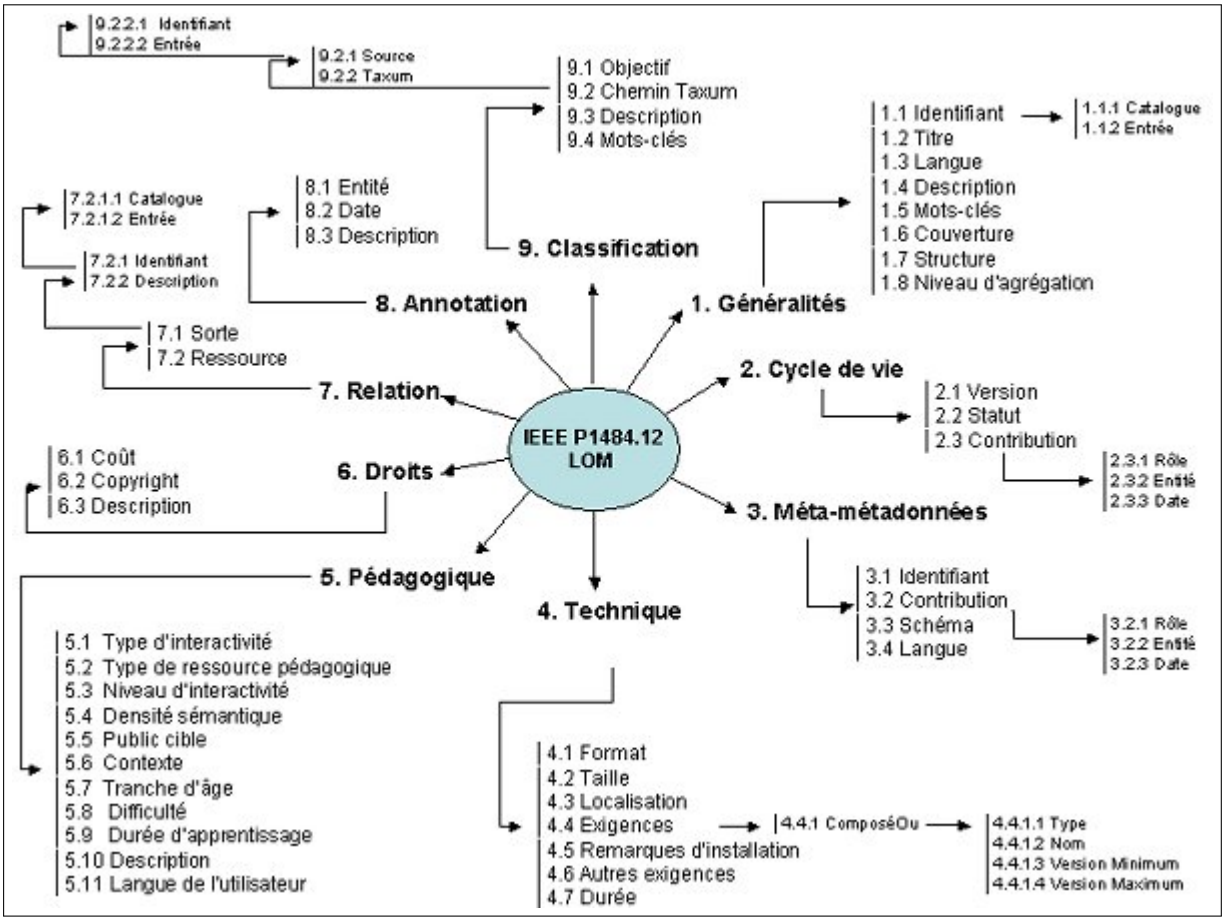

FIGURE 4.2 – Carte heuristique des métadonnées de LOM

LOM intègre les 15 champs du Dublin Core.
Dans ces neuf catégories on trouve [13]

– *Généralités* : Cette catégorie regroupe des caractéristiques indépendantes du contexte. L'objectif est la description et l'identification d'un objet pédagogique (par exemple, titre, langue, mots clefs).
– *Cycle de vie* : Cette catégorie décrit l'état actuel d'un objet pédagogique et qui y a contribué durant son évolution (par exemple, version, statut, entités contributives).
– *Méta-métadonnées* : Cette catégorie permet de garder une trace du schéma de métadonnées utilisé pour décrire les objets pédagogiques. Ces informations sont utiles pour envisager l'interopérabilité des différents systèmes et pour l'échange de données (par exemple, identifiant, langue).
– *Technique* : Cette catégorie regroupe les caractéristiques techniques d'un objet pédagogique (par exemple, format, taille, remarques d'installation).
– *Pédagogique* : Cette catégorie est le cœur du LOM. Elle accomplit une fonction de description pédagogique : le contexte d'utilisation (par exemple, scolaire, secondaire, supérieur, formation continue), le type de public cible (par exemple, enseignant, auteur, apprenant), le temps d'apprentissage type, etc.

13. http ://fr.wikipedia.org/wiki/Learning_Object_Metadata

– *Droits* : Cette catégorie spécifie les conditions d'utilisation d'un objet pédagogique (par exemple, coûts, droits d'auteur).
– *Relation* : Cette catégorie permet de lier les objets pédagogiques entre eux (par exemple, est partie de, est version de, est format de, est référencée).
– *Annotation* : Cette catégorie permet de laisser des remarques à propos d'un objet pédagogique, tout en gardant en mémoire la description de la personne qui commente et la date de ce commentaire.
– *Classification* : Cette catégorie permet d'assigner des classifications aux objets pédagogiques (par exemple, classification par thèmes).

Le modèle de métadonnées LOM est très complet. Seulement il prend insuffisamment en compte la représentation d'un objet pédagogique dans le cadre d'un scénario pédagogique. Actuellement, l'organisme ISO travaille sur l'amélioration du standard LOM pour la publication d'une nouvelle norme internationale pour décrire les contenus pédagogiques. IL s'agit du MLR [14] (Management Learning Resources). Les travaux sur MLR sont toujours en cours, uniquement une première partie de ce standard a été publié jusqu'à aujourd'hui. LOM est également repris par d'autres standards tels qu'IMS-LD et SCORM, nous les décrivons brièvement dans ce qui suit.

4.3.3/ IMS-LD

Les spécifications du langage IMS-LD (IMS Learning Design) ont été mises au point par le consortium IMS en 2003. Contrairement aux autres normes qui sont centrées sur la description d'un contenu pédagogique, cette spécification fait le postulat que ce sont avant tout les activités d'apprentissage associées qu'il faut mettre au centre de la modélisation. L'objectif d'IMS-LD est donc de fournir une infrastructure d'éléments capables de décrire n'importe quelle conception de scénario d'apprentissage d'une manière formelle.
L'approche dans IMS-LD est de définir un noyau complet d'éléments aussi simplement que possible, avec quelques extensions. IMS-LD repose sur un méta-modèle conceptuel et propose de décrire les scénarios d'apprentissage à l'aide de langages de modélisation pédagogiques permettant de définir les relations entre (1) les objectifs en termes de connaissance (2) les acteurs de l'apprentissage (3) les activités réalisées et (4) l'environnement et les contenus nécessaires à la mise en place du scénario.
La structure d'implémentation d'IMS-LD est composé de trois niveaux. Au niveau A on dispose de tous les éléments représentés dans la figure 4.3, à l'exception des éléments *propriété*, *condition* et *notification*. Les éléments de ce niveau permettent de décrire des unités d'apprentissage mais dont le schéma d'exécution sera identique d'une session à l'autre pour chaque utilisateur du système. En d'autres termes, le niveau A ne permet pas de personnaliser l'exécution d'un objet pédagogique. Le niveau B sert à la personnalisation des objets pédagogiques. La combinaison des concepts de propriétés et de

14. http ://sticef.univ-lemans.fr/num/vol2010/08r-bourda/sticef_2010_bourda_08r.htm

conditions introduits à ce niveau, permet de personnaliser la situation d'apprentissage. Le dernier niveau, le niveau C introduit le concept de notification, action déclenchée sur l'occurrence d'un événement particulier, par exemple fin d'un évènement, mise à jour d'une propriété, etc. L'élément notification permet ainsi de dynamiser l'exécution d'un objet pédagogique.

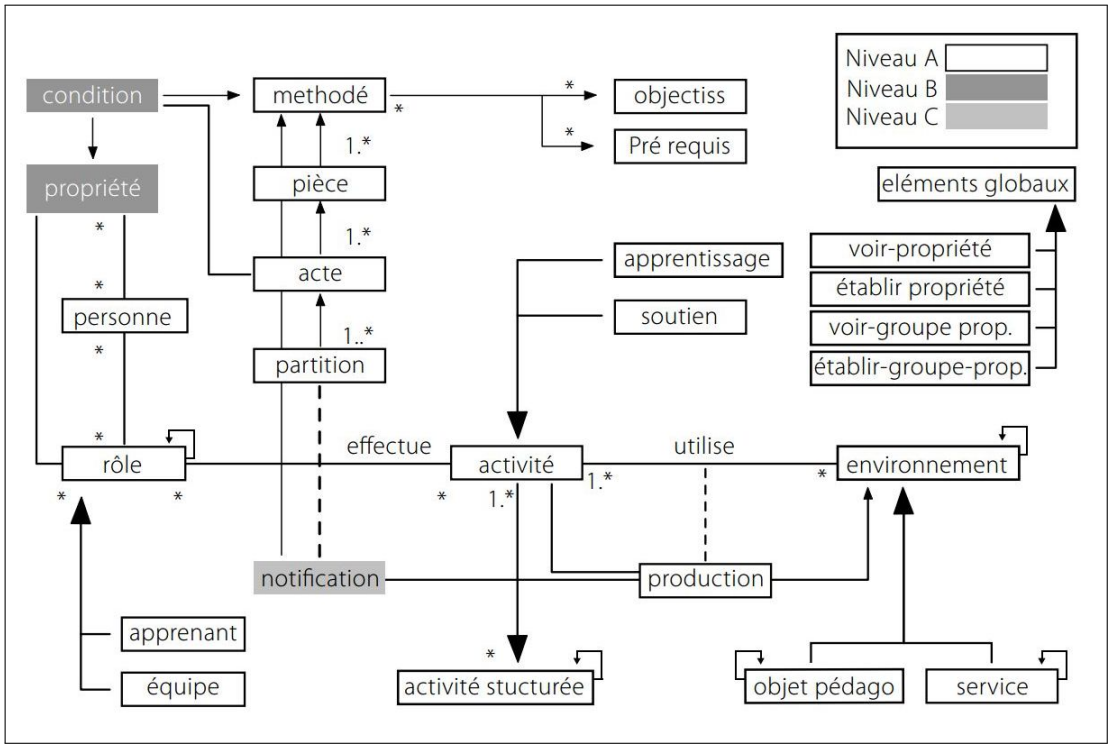

FIGURE 4.3 – Architecture de la spécification IMS-LD

Concrètement, au niveau A, un objet pédagogique est décrit dans un fichier compressé appelé *content package* (figure 4.4).

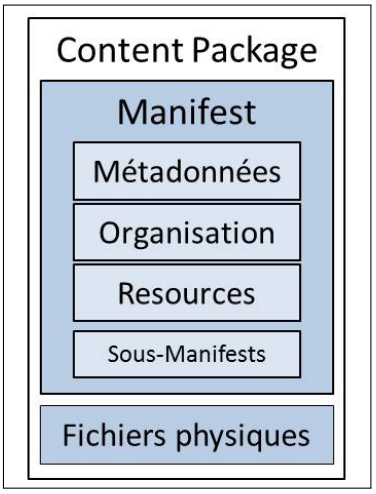

FIGURE 4.4 – Composition du content package

Ce dernier est composé d'un fichier XML, le *imsmanifest.xml*, et de l'ensemble des ressources physiques. Le manifest contient trois principales sections : une section qui regroupe les métadonnées de description de l'objet pédagogique (exemple, les métadonnées de LOM) ; une section organisation décrivant la structure et l'organisation de l'objet pédagogique ; et une section ressource permettant de déclarer tous les fichiers constituant l'objet pédagogique (déclaration en utilisant les URI). Le manifest peut aussi contenir des sous-manifests si d'autres objets pédagogiques sont imbriqués.

Le content package contient toutes les informations nécessaire sur un objet pédagogique dans un format XML qui peut être lu par n'importe quel système de diffusion ou plateforme conforme à la spécification IMS-LD et peut être réutilisé sur n'importe quelle outil équipé adéquatement. L'inconvénient d'IMS-LD est qu'il ne propose pas de concepts permettant de décrire le contexte des activités proposées. Le métamodèle IMS-LD a été conçu pour atteindre un certain niveau d'interopérabilité et pas un certain niveau d'expressivité, et ceci limite fortement sa capacité à décrire des apprentissages situés.

4.3.4/ SCORM

SCORM [15] (Sharable Content Object Reference Model) n'est pas une norme à proprement parler mais plutôt un agrégat de standards. Il récupère le meilleur des normes LOM et IMS-LD. La norme SCORM d'ADL [16] s'impose aujourd'hui comme le standard en matière de conception de cours et de plateformes e-learning. L'argument de taille de SCORM, c'est la garantie de l'interopérabilité entre les contenus e-learning et les LMS. Les LMS qui respectent le modèle SCORM peuvent ainsi intégrer n'importe quels contenus e-learning et inversement. Ceci assure la pérennité des contenus SCORM, puisqu'ils sont utilisables avec tous les outils de développement existants et ne sont pas influencés par une éventuelle évolution technologique sur le LMS ou d'un changement radical de plateforme pour leur diffusion.
Il y a deux principales versions de SCORM. La version la plus largement utilisée est SCORM 1.2. Une nouvelle version SCORM 2004 a été définitivement stabilisée en 2006.
SCORM propose une répartition des objets pédagogiques en trois niveaux : Les ressources numériques élémentaires qu'on appelle assets, les objets de contenu partageables qu'on appelle SCO (Sharable Content Object) et les agrégats de contenu :
Un asset est la forme la plus élémentaire d'un contenu. Cela peut être du texte, des images, des vidéos, des pages Web, etc. Autrement dit, un asset peut être tout type de ressource numérique pouvant être affichée par un navigateur internet. Seulement un asset ne communique pas avec un LMS. Les assets sont associés à des métadonnées qui décrivent leur contenu.
Un SCO est composé d'un ou plusieurs assets, ainsi que d'un dispositif de communica-

15. http ://scorm.com/scorm-explained/
16. http ://www.adlnet.org/

tion avec le LMS. En d'autres termes un SCO est une ressource au même titre qu'un asset mais qui communique avec une plateforme de formation. Les SCO sont des unités qui doivent rester relativement petites, mais il n'existe pas de normes pour définir sa taille, c'est alors au concepteur d'évaluer cette taille. Les SCO aussi bien que les assets sont associés à des métadonnées qui décrivent leur contenu et facilitent leur recherche. Un SCO est décrit de manière à pouvoir être réutilisé dans un autre cours et reconnaissable par une plateforme SCORM.
Pour définir un contenu SCORM, l'auteur va construire des SCO, qui sont les briques de base d'un cours, puis expliquer la façon de les lier entre eux. Ainsi plusieurs SCO forment un objet pédagogique et un ou plusieurs objets pédagogiques peuvent former un cours.
Les anciennes versions de SCORM 2004 se composent essentiellement de deux parties :

– *Le modèle d'agrégation de contenu :* Ce modèle correspond aux spécifications IMS-LD. Au cœur de la description SCORM nous retrouvons le fichier imsmanifest.xml. C'est par ce fichier que l'on va charger les informations décrivant la structure d'un parcours complet. Le fichier manifest contient la description des différents composants (les assets et les SCO) du modèle de contenu. Dans SCORM, des métadonnées sont attachées à ces composants. Les métadonnées doivent permettre d'identifier, de classer, d'informer sur les intentions de l'auteur et de chercher les objets pédagogiques. Les métadonnées utilisées respectent la spécification LOM. Le manifest contient aussi l'organisation des objets pédagogiques en hiérarchie. Finalement le manifest indique la liste des ressources physiques utilisées dans le parcours pédagogique (figure 4.5).

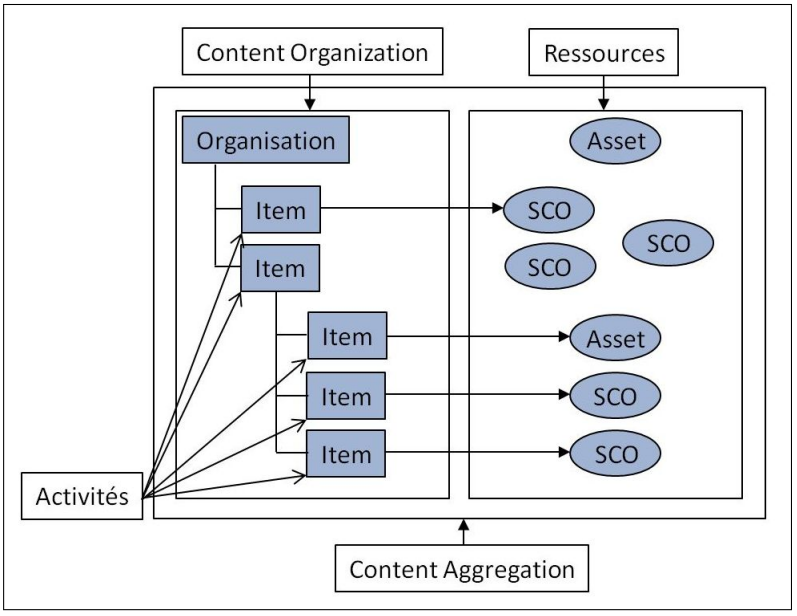

FIGURE 4.5 – Le modèle d'agrégation de contenu SCORM

– *Le modèle d'environnement d'exécution :* Une communication est nécessaire entre les objets pédagogiques et le LMS. Pour ce faire, ADL a établi un envoi normalisé d'informations dans les deux sens, et compatible avec les technologies Web. L'envi-

ronnement d'exécution fonctionne à partir d'une API (Application Program Interface) qui exploite un modèle de données permettant la standardisation des échanges entre un LMS et le navigateur de l'apprenant. Pour que les contenus soient interopérables il faut s'assurer que leur lancement sur une plateforme suit une procédure bien définie et qu'ils disposent de fonctions de communication clairement identifiées. La description du fonctionnement de SCOM est donné dans le livre blanc de (Amalric, 2007). Dans SCORM, le dialogue entre le LMS et les SCO se fait à l'aide d'une série de fonctions écrites en javascript et disponibles dans un objet API. Un SCO doit trouver cet objet pour interagir avec lui. Pour trouver cet objet API, il suffit d'ajouter du code javascript au début de la page de départ du SCO. Une fois L'API SCORM trouvé par le SCO, il est dès lors possible d'utiliser des fonctions de communication avec cet API (exemples : LMSInitialize () et LMSFinish() pour gérer l'état d'exécution d'une ressource, ou encore LMSGetValue() pour récupérer des valeurs du LMS et la transmettre à la ressource). Ainsi le développeur du contenu n'a pas à se soucier de la communication avec le serveur, il doit seulement trouver l'API et appeler les fonctions adéquates en JavaScript. Cette séparation du client et du serveur est essentielle avec SCORM et assure ainsi la portabilité des contenus (Amalric, 2007).

Par rapport aux anciennes versions de SCORM, SCORM 2004 intègre un nouveau modèle, le modèle de séquencement et de navigation. Ce modèle permet de mettre en œuvre des parcours de formation selon des situations plus complexes. Jusqu'à aujourd'hui, la complexité de ce modèle pose un frein à son intégration dans les LMS, par conséquent il n'est pas très utilisé. Nous décrivons brièvement le principe de fonctionnement de ce modèle. Le modèle de séquencement et de navigation décrit comment doivent être traitées les instructions de branchement et de coordination d'activités pédagogiques suivant un arbre d'activités. Une activité de séquençage correspond à un élément dans le manifeste du cours. Les activités sont imbriquées dans des relations parent/fils pour former l'arbre d'activités. L'arbre d'activités est dérivé de la structure définie par la section organisation du modèle d'agrégation de contenu (figure 4.6).
A chaque activité sont associées des données de suivi et des règles de séquençage. Les données de suivi représentent l'état actuel de l'activité (par exemple, son statut, son score) et les règles de séquençage définissent comment cette activité devrait être séquencée. Le résultat de l'application d'une règle permet par exemple de cacher une activité, de passer une activité, de quitter la formation, etc. Ces règles sont spécifiées dans le manifeste du cours.

SCORM présente certaines limites, par exemple, un SCO ne peut pas directement transmettre des données à un autre SCO. De plus le concept d'activité de l'apprenant n'est pas défini de façon explicite dans le modèle. SCORM privilégie donc une approche centrée davantage sur les contenus que sur les activités. Finalement certains développeurs de LMS annoncent que créer du contenu SCORM est une tâche lourde et peut donc présenter un coût de production élevé. Ils précisent que ce travail est pertinent, malgré

ces contraintes, si le contenu est par la suite réutilisé ou échangé.

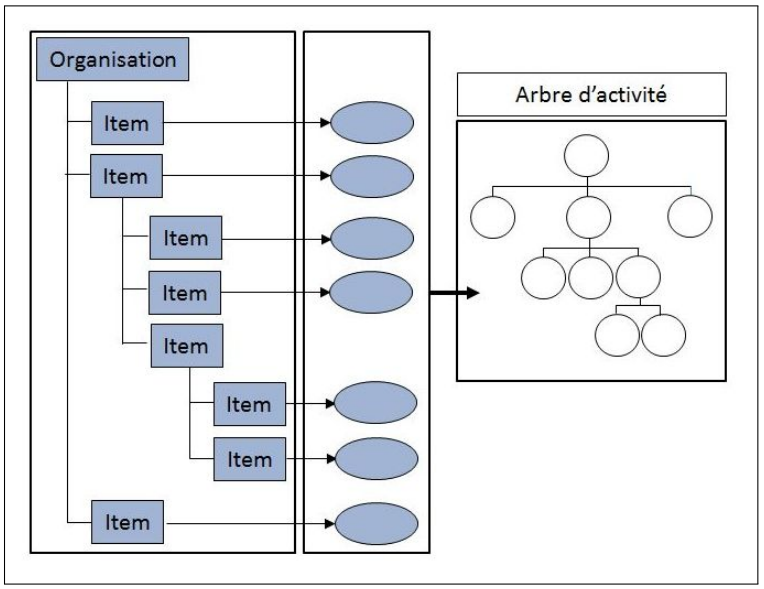

FIGURE 4.6 – Le modèle de séquencement et de navigation SCORM

4.3.5/ TIN CAN

TIN CAN API[17] est une norme récemment finalisée qui est né de la volonté d'ADL de proposer une norme plus souple et adaptée à la multiplicité des objets pédagogiques. TIN CAN API s'inscrit dans un dispositif destiné à remplacer SCORM. Comme SCORM, TIN CAN API sert à suivre des activités de formation et à les transmettre à une plateforme de gestion de formation. Mais à l'inverse de SCORM qui nécessite une plateforme LMS centralisant l'ensemble des contenus de formations, TIN CAN étend la notion d'apprentissage à n'importe quel contenu, où qu'il se trouve et quel que soit sa nature : une vidéo sur youtube, un entretien avec un collègue, la lecture d'un article, un blog, etc. En effet le modèle SCORM impose l'existence des ressources d'apprentissage dans un SGA (Système de Gestion de l'Apprentissage). Cette limitation est dépassée dans TIN CAN qui permet l'agrégation des traces d'apprentissage à partir de plusieurs sources et donc l'utilisation des ressources d'apprentissage existantes sur d'autres serveurs.
TIN CAN ajoute un nouveau terme, un LRS (Learning Record Store), qui définit une sorte de base de données séparé mais lié à une plateforme de formation à distance (LMS). On parle donc de LMS couplé avec un service LRS. Le principe est que chaque apprenant peut créer des entrées personnelles (des ressources d'apprentissage stockées sur des plateformes différentes comme Slideshare[18] ou Vimeo[19]) et les intgrer à un LRS. Ces entrées reconstruisent virtuellement ensuite les activités réalisées par l'apprenant en dehors des activités de formation traditionnelles proposés par le LMS.

17. http ://tincanapi.com/
18. http ://fr.slideshare.net/
19. https ://vimeo.com/

Un de ses inconvénients de TIN CAN est qu'il a la capacité de créer une très grande quantité de données. Mais ces données que valent-elles vraiment ? Comment interpréter ces données et déterminer celles qui sont vraiment utiles en termes de développement de compétences ? Deuxième inconvénient, la mise en œuvre de la norme nécessite de disposer d'un LRS, c'est à dire un conteneur capable de lire un fichier au format TIN CAN et de stocker les interactions avec l'apprenant. Or la plupart des LMS d'aujourd'hui ne sont pas équipés de LRS et utilisent majoritairement la norme SCORM.

4.3.6/ LRMI

L'Initiative LRMI [20] (Learning Resource Metadata Initiative) est d'élaborer un cadre commun de métadonnées pour décrire les ressources d'apprentissage sur le Web. LRMI est pris en charge par les moteurs de recherche Google, Bing, Yahoo et, liée à schema.org [21] ; utilise le Web sémantique, et est pris en charge par AEP [22] (Association of Educational Publishers) et CC [23] (Creative Commons). Il s'agit essentiellement du système de Schema.org étendu aux besoins en éducation.
LRMI vise précisément à introduire des critères utiles aux formateurs et aux apprenants dans des métadonnées qui seront utilisées par les éditeurs de contenus éducatifs. Par exemple, la ressource est-elle destinée aux professeurs ou aux étudiants ? Est-ce un contenu, un devoir, un test ? S'il s'agit d'une séquence de formation, dure t-elle 10 minutes, 30 minutes, 2 semaines ? Est-elle destinée à des adultes ou des enfants ? En fait, il ne s'agit pas de métadonnées telles que nous les connaissons, mais de microdonnées que l'on insère dans les balises HTML de contenu telles , , <a href>, etc. En fait n'importe quelle balise peut accueillir les items selon les spécifications de Schema.org.
L'implémentation de cette norme permet d'améliorer la découverte de contenus d'apprentissage et permet de personnaliser le processus d'apprentissage à l'échelle du Web et par conséquent étendre les outils d'apprentissage actuel dans l'espace du Web. La proposition LRMI est a été officiellement intégrée et adoptée par Schema.org en 2013.

4.3.7/ QUE CHOISIR POUR NOTRE MODÈLE ?

Dans notre modèle nous nous focalisons principalement sur deux points : la description des objets pédagogiques et la modélisation de scénarios pédagogiques personnalisés.
Nous proposons de décrire les contenus pédagogiques proposés par CrossKnowledge en utilisant des informations issues des schémas de description des objets pédagogiques. Une solution peut provenir des initiatives de normalisation qui visent à

20. http ://www.lrmi.net/
21. http ://schema.org/
22. http ://www.aepweb.org/
23. http ://fr.wikipedia.org/wiki/Creative_Commons

établir des règles pour faciliter le partage et la réutilisation des contenus pédagogiques. Afin de respecter les techniques de modélisation déjà utilisées au sein de CrossKnowledge, nous choisissons d'utiliser le schéma LOM. Seulement l'utilisation des métadonnées telle qu'elle est préconisée par les normes ne suffit pas à résoudre les problèmes de réutilisabilité et d'accessibilité. Afin de combler les lacunes de LOM, nous remplaçons l'utilisation de XML par un langage plus expressif, en proposant une ontologie de LOM.
Outre le niveau d'expressivité des ontologies, ces dernières peuvent nous aider à résoudre les problèmes de réutilisabilité et d'accessibilité. Ces problèmes ne seront résolus que si le système et les acteurs partagent le même sens accordé aux métadonnées : les objets pédagogiques sont souvent conçus et développés par des organisations et des formateurs différents, constituant généralement des contenus autonomes mais très hétérogènes au niveau sémantique. Des conflits sémantiques surviennent puisque les systèmes n'utilisent pas la même interprétation de l'information qui est définie différemment d'une organisation à une autre. Les besoins immédiats demandent l'application de modèles de métadonnées en vigueur pour rendre les contenus d'apprentissage réutilisables pour assurer l'interopérabilité sémantique des plateformes hétérogènes. L'utilisation des ontologies est une solution intéressante.

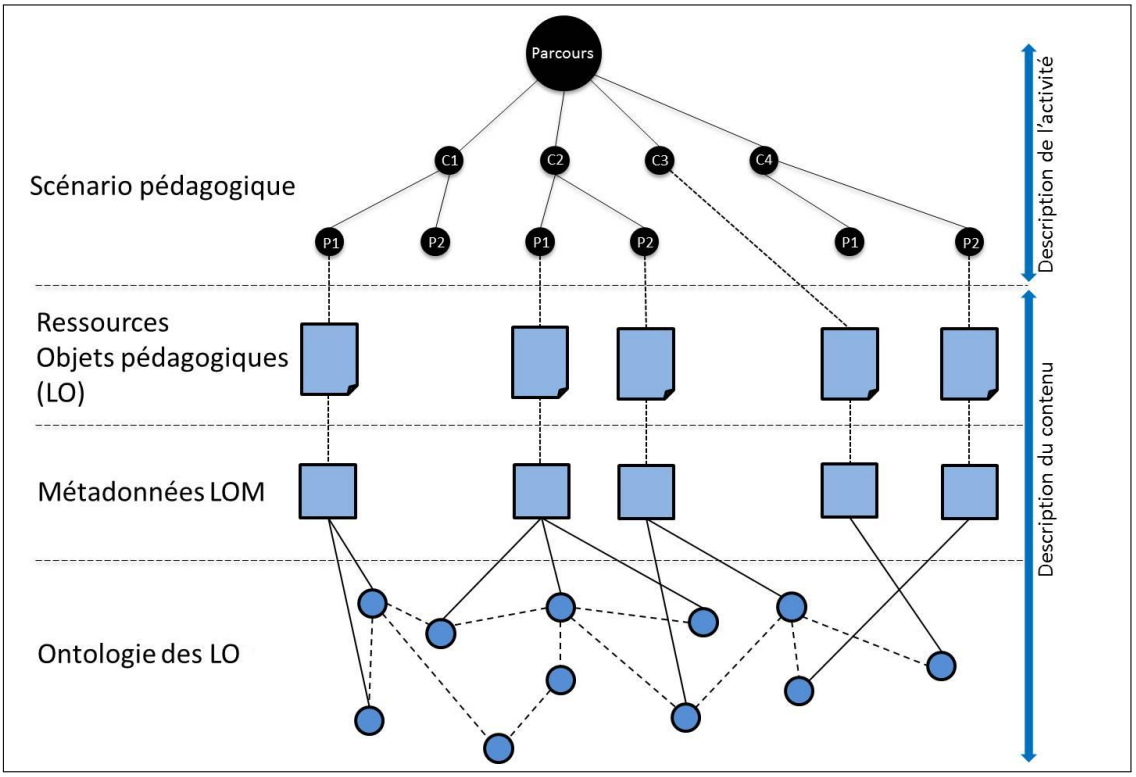

FIGURE 4.7 – Modèle des objets pédagogiques

L'utilisation d'une ontologie permet une meilleure représentation des connaissances. Elle permet à l'ordinateur la production de nouvelles connaissances par le biais d'inférences.

Elle permet aussi à l'homme et à la machine d'accorder des sens communs aux termes utilisés dans un domaine particulier, ici en apprentissage électronique, afin de lever toute ambiguïté pendant les traitements.
La figure 4.7 montre un aperçu du modèle que nous proposons. Nous détaillerons la description de ce modèle dans le chapitre 5.
Nous définissons un scénario pédagogique comme « un ensemble ordonné d'activités, régies par des acteurs qui utilisent des objets pédagogiques ». Dans le modèle IMS-LD un parcours de formation est considéré comme une agrégation d'objets pédagogiques. Dans le modèle SCORM un scénario est formé d'activités imbriquées dans des relations parent/fils. Dans ces deux modèles un scénario est vu uniquement comme l'agrégation d'objets pédagogiques et non pas comme un enchainement d'activités. Afin de remédier à cette lacune et de proposer des scénarios plus élaborés et complexes, nous proposons d'utiliser des relations sémantiquement riches à travers notre modèle ontologique.
Finalement dans tous les standards décrits précédemment le contexte d'utilisation des objets pédagogiques n'est pas pris en compte. Afin d'augmenter la réutilisabilité et afin de personnaliser les scénarios et la livraison des objets pédagogiques aux apprenants, nous proposons d'enrichir notre modèle par une ontologie de contexte et des règles métiers afin de dicter le comportement du système dans un contexte particulier.

4.4/ CONCLUSION

Nous avons consacré ce chapitre à analyser le concept d'objet pédagogique. En nous basons sur les travaux menés dans le domaine des langages de modélisation pédagogiques, nous proposons une nouvelle définition en identifiant deux classes différentes d'objets pédagogiques : des objets pédagogiques qui permettent de structurer et d'organiser un parcours de formation et l'organiser et des objets pédagogiques qui correspondent aux ressources physiques ou numériques nécessaires à la réalisation des activités d'apprentissage. Chaque classe est caractérisée par un ensemble de métadonnées. Aujourd'hui, les organismes de normalisation proposent différents schémas de description des objets pédagogiques par des métadonnées. En analysant les différents modèles tels que le Dublin Core, LOM, IMS-LD et SCORM, nous avons pu identifier un certain nombre de faiblesses de ces modèles.
De notre côté, afin d'obtenir toutes les qualités demandées par un LMS, comme la réutilisabilité des contenus, l'accessibilité, l'interopérabilité et la durabilité, nous prenons en compte les théories de l'éducation ainsi que des contextes d'apprentissage différents. Nous basons notre approche sur des technologies du Web sémantique, dont l'utilisation des ontologies, tout en respectant les normes en apprentissage en vigueur. Dans le chapitre suivant nous décrivons l'architecture de notre système pour le m-learning, en définissons plus clairement notre modèle des objets pédagogiques.

5

ARCHITECTURE DU SYSTÈME

Dans le domaine de l'enseignement à distance, ces dernières années ont été marquées par l'essor du m-learning. Cet essor est lié à deux facteurs : Le premier facteur est le développement continu d'innovations dans le domaine des technologies mobiles et le second facteur réside dans l'extrême degré d'appropriation et d'usage, par tout un chacun, des appareils mobiles connectés à internet. L'apprentissage devient situé, contextuel, et personnel. Ce phénomène entraîne un nouveau paradigme dans le domaine de la formation à distance où l'environnement ne s'adapte plus à l'apprentissage mais où l'apprentissage doit s'adapter dynamiquement à un environnement en constant changement.
L'apparition de ce nouveau paradigme nécessite l'évolution des méthodes de formation. Pour répondre à ce paradigme, nous avons identifié un ensemble d'approches souvent basées sur des systèmes de recommandation. Ces systèmes permettent une adaptation dynamique du contenu, en tenant compte d'un profil de consommation. La conception d'un tel système doit répondre à une contrainte forte : la pertinence de l'information proposée. Plus l'information est pertinente, plus son appropriation par l'utilisateur final sera forte. Dans le domaine du m-learning, la pertinence d'un apprentissage sur une durée d réside dans la pertinence de la combinaison des éléments composant cet apprentissage, malgré les variations du contexte pendant cette durée d. Pour répondre à cet objectif de pertinence, nous définissons dans ce chapitre, de manière formelle, l'architecture d'un nouveau type de système de recommandation sensible au contexte. Ce système se base sur les technologies du Web sémantique et des techniques d'optimisation combinatoire, proposant des combinaisons d'objets pédagogiques (LO) adaptées au contexte de l'apprenant. Cette proposition est une extension des systèmes de recommandation basés sur le contenu, utilisant une ontologie associée à des mécanismes de pondération. L'architecture utilisée est issue des systèmes hypermédia adaptatifs en couches, et est constituée de deux grandes parties : un serveur de connaissances et un modèle d'adaptation. Le serveur de connaissances est formé de trois couches : un modèle des LO, un modèle de contexte et un modèle de contraintes qui regroupe un ensemble de règles

métiers modélisant le savoir-faire du fournisseur des contenus. La deuxième partie du système, le modèle d'adaptation, permet de proposer des recommandations sous forme de combinaisons pertinentes de LO ; le système est constitué de manière à pouvoir utiliser des algorithmes d'optimisation combinatoire adaptés à notre domaine d'application. Nous présentons dans ce chapitre une description détaillée de chaque partie de notre architecture.

Sommaire

5.1 Présentation générale de l'architecture 98
5.2 Serveur sémantique de connaissances 100
5.2.1 Le Web sémantique 101
5.2.2 Modélisation par les ontologies 104
5.2.3 Modèle des LO 105
5.2.4 Modèle de contexte 110
5.2.5 Modèle de contraintes 120
5.3 Modèle d'adaptation 124
5.4 Conclusion 125

5.1/ PRÉSENTATION GÉNÉRALE DE L'ARCHITECTURE

De nouveaux usages apportés dans le domaine de l'apprentissage se sont multipliés sous différentes modalités. Dans le cadre de l'apprentissage au sein des entreprises, nous cherchons à développer un système m-learning dont les principaux enjeux sont : (1) l'apprentissage au travail quels que soient l'heure, le lieu, le dispositif de délivrance, les contraintes technologiques des processus d'apprentissage de manière adaptée au profil de l'apprenant ; (2) l'apprentissage continu au travers des différents contextes. Nous proposons une approche pour un système m-learning contextuel et adaptatif intégrant des stratégies de recommandation de scénarios de formations sans risque de rupture.

Dans l'objectif de développer un tel système m-learning, nous commençons par identifier différents niveaux d'hétérogénéité : (1) hétérogénéité sémantique et (2) hétérogénéité d'usage : D'un côté, en e-learning les ressources sont conçues et développées par des organisations et des formateurs différents, constituant généralement des contenus autonomes mais aussi hétérogènes au niveau sémantique. En effet, des conflits sémantiques surviennent puisque les systèmes n'utilisent pas la même interprétation de l'information qui est définie différemment d'une organisation à l'autre. Les besoins immédiats demandent l'application de standards en vigueur pour rendre les contenus réutilisables et afin d'assurer l'interopérabilité sémantique des plateformes e-learning hétérogènes. D'un autre côté, les apprenants, qui sont les principaux acteurs d'une plateforme d'apprentissage, ont des connaissances et des objectifs différents et se situent dans des

contextes d'apprentissages différents (hétérogénéité d'instant, hétérogénéité de durée, hétérogénéité de support visuel, hétérogénéité de niveau sonore, etc.). Il faut dans ce cas avoir une meilleure connaissance du contexte d'apprentissage et s'interroger efficacement sur les stratégies pédagogiques à mettre en place pour répondre au mieux aux besoins de chaque apprenant.
Cette section est dédiée à la présentation de l'architecture générale de notre système de recommandation pour le m-learning dont la principale caractéristique réside dans sa capacité à construire des parcours de formation selon les contraintes contextuelles de l'apprenant. L'enjeu du système à construire est de répondre à la fois au verrou d'hétérogénéité sémantique (aspect statique du système) et au verrou d'hétérogénéité d'usage (aspect dynamique et adaptatif du système).

Pour la création d'une nouvelle architecture de systèmes recommandations, nous partons du postulat que les contenus d'apprentissage ainsi que le contexte sont modélisés au sein d'une ontologie et par conséquent, la qualité du contenu est avérée. Un système de recommandation basé sur le contenu pourrait être construit à partir de ce postulat. Néanmoins, il ne permettrait pas de représenter le savoir-faire du fournisseur en matière de qualification d'un objet pédagogique et en matière de combinaison pertinente des LO. Il est donc nécessaire de trouver un moyen qui représente ce savoir-faire sous la forme de règles paramétrables. La qualification des offres dans les systèmes de recommandation par filtrage collaboratif est réalisée par rapport à des notes qui sont données par les utilisateurs aux items. Nous choisissons de nous inspirer de cette idée pour permettre de donner à notre architecture des fonctions de pondération des LO par rapport aux règles métiers. Ce système évolutif de règles et de pondération permet de traduire le savoir-faire du fournisseur des contenus, et permet une qualification constante du contenu tout au long de son cycle de vie.

Dans les systèmes adaptatifs, le contenu et les informations relatives aux utilisateurs sont représentés dans la couche couramment appelée *modèle de domaine*. Les travaux sur les systèmes adaptatifs utilisent de plus en plus les technologies du Web sémantique pour représenter ce modèle, notamment en utilisant les outils de représentation ontologiques. Dans notre démarche, nous faisons une distinction entre la modélisation des LO et la modélisation du contexte. Nous divisons ainsi le *modèle de domaine* en deux sous-modèles : le *modèle des LO* et le *modèle de contexte*. Pour la modélisation des règles métiers, nous utilisons le terme *modèle de contraintes*, en accord avec la terminologie proposée par l'architecture en couches LAOS (voir chapitre 3). De même nous gardons le terme *modèle d'adaptation* de l'architecture LOAS pour qualifier la couche comportementale du système.

Nous proposons ainsi une architecture d'un système de recommandation pour le m-learning articulée en deux parties : La première partie est constituée d'*un serveur de connaissances* où les données et les processus métiers sont modélisés par une ontologie évolutive et des règles métier. Cette partie de l'architecture regroupe le modèle

des LO, le modèle de contexte et le modèle de contraintes. La deuxième partie, correspond au *modèle adaptatif*, et est basée sur des algorithmes de métaheuristiques permettant d'analyser les règles métier et l'ontologie pour permettre une bonne combinaison des contenus d'apprentissage. La figure 5.1 présente un aperçu de cette architecture. Coupler les techniques de modélisation sémantique des contenus et du contexte apprenant avec des algorithmes performants issus du domaine de l'optimisation combinatoire, constituent notre système de recommandation pour l'optimisation de l'offre m-learning.

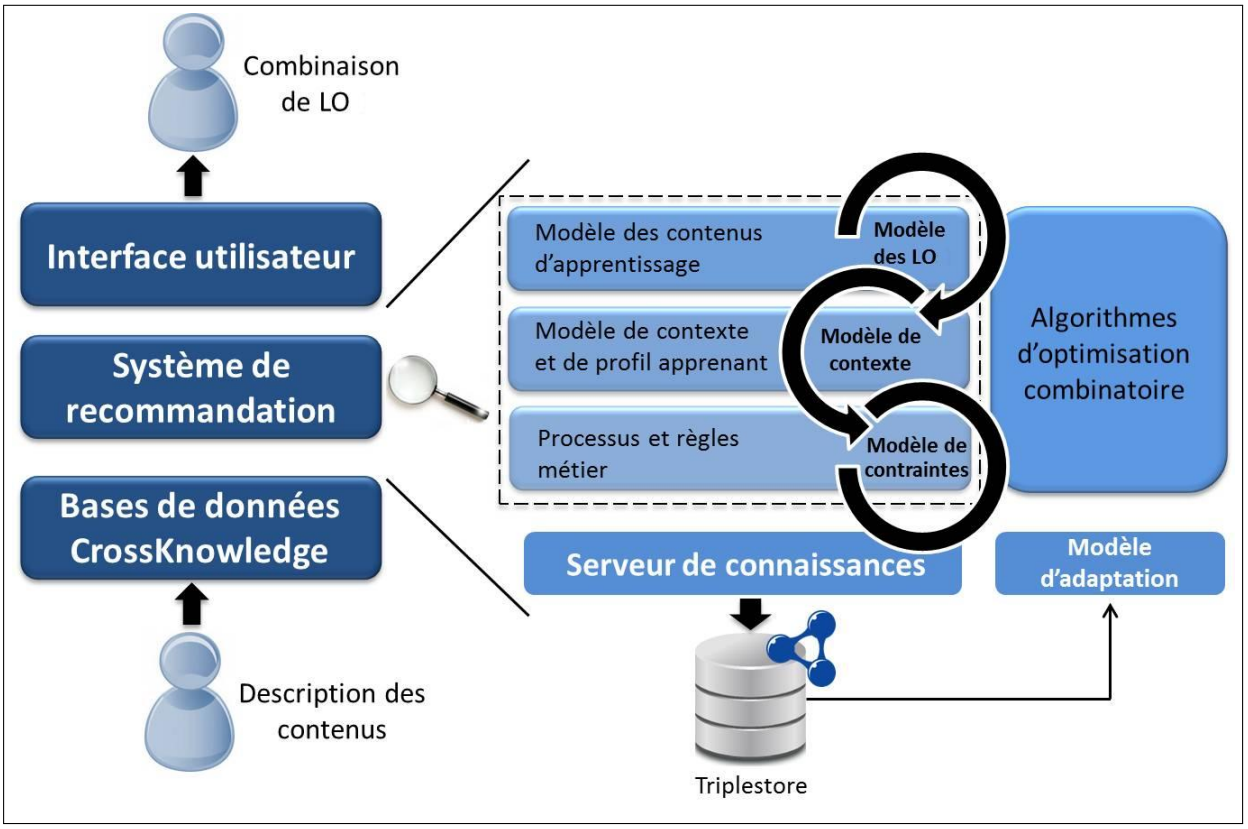

FIGURE 5.1 – Architecture générale du système m-learning

Dans la section suivante nous décrivons la première partie de notre architecture, le serveur sémantique de connaissances.

5.2/ SERVEUR SÉMANTIQUE DE CONNAISSANCES

Dans le serveur sémantique de connaissances nous modélisons les contenus d'apprentissage, le contexte et les connaissances des experts métiers. Pour cette modélisation, l'apport des technologies du Web sémantique est significatif. La figure 5.2 illustre l'architecture générale du serveur sémantique de connaissances.

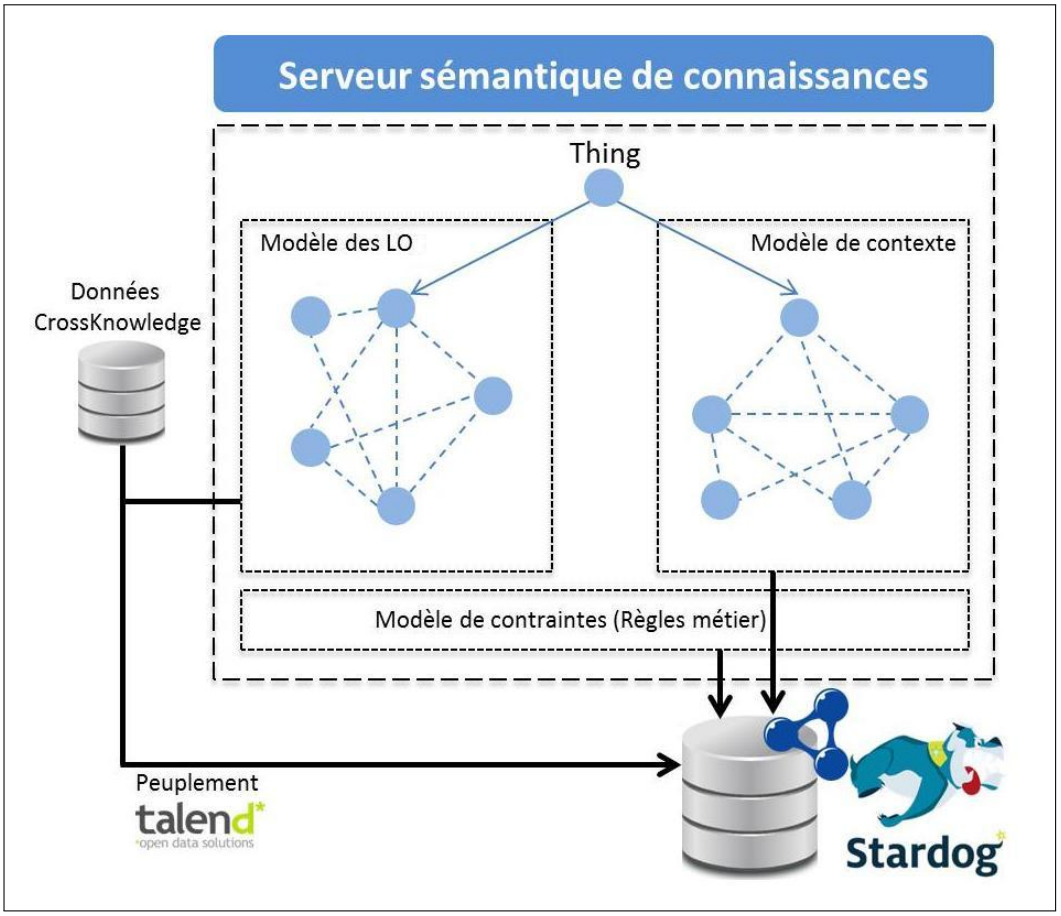

FIGURE 5.2 – Architecture du serveur sémantique de connaissances

Dans cette section nous présentons tout d'abord un bref état de l'art du Web sémantique suivi d'une introduction aux ontologies. Ensuite, nous définissons et décrivons les trois modèles du serveur sémantique de connaissances : Le premier modèle, le modèle des LO, permet de présenter les contenus sous formes d'instanciation d'objets pédagogiques. Le deuxième modèle, le modèle de contexte, permet de présenter le contexte d'apprentissage. Finalement le troisième modèle, le modèle de contraintes, permet de présenter le savoir-faire des experts métiers et les contraintes spécifiques aux contenus suivant les contextes des apprenants.

5.2.1/ LE WEB SÉMANTIQUE

Depuis sa création, Internet a connu de nombreuses évolutions. Dans son état initial le Web 1.0 était exclusivement constitué de pages statiques qui étaient rarement mises à jour car cette tâche était longue et fastidieuse. La génération dynamique de pages à partir de bases de données, la facilité à communiquer et le développement des réseaux sociaux sont autant d'éléments qualifiant le terme Web 2.0. Malgré cette évolution, un inconvénient du Web 1.0 est cependant toujours présent : les données présentées sous

forme de pages par Internet n'ont de sens que pour les yeux d'un humain. Le code des pages Internet est pris en charge par le navigateur qui a pour rôle de traduire et d'afficher les données à l'écran.

En 1994, la création du W3C fut annoncé à la conférence WWW à Genève et c'est à cette période que Tim Berners-Lee a dressé les objectifs du W3C et a montré les besoins d'ajouter de la sémantique au Web futur. Il montre qu'avec la forte croissance du Web, il est de plus en plus difficile pour les humains de rechercher et de traiter des données présentes sur ce réseau. Le Web sémantique place la capacité de traitement et de calcul de plus en plus grande des machines face à l'immensité du Web en donnant les moyens nécessaires aux machines pour traiter les données. Cela permet aux utilisateurs d'accéder plus facilement et de manière plus intuitive aux données par le biais de services utilisant les technologies mises à disposition par le Web sémantique. Tim Berners-Lee définit le Web sémantique comme un ensemble de technologies ayant pour objectif principal de rendre accessible aux machines la signification (la sémantique) des données présentes sur le Web (Berners-Lee et al., 2001).

Le Web sémantique ouvre de nombreuses perspectives parmi lesquelles on trouve :
- *L'optimisation de la coopération entre les machines (l'interopérabilité) :* Le Web sémantique permet aux machines de comprendre la signification des informations par le biais de langages sémantiques définissant les données. L'utilisation d'un langage commun permet de renforcer l'interopérabilité entre les machines.
- *Le raisonnement par inférence :* C'est une opération qui consiste à tirer une conclusion à partir de données connues. Cela permet donc de déduire de nouvelles informations à partir de l'existant. C'est un processus de création qui permet l'expansion des connaissances. Tirer des conclusions, faire des déductions sont des choses qui étaient jusque-là réservées aux êtres dotés d'un esprit logique. Grâce à la sémantique, il est possible de doter les machines de ce mécanisme (voir annexe A pour plus de détails sur les techniques de raisonnement).

Après plusieurs années de travail, le W3C débute l'émission de recommandations concernant le Web sémantique en 1997. Parmi ces recommandations se trouve notamment une recommandation concernant le langage RDF (Resource Description Framework), un standard du Web sémantique. Par la suite, de nombreuses autres recommandations ont été émises par le W3C : RDFS (RDF Schema) et OWL (Web Ontology Language) en 2004 ou encore SPARQL en 2008 (annexe A).
En 1998, Tim Berners-Lee publie un documen (Berners-Lee, 1998) sur ce qui sera plus tard appelé le Web sémantique. Dans ce document, il présente le Web sémantique comme une sorte d'extension du Web des documents, qui constitue une base de données à l'échelle mondiale, afin que toutes les machines puissent mieux lier les données du Web. Dans un autre document (Berners-Lee, 2000), il propose une représentation graphique, le *layer cake*[1], qui montre l'agencement des différentes briques technologiques

1. http ://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

du Web sémantique.
Dans le layer cake illustré par la figure 5.3, le Web sémantique est représenté par un ensemble de technologies et de concepts organisés sous la forme d'une architecture en couches :

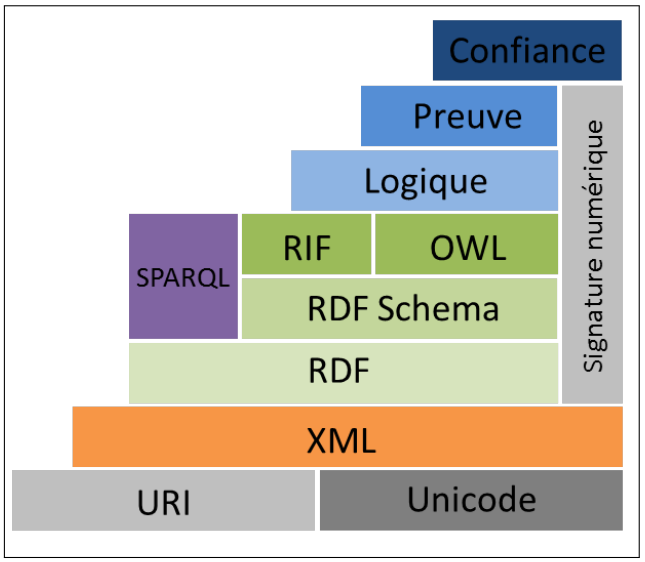

FIGURE 5.3 – Architecture en couches des standards du Web sémantique

– *La couche URI (Universal Resource Identifier) :* Cette couche a pour objectif l'identification des ressources (pas nécessairement sur le Web).
– *La couche XML :* offre un métalangage permettant de représenter de manière structurée et personnalisée des informations (en utilisant les balises de son choix). Cette couche contient, en plus du langage XML, des technologies comme les DTD (Document Type Definition), le langage XML Schema ou encore le langage de mise en forme XSL (Extensible Stylesheet Language).
– *La couche RDF :* est le premier élément de l'architecture apportant une réelle plus-value sémantique. RDF est un langage offrant la possibilité de décrire des ressources Web par l'utilisation de triplets <sujet, prédicat, objet>.
– *Les couches RIF (Rule Interchange Format), RDF Schema et Ontologies OWL :* visent à développer les capacités d'expression de la couche RDF en offrant notamment la possibilité de hiérarchiser des concepts ou de définir des restrictions sur des propriétés.
– *La couche SPARQL :* représente un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF.
– *Les couches logique et preuve :* ont respectivement pour objectif de permettre l'utilisation de mécanismes d'inférence (déduction de connaissances) et de prouver la véracité des connaissances ainsi produites.
– *La couche confiance :* constitue l'interface entre le Web sémantique et l'utilisateur. L'utilisateur qui travaille avec des programmes sémantiques qui réalisent certaines opérations à sa place doit avoir confiance dans les résultats qui lui sont proposés.

La couche confiance a donc pour but d'expliquer ses raisonnements et de garantir la fiabilité de ces derniers en utilisant les fonctionnalités des deux couches précédentes que sont la couche preuve et la couche logique.

D'après la figure 5.3, le Web sémantique passe par la création d'un ensemble de langages de modélisation des connaissances manipulables par les machines. En effet, ces langages travaillent principalement sur des données très formalisées et structurées et ont besoin d'instructions pour savoir comment manipuler ces données. Pour cette raison, le Web sémantique se propose de représenter les connaissances sous forme de structures utilisant la logique de description, à savoir *les ontologies*, que nous présentons dans la section suivantes.

5.2.2/ MODÉLISATION PAR LES ONTOLOGIES

L'ontologie est un concept central du Web sémantique. L'ontologie est à l'origine un domaine philosophique qui peut être décrit comme l'étude de l'être, de ses modalités et de ses propriétés. Platon fut un des premiers à décrire le concept d'ontologies en introduisant notamment une différence entre le monde des objets (ou le monde réel) et le monde des idées. Par la suite, Aristote, un des étudiants de Platon, a posé les bases des ontologies en introduisant des notions comme les catégories (concept) et la subsumption (relation hiérarchique entre concepts). Aristote peut être considéré comme le créateur de la notion de taxonomie. Aristote a également proposé plusieurs règles d'inférence, toujours utilisées aujourd'hui dans de nombreux domaines, comme les syllogismes, le modus ponens et le modus tollens.

La définition d'une ontologie au sens informatique, puise ses ressources dans les domaines de la représentation des connaissances et des réseaux sémantiques :
Dans le domaine de la représentation de la connaissance, les chercheurs tentent de trouver un moyen de représenter diverses connaissances du monde de façon à ce qu'elles soient utilisables par des ordinateurs et afin de pouvoir effectuer des raisonnements dessus. Ces connaissances sont exprimées sous forme de symboles auxquels est donné un sens ou une sémantique.
Dans le domaine des réseaux sémantiques, l'idée d'utiliser ces réseaux est attribuée à (Quillian, 1968). L'objectif central des réseaux sémantiques est de décrire la réalité sous forme de graphes (réseaux) composés de nœuds qui représentent les concepts, reliés par des arcs exprimant les relations entre les concepts. Les relations les plus spécifiques de ce type de réseau sont les liens *sorte_de*. D'autres relations sont utilisées comme *à_pour_partie*, *est_composée_de*, etc. La relation *sorte_de* permet de définir l'une des notions les plus importantes des réseaux sémantiques et de tout autre formalisme intégrant un minimum de logique, qui est la déduction par héritage des propriétés. L'héritage est essentiellement basé sur la transitivité de la relation *sorte_de*. De cette notion d'héritage se dégage le fait qu'il est plus économique d'attacher une caractéristique commune à un

ensemble de concepts au niveau le plus haut de la hiérarchie. De cette manière, il n'est pas besoin de stocker en mémoire pour chaque concept l'ensemble des propriétés, et cela augmente le pouvoir d'expression de recherche d'information mais également des commodités de modélisation : Les propriétés qui lui sont propres suffisent, les autres peuvent être déduites.

Suite à ces travaux, est apparu une des définitions de l'ontologie qui fait autorité dans le domaine, celle de Gruber : « *une ontologie est une spécification explicite d'une conceptualisation* » (Gruber, 1993). Le caractère formel d'une ontologie implique que celle-ci doit être exploitable par des machines tandis que le caractère partagé de sa conceptualisation indique que les créateurs de l'ontologie doivent avoir une même vision du domaine représenté. Une ontologie est ainsi vue comme une structure de données composée de concepts et de relations entre ces dernières et représentant les connaissances d'un domaine.

Pour résumer, les ontologies permettent de standardiser le vocabulaire, d'uniformiser le langage d'échange entre les différents acteurs, de comparer les différents systèmes et de structurer la connaissance pour simplifier l'analyse et la synthèse des connaissances d'un domaine. S'appuyant sur les ontologies, le développement du Web sémantique ouvre de nouvelles possibilités et des défis à la conception d'une génération de systèmes adaptatifs, rendant possible la modélisation du profil et du contexte des utilisateurs (Razmerita, 2003). Dans le cadre de l'adaptation de parcours de formation à l'apprenant, l'ontologie est devenue une solution incontournable. Elle permet la construction de modèles de connaissance complexes qu'il est possible d'utiliser pour modéliser à la fois les utilisateurs du système, leur contexte et le domaine d'application de manière intelligible.
Dans la couche serveur sémantique de connaissances de notre proposition, et à des fins de raisonnement, une seule ontologie est considérée que nous appelons ontologie de domaine du m-learning. Cependant, nous distinguons deux sous-parties de cette ontologie : (1) le modèle des LO et (2) le modèle de contexte. Dans la section suivante nous commençons par décrire le modèle des LO.

5.2.3/ MODÈLE DES LO

Un contenu d'apprentissage est une instanciation d'objets pédagogiques, ou LO. L'idée fondamentale derrière la création des LO est la possibilité de construire un parcours de formation autour de composants de petite taille qui peuvent être réutilisés plusieurs fois dans différents contextes d'apprentissage. Seulement, ces LO sont souvent conçus et développés par des organisations et des auteurs différents constituant des contenus d'apprentissage autonomes et sémantiquement hétérogènes. Ils sont ainsi difficilement réutilisables car ils n'ont pas été conçus à cet effet. Il est alors indispensable de penser à une modélisation partagée des LO en vue de les rendre facilement accessibles, exploitables, réutilisables et sémantiquement interopérables.

À l'instar de tout système d'information requérant un mécanisme de description formelle de ses ressources par des éléments de métadonnées, les systèmes d'apprentissage se démarquent aussi par la conception et la mise en place de leurs propres modèles de description des LO. Les métadonnées sont devenues l'un des mécanismes essentiels pour décrire, référencer et localiser des ressources dans le système. Comme expliqué dans le chapitre 4, nous proposons pour modéliser les LO d'utiliser des informations issues des schémas de description des ressources pédagogiques par des métadonnées. Dans notre cas nous nous intéressons plus particulièrement au standard LOM.

Binding de LOM :
LOM ou Métadonnées pour les Objets Pédagogiques, permet de définir la structure d'une instance de métadonnée pour un LO. Les descripteurs de LOM peuvent être utilisés dans la conception des systèmes d'apprentissage pour l'indexation des LO. Pour cela ils doivent être implémentés dans un langage structuré.
La représentation du modèle abstrait dans un format spécifique est appelé binding. Aujourd'hui il existe deux types de binding du schéma LOM, soit du binding XML, soit du binding RDF (Ghebghoub et al., 2009) :
Le binding XML est facile à implémenter, cependant il reste insuffisant pour la représentation de tous les éléments de LOM puisqu'il ne permet pas d'exprimer la sémantique de ses éléments. Quant au binding RDF il définit un ensemble de constructions RDF qui facilitent l'introduction des métadonnées de LOM dans le web, et il est complété par RDFS pour la définition des classes, des propriétés, etc. L'avantage de ce deuxième type de binding est qu'il rajoute de la sémantique aux éléments de LOM, sauf qu'il n'est pas assez expressif pour définir toutes les contraintes de LOM. Prenons l'exemple des éléments *Title* et *Entry* de la catégorie *General* (voir chapitre 4 pour avoir une description des éléments de LOM) qui sont des éléments obligatoires dans LOM. En utilisant RDF et RDFS on ne peut préciser qu'une propriété est obligatoire ou contraindre son utilisation à une seule fois pour une ressource (Bourda, 2002). Comme deuxième exemple, RDF et RDFS ne permettent pas d'exprimer l'inverse d'une relation : ainsi, dire qu'un *LO x has_part LO y*, ne permettra pas d'induire que le *LO y is_part_of LO x*.
Qu'elles soient implémentées en XML ou en RDF et RDFS, l'utilisation des métadonnées de LOM pose un certain nombre de problèmes. Le manque d'expressivité nous pousse à utiliser un autre formalisme plus puissant. Utiliser une ontologie de LOM pour indexer les ressources pédagogiques permet une meilleure compréhension des éléments et des valeurs proposées et en conséquence cela facilite leurs descriptions.

Afin de déterminer quel langage est le plus approprié pour résoudre le problème d'expressivité, nous nous sommes penchés sur l'identification de la logique de description $\mathcal{LD}$ nécessaire pour la construction de notre modèle (voir annexe A pour avoir une description des familles de $\mathcal{LD}$). $\mathcal{LD}$ est une famille de formalismes pour représenter les connaissances d'un domaine d'une façon structurée et formelle. Une caractéristique fon-

damentale de ces langages est leur sémantique descriptive formelle.
Nous partons d'une logique minimale $\mathcal{ALC}$ et nous rajoutons à cette logique les constructeurs nécessaires afin de définir toutes les contraintes de LOM. Nous avons ainsi besoin des constructeurs suivants :

- *Nominal* $\mathcal{O}$ *:* Exemple, l'élément *Statues* de la catégorie *Lifecycle* doit absolument avoir l'une de ces valeurs {*draft, final, revised, unavailable*}.
- *Fonctionnalité* $\mathcal{F}$ *:* Exemple, pour restreindre l'utilisation de l'élément *Title* de la catégorie *General* à une seule fois.
- *Restriction de nombre qualifié* $\mathcal{Q}$ *:* Exemple, pour fixer une cardinalité minimale de 1 pour l'utilisation de l'élément *Keyword* de la catégorie *General*.
- *Hiérarchie des relations* $\mathcal{H}$ *:* Exemple, l'élément *Type* de la catégorie *Relation* est traduit pas la relation *has_type_relation* et ses sous-relations *is_part_of*, *is_version_of*, etc.
- *Transitivité des relations* $\mathcal{R}^+$ *:* Exemple, si *LO x is_part_of LO y* et *LO y is_part_of LO z* alors *LO x is_part_of LO z*.
- *Relations inverses* $\mathcal{I}$ *:* Exemple, pour indiquer que la relation *is_part_of* est l'inverse de la relation *has_part*.
- *Conjonction de rôles* $\mathcal{R}$ *:* Exemple, pour indiquer que la relation *has_learning_time_laps* est issue de la conjonction des deux relations *has_learning_time_beginning* et *has_learning_time_end*.

De cette liste de constructeurs, nous pouvons conclure que la $\mathcal{LD}$ minimale et nécessaire à la construction de notre modèle des LO correspond à la famille de description $\mathcal{SROIQ}$. Cette $\mathcal{LD}$ est équivalente au langage ontologique OWL 2. L'utilisation d'OWL 2 pour modéliser LOM permettra aux humains de mieux renseigner les éléments de la norme afin d'avoir une meilleure collaboration homme/machine et de mieux décrire les relations reliant les LO entre eux. L'ontologie OWL 2 nous permet de aussi de profiter d'un système d'inférences plus puissant, ainsi que d'une vérification de consistance de l'ontologie plus poussée.

Construction du modèle des LO :
Maintenant que nous avons identifié le degré de complexité que doit avoir notre ontologie, nous devons la construire. Pour cela, nous avons étudié la structure LOM et les structures XML des différents éditeurs de LOM. Nous avons en particulier travaillé avec LOMPad [2]. Nous nous sommes aussi intéressé aux travaux de recherche de (Ghebghoub et al., 2009). Ces derniers proposent un ensemble de règles pour transformer le schéma LOM en une ontologie. Nous nous sommes inspirés de ces règles pour la création de notre modèle de LO.

Nous présentons ci-dessous ces règles :

- *R1 :* Un LO et l'ensemble des catégories de LOM sont représentés respectivement par les concepts *Learning_Object* et *Learning_Metadata*.

2. http ://helios.licef.ca :8080/LomPad/en/index.htm

- *R2 :* Chaque catégorie de LOM est représentée par un sous-concept de *Learning_Metadata*. Exemple la catégorie *General* est représentée par *General_Metadata*, un sous-concept de *Learning_Metadata*.
- *R3 :* Le concept représentant un LO est relié aux sous-concepts représentant les différentes catégories de LOM par des relations sémantiques.
- *R4 :* Les éléments de LOM sont à leur tour représentés soit par un concept, un attribut ou une relation selon leur type. En effet, on énumère trois types éléments :
  - *Les éléments simples (ou atomiques) :* Ce sont des éléments avec des valeurs prédéfinies. Exemple, l'élément *Structure* de la catégorie *General* ou l'élément *Semantic density* de la catégorie *Educational* sont des informations prédéfinies/subjectives.
  - Les éléments simples : Ce sont des éléments sans valeur prédéfinie. Exemple, l'élément *Description* ou l'élément *Language* de la catégorie *General* sont des informations non prédéfinies/objectives.
  - Les éléments composées : Exemple, l'élément *Resource* de la catégorie *Relation* ou l'élément *Contribute* de la catégorie *Life Cycle* sont des informations complexes. La description de ces éléments nécessite l'utilisation d'autres éléments de LOM.

Les règles R5, R6 et R7 concernent respectivement les éléments simples sans valeurs, les éléments simples avec valeurs prédéfinies et les éléments composés :

- *R5 :* Si un élément *e* est un élément simple avec des valeurs prédéfinies, alors *e* est représenté par un concept et ses valeurs sont représentées par des individus. Exemple, l'élément *Structure* est représenté par le concept *Structure* et l'ensemble de ses valeurs {*Atomic, Collection, Networked, Hierarchical, Linear*} par des individus.
- *R6 :* Si un élément *e* est un élément simple sans valeur prédéfinie, alors *e* est représenté par un concept associé à la catégorie LOM le représentant. Exemple, les éléments *Title* et *Description* de la catégorie *General* sont représentés par deux sous-concepts de *General_Metadata*.
- *R7 :* Si un élément *e* est un élément composé, alors *e* est représenté par un concept. Exemple, pour l'élément *Identifier* qui apparait dans les catégories *General*, *Relation* et *Meta-metadata*, sémantiquement parlant l'identifiant ne peut être considéré comme sous-concept de chacune de ces catégories comme dans le XML binding. Dans ce cas l'identifiant est représenté par un concept nommé *Identifier*, et des relations sémantiques sont alors créées pour relier les trois catégories *General*, *Relation* et *Meta-metadata* à ce concept.

Deux dernières règles sont rajoutées :

- *R8 :* Pour tout concept de LOM est associée une relation permettant de le relier avec le concept *Learning_Object*. La relation a pour domaine le concept *Learning_Object* et pour co-domaine le concept représentant l'élément. Par exemple, les éléments *Title* et *Description* de la catégorie *General* sont reliés au concept *Learning_Object* respectivement par les deux relations *has_lo_title* et *has_lo_description*.

– *R9 :* Un nouvelle catégorie (hors celles de LOM) est rajoutée au modèle des LO. Cette catégorie regroupe des métadonnées spécifiques au catalogue de CrossKnowledge. Dans cette catégorie on peut retrouver des éléments tels que *Theme*, *URL*, *Short_Title*, *Publisher*, etc. La représentation de cette catégorie respecte les mêmes règles décrites précédemment.

Puisque notre approche est motivée aussi par le fait que les LO sont en relation avec d'autres objets pédagogiques, nous mettons en avant l'élément *Kind* de la catégorie *Relation* et l'élément *Structure* de la catégorie *General* qui vont nous permettre de décrire les relations sémantiques entre les LO. Nous décrivons ici certaines de ces relations :

– Dans la catégorie *Relation* de LOM et tout particulièrement pour l'élément *Kind* (type de la relation), on remarque les valeurs suivantes *is_format_of*, *requiert*, *is_part_of*, etc. Nous avons utilisé les valeurs proposées pour cet élément afin de créer des relations permettant de relier un LO à un autre.
  Prenons l'exemple d'un parcours de formations composé d'un cours sur les bases de données sous forme de site Web. Ce cours est constitué de plusieurs pages Web, chacune représentant le contenu d'un chapitre. On suppose un premier chapitre intitulé introduction aux bases de données et aux SGBD, suivi d'un deuxième chapitre consacré aux modèles relationnels puis d'un autre chapitre sur le langage SQL. Le premier chapitre contient une figure illustrant l'architecture des SGBD, cette figure est considérée elle aussi comme un LO. Dans ce cas la relation *is_part_of* va permettre de relier la figure au chapitre 1, ainsi que les trois chapitres à l'ensemble du cours.
– L'élément *Structure* de la catégorie *General* quant à lui nous permet de créer de nouvelles relations. Si cet élément prend la valeur *Linear* cela veut dire que le LO est lui-même un ensemble de LO qui sont parfaitement ordonnes. On déduit alors que ce LO est composé d'autres LO qui peuvent avoir une relation *has_previous* et *has_next*.
  Si on reprend l'exemple précédent du cours sur les bases de données et que l'on considère que la structure du cours est de type linéaire, il est intéressant alors de pouvoir dire que le chapitre 1 a pour suivant (*has_next*) le chapitre 2 qui a pour suivant le chapitre 3. Inversement on peut utiliser la relation *has_previous* entre les trois chapitres.
– La valeur *Hierarchical* de l'élément *Structure* définit un LO comme étant un ensemble de ressources dont les relations peuvent être représentées par une structure hiérarchique, ce qui nous permet d'utiliser des relations de type père et fils entre les LO qui constituent un parcours de formation.

La figure 5.4 illustre un aperçu simplifié du modèle des LO.
Cette modélisation partagée basée sur un squelette ontologique est complétée par une description du contexte mobile de l'apprenant (localisation spatiale, localisation temporelle, description de son profil et du support d'apprentissage). Nous décrivons dans la prochaine section, la deuxième partie de notre ontologie, à savoir le modèle de contexte.

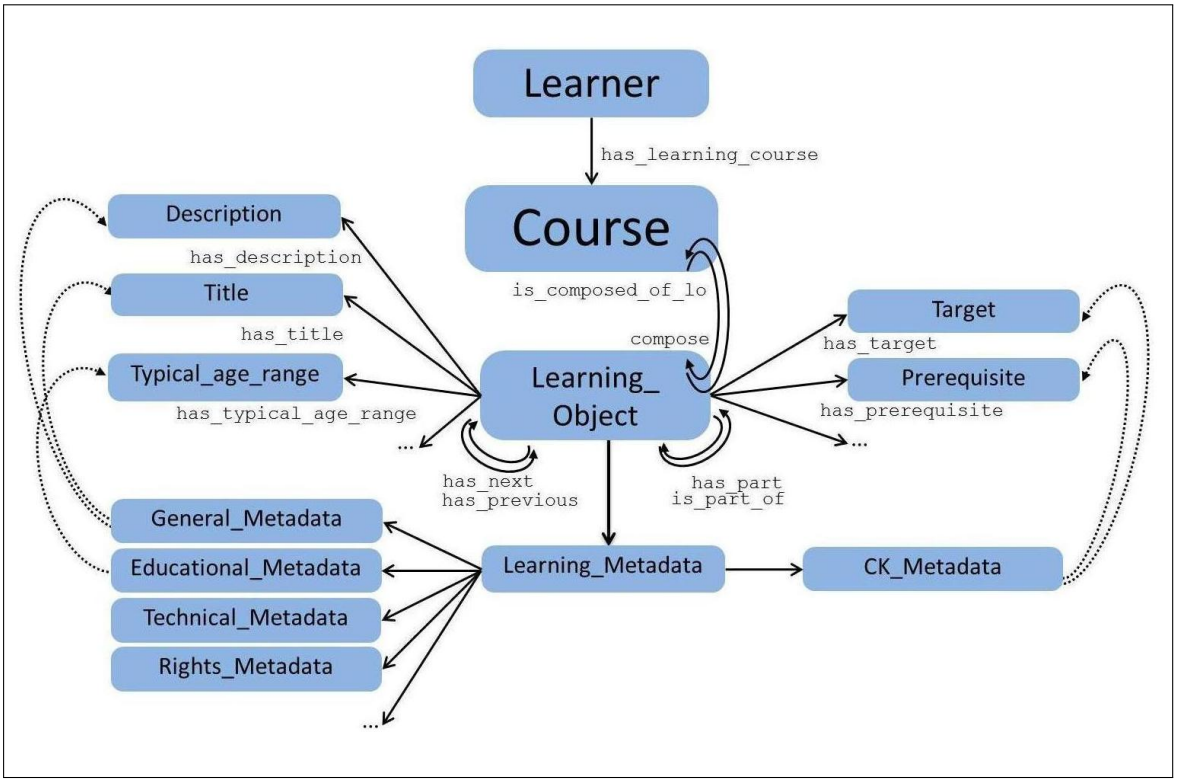

FIGURE 5.4 – Aperçu simplifié du modèle des LO

5.2.4/ MODÈLE DE CONTEXTE

Le contexte d'apprentissage est un aspect crucial dans le m-learning. Il faut donc déterminer selon le contexte actuel de l'apprenant quelles sont les ressources à envoyer, de quelle manière, à quel moment, sur quelle interface, etc. Tout le processus d'apprentissage doit s'adapter à ces changements de contexte. Pour prendre en compte le contexte dans un système m-learning, il est nécessaire de trouver un moyen de représenter le contexte dans ce dernier. Cette représentation doit fournir un cadre cohérent pour mémoriser et traiter les informations du contexte pour réagir aux changements de l'environnement. Il en résulte alors le modèle de contexte.

Il existe plusieurs méthodes de représentation du modèle contextuel : modèle à base de paires attribut/valeur, modèle à base de mots clés, modèle à base de schéma XML, modèle graphique UML, etc. Comme discuté dans le chapitre 2, aucun de ces modèles n'assure l'interopérabilité des données au niveau sémantique. De plus une représentation du contexte doit permettre d'effectuer des raisonnements en vue d'une adaptation. Nous soutenons qu'une modélisation à base d'un squelette ontologique est plus appropriée pour définir le modèle de contexte.
Pour avoir une meilleure visibilité et une meilleure compréhension de cet environnement, nous proposons d'organiser les données qui constituent le modèle de contexte en différentes parties, suivant l'organisation en dimensions présentée dans le chapitre 2.

Modélisation de la localisation spatiale :
Nous considérons que la localisation spatiale de l'apprenant mobile est un facteur d'une grande importance afin de lui proposer des contenus adaptés à l'environnement dans lequel il se trouve.

Il existe différentes façons plausibles pour caractériser la localisation d'un utilisateur :
- *Le nom de la localisation spatiale, Location_Name :* Ce concept réfère au nom d'une localisation. Exemple, le Louvre.
- *Le type de la localisation spatiale, Location_Type :* Ce concept réfère au type d'une localisation. Exemple, le Louvre est de type musée.
- *Les propriétés de la localisation spatiale, Location_Properties :* Ce concept réfère aux caractéristiques de l'environnement autour de l'utilisateur. Dans certaines situations, il est nécessaire de mesurer les caractéristiques de l'environnement autour de l'utilisateur (environnement calme, bruyant, etc.) afin de régulariser le fonctionnement du système. Par exemple, le système peut mesurer le niveau de bruit pour ajuster le niveau sonore des haut-parleurs d'un Smartphone.
  Dobson propose aussi de décrire une localisation spatiale par ces caractéristiques : publique, privée, en mouvement, ou fixe (Dobson, 2005).
- *Les coordonnées géographiques, Geographic_Coordinates (Latitude, Longitude, Elevation) :* La forme la plus simple de localisation spatiale se réfère aux coordonnées géographiques. En exploitant la modélisation sémantique par les concepts précédemment décrits, il est possible de caractériser une localisation à la fois en termes de coordonnées géographiques et sa représentation sémantique. Exemple, une base de données spatiale telle que la base de Geonames Servers [3] permet d'associer à des coordonnées géographiques un nom, une adresse et un type. Il est alors possible par exemple, de savoir qu'aux coordonnées «48.861073°, 2.335784°» est situé le Louvre de type musée et ainsi renvoyer sur le mobile de l'utilisateur de la documentation concernant le musée du Louvre.
  Par coordonnées géographiques d'un lieu, on entend la latitude, la longitude et l'élévation.
- *Relative à (à côté de, à gauche de, etc.) :* Les concepts précédemment décrits peuvent être reliées par diverses relations sémantiques spatiales comme à *is_next_to*, *is_on_left_of*, etc., qui permettent d'inférer des liens entre les localisations. De cette manière, il est possible par exemple de dire, que la Tour Eiffel est à côté du Champ de Mars, et que les deux localisations se trouvent à Paris.

Nous proposons de représenter la dimension spatiale par le concept *Spatial_Specifications*. La figure 5.5 illustre un aperçu simplifié de la dimension spatiale (la figure ne montre que les relations hiérarchiques entre les concepts, et pas les relations sémantiques).

3. http ://www.geonames.org

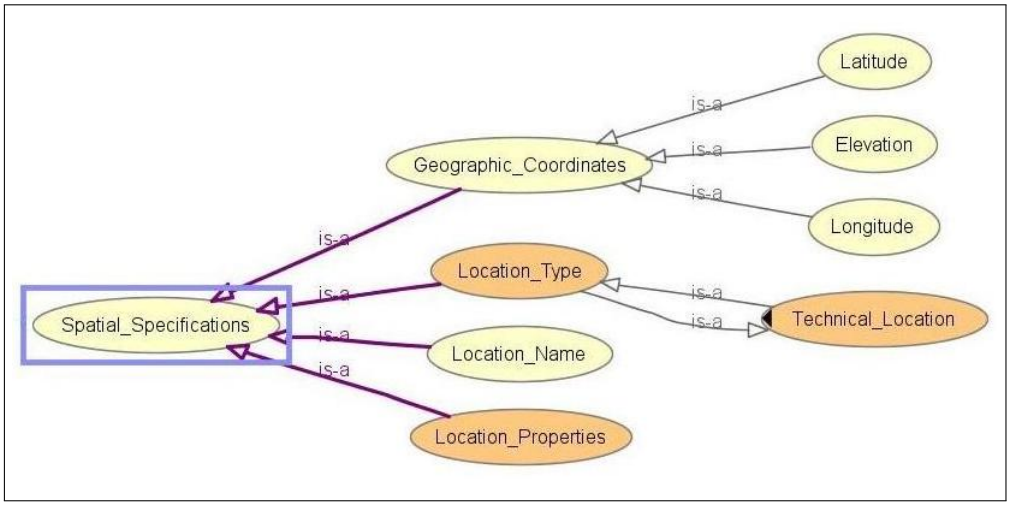

FIGURE 5.5 – Aperçu simplifié des concepts de la dimension spatiale

Modélisation de la localisation temporelle :
Nous considérons que le contexte temporel a une influence sur l'activité de recherche de l'apprenant mobile, aussi bien pour la localisation temporelle de l'apprenant que pour la description des propriétés temporelles des objets pédagogiques.
Une norme internationale (ISO 8601) [4] a été définie par l'Organisation Internationale de Normalisation pour la description des instants et des intervalles de temps, avec l'objectif d'éviter tout risque de confusion dans les communications internationales en raison du grand nombre de notations nationales différentes. L'ontologie OWL-Time [5] est aujourd'hui une référence pour la représentation du temps. Une définition complète d'OWL-Time est proposée par (Pan, 2007). Nous proposons de nous inspirer de cette ontologie pour décrire notre dimension temporelle.

Sous le concept *Temporal_Specifications*, nous proposons premièrement d'organiser la dimension temporelle en deux sous-catégories : *Abstract_Time* et *Concrete_Time*.
La catégorie *Abstract_Time* regroupe principalement les concepts suivants :
– *Le temps de la journée, Time_Of_Day :* Ce concept permet de représenter les différents moments de la journée. Exemple, si nous étudions la routine quotidienne des utilisateurs, on peut distinguer principalement cinq périodes, à savoir matin, midi, après-midi, soir et nuit.
– *Le jour de la semaine, Day_Of_Week :* Ce concept permet de différencier les jours de la semaine entre les jours de travail (lundi, mardi, mercredi, jeudi, vendredi) et les jours de repos (généralement samedi et dimanche).
– *Le mois de l'année, Month_Of_Year :* Ce concept permet d'identifier les différents mois de l'année.
– *Les périodes de vacances, Holiday :* Ce concept permet de déterminer les périodes de

4. http ://www.iso.org/iso/catalogue_detail ?csnumber=40874
5. http ://www.w3.org/TR/owl-time/

vacances ainsi que les jours fériés.
– *Les saisons, Season :* Ce concept permet d'identifier les différentes saisons de l'année.

Dans la catégorie *Concrete_Time* nous retrouvons les concepts suivants :
– *La date, Date :* La notation internationale pour une date est YYYY-MM-DD, où YYYY représente l'année définie par rapport au calendrier grégorien, MM représente le mois de l'année, ses valeurs étant entre 01 (janvier) et 12 (décembre) et DD représente le jour du mois allant de 01 à 31.
  Afin de modéliser des données temporelles en OWL dans un contexte de calendrier, nous pouvons exploiter les types dédiés mis à disposition, soit : *xsd :gYear*, *xsd :gMonth*, *xsd :gDay*. Ces types sont définis par le schéma XML et repris par OWL. Ils peuvent être utilisés pour modéliser des données temporelles telles que la date de naissance d'un apprenant, la date à la quelle il peut accéder à un contenu, etc.
– *L'heure de la journée, Date_Time :* Ce concept permet de décrire une heure spécifiée d'un jour spécifié.
  Le type dédié mis à disposition *xsd :dateTime*, permet de combiner la modélisation date et heure. Ce type de modélisation temporelle est utilisé par exemple pour sauvegarder la date de création d'un LO.
– *L'heure, Clock :* Dans un contexte d'horloge, la notation internationale débute par la lettre T (pour Time), suivie des éléments utiles, dans cet ordre : heures (de 00 à 24), minutes (de 00 à 59) puis secondes (00 à 59) en utilisant ' :' comme séparateur.
  Ici aussi un type dédié *xsd :Time* est mis à disposition par le schéma XML.
– *Le fuseau horaire, Time_Zone :* Les dates et le temps sont définis par rapport à un fuseau horaire. La norme internationale permet d'indiquer le fait qu'une heure est mesurée par rapport au temps universel (UTC), ou, dans le cas contraire, de spécifier le décalage horaire positif ou négatif.
– *La durée, Laps :* Il est également possible d'indiquer une durée d'un évènement (exemple, un LO dure 30 min) ou un intervalle de temps (exemple, une formation doit être suivie entre 14h et 15h). Pour ce faire, deux approches de modélisation sont possibles : La première approche consiste à définir des unités temporelles *Temporal_Unit* afin de mesurer une quantité de temps dans l'absolue (nombre de secondes, de minutes, etc.). La deuxième approche consiste à exploiter les relations de (Allen and Ferguson, 1994) afin de représenter des actions au cours du temps. Exemple, la relation *starts* permet d'indiquer qu'une formation doit démarrer le 2014-10-17 à 14h.

Le lien entre les deux représentations abstraite et concrète du temps est mis en œuvre par des relations simples de l'ontologie permettant d'inférer d'un point temporel des intervalles de temps. De plus, des relations supplémentaires sont définies permettant de lier les concepts de la dimension temporelle à la modélisation des objets pédagogiques (exemple, *has_duration*, *starts*, *ends*, etc.). Cette description sémantique du temps est plus significative dans la vie quotidienne des utilisateurs. La figure 5.6 illustre un aperçu simplifié de la dimension temporelle.

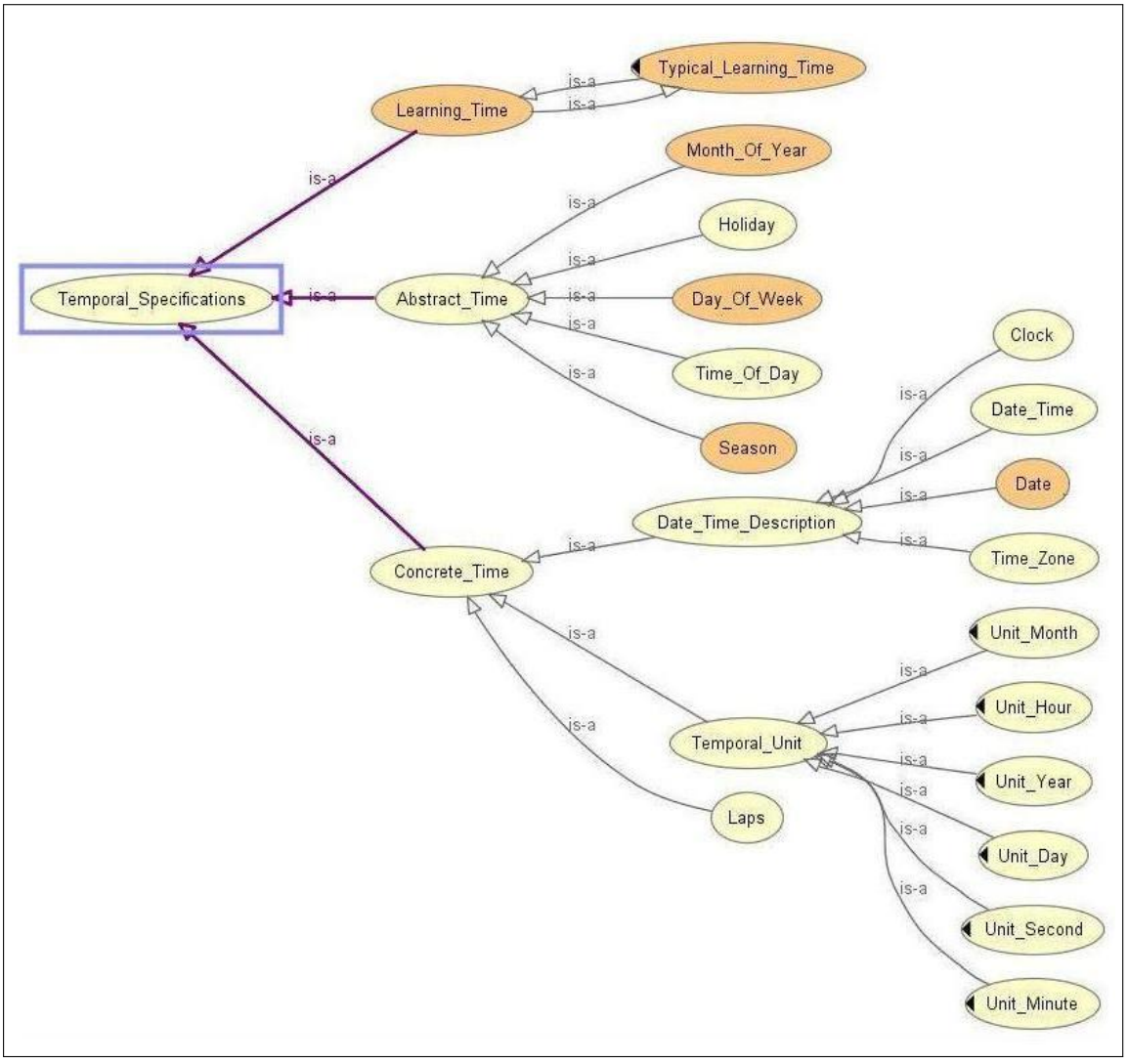

FIGURE 5.6 – Aperçu simplifié des concepts de la dimension temporelle

Modélisation de l'utilisateur :
Afin d'offrir des fonctionnalités personnalisées, un système de recommandation doit : établir les caractéristiques de l'utilisateur à considérer, établir la représentation du modèle utilisateur et finalement choisir ou définir une méthode pour développer le profil conformément à un problème spécifique (Brusilovsky et al., 2007). Un modèle utilisateur ou encore un profil utilisateur est une collection de données personnelles associées à un utilisateur spécifique. L'exploitation d'un profil apprenant est un moyen d'adapter l'apprentissage aux spécificités de chaque utilisateur.

Au sein d'une plateforme d'apprentissage plusieurs types d'utilisateurs interviennent. Ces utilisateurs sont principalement les apprenants, les enseignants (au sens large : responsable, auteur, tuteur, etc.), le personnel administratif et le concepteur informatique (Kouninef et al., 2007). Ces différents acteurs du système exploitent des ressources différentes

pour les tâches qu'ils ont à accomplir : L'enseignant-responsable (*Teacher*) est chargé de former les apprenants et il est responsable d'un ou plusieurs modules d'apprentissage. Un enseignant-auteur (*Author*) crée et rédige le matériel pédagogique se rapportant à sa spécialité. L'enseignant-tuteur (*Tutor*) effectue un suivi du travail des apprenants. L'apprenant (*Learner*) de son côté, consulte en ligne les objets pédagogiques qui lui sont recommandés, organise son travail, effectue des exercices, etc. L'administrateur (*Administrator*), quant à lui, assure l'installation et la maintenance du système et des objets pédagogiques, s'occupe de l'inscription administrative des apprenants, gère les accès et les droits aux contenus, etc. Finalement le concepteur informatique (*Technician*) réalise les objets pédagogiques que ce soit du texte, des images, des vidéos, des parcours entier de formation, etc. Sous le concept *User_Type* nous modélisons les différents acteurs d'un système m-learning (figure 5.7).

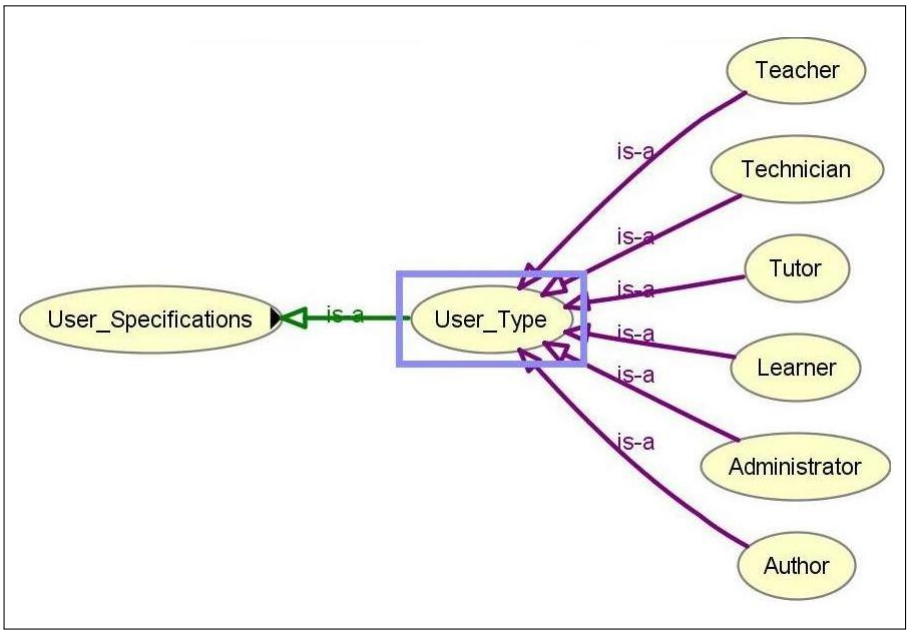

FIGURE 5.7 – Aperçu simplifié des différents acteurs du système

Chaque utilisateur du système est caractérisé par un ensemble de données. Nous proposons de classer ces données en deux catégories : la première catégorie *General_Information*, regroupe des données générales sur l'utilisateur (exemple, nom, prénom, date de naissance, nationalité, etc.) et la deuxième catégorie *Specific_Information*, varie en fonction des différents types d'utilisateurs (exemple, les préférences, les connaissances, etc. concernent l'apprenant ; la réputation, la bibliographie, etc. concernent les auteurs ; les compétences concernent l'administrateur). Afin d'illustrer les caractéristiques des différents profils utilisateurs, nous considérons un fragment de l'otologie présenté dans la figure 5.8.

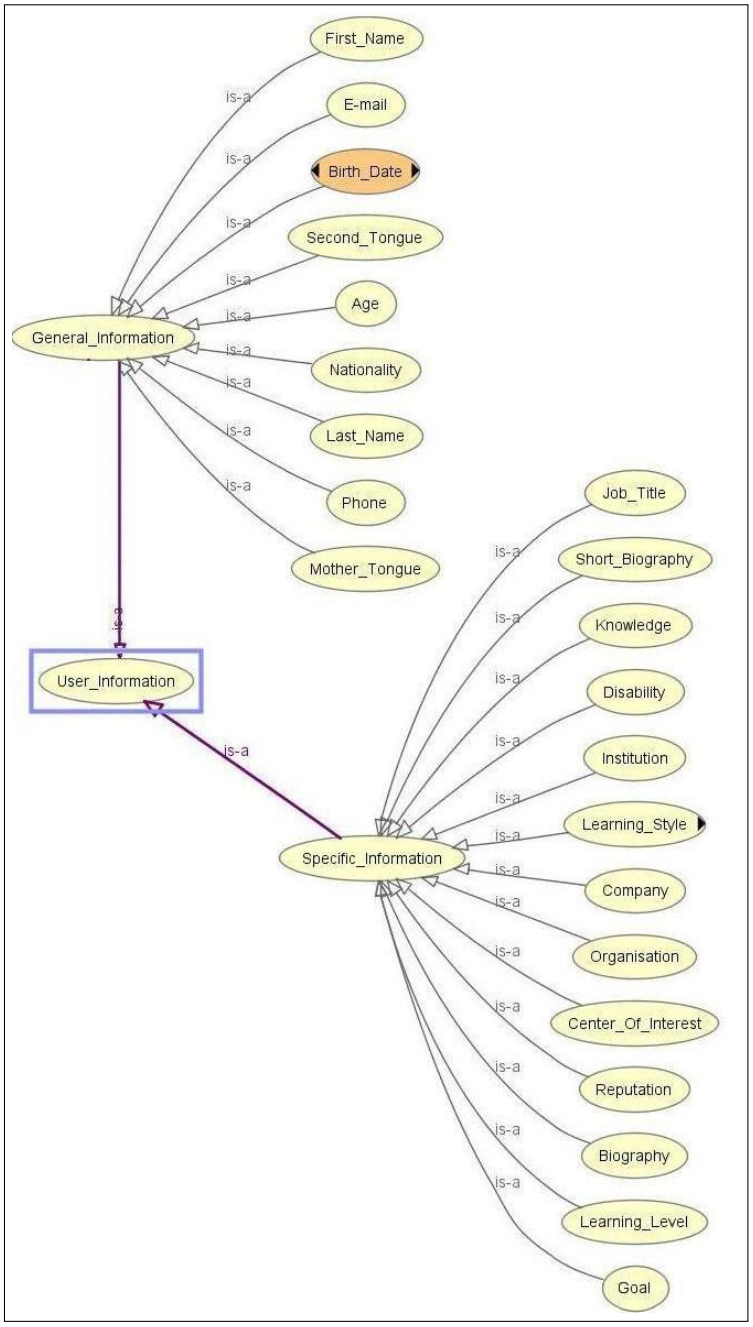

FIGURE 5.8 – Aperçu simplifié des concepts de la dimension utilisateur

Dans la catégorie *General_Information* nous retrouvons les principaux concepts suivants :

– *First_Name :* Prénom de l'utilisateur.
– *Last_Name :* Nom de l'utilisateur.
– *Birth_Date :* Date de naissance de l'utilisateur.
– *Mother_Tongue :* Langue maternelle de l'utilisateur.
– *Second_Tongue :* Autres langues maitrisées par l'utilisateur.
– *Nationality :* Nationalité de l'utilisateur.
– *E_Mail :* Adresse électronique de l'utilisateur.

– *Phone_Number :* Numéro de téléphone de l'utilisateur.
– *Age :* Age de l'utilisateur.
– *Sexe :* Sexe de l'utilisateur.

Dans la catégorie *Specific_Information* nous retrouvons les principaux concepts suivants :
– *Institution :* permet à l'enseignant d'indiquer à quel établissement d'enseignement il est affecté. Exemple, Université de Bourgogne.
– *Company :* permet à l'enseignant d'indiquer dans quelle compagnie il travaille. Exemple, CrossKnowledge.
– *Reputation :* permet d'indiquer la réputation d'un enseignant. Exemple, réputation nationale ou internationale.
– *Job_Title :* permet d'indiquer le type de poste qu'occupe un enseignant. Exemple, professeur.
– *Biography :* sert à justifier la réputation d'un enseignant responsable ou d'un auteur.
– *Short_Biography :* permet de proposer une biographie résumée.
– *Learning_Style :* Il existe aujourd'hui plusieurs modèles permettant de déterminer le style d'apprentissage d'un utilisateur : exemple, le modèle Grasha-Riechmann (Grasha and Riechmann, 1975), le modèle Kolb (Kolb, 1984), le modèle Felder-Silverman (Felder and Carolina, 1988), le modèle VAK (Visuel, Auditif, Kinesthésique) (Fleming and C.Mills, 1992), etc. Bien qu'aucune recherche n'appuie un modèle en particulier, nous préférons dans notre cas utiliser le modèle VAK. Ce modèle de base est grandement employé en raison de sa simplicité. A travers un simple test, il est possible de déterminer les préférences d'apprentissage visuelles, auditives et kinesthésiques/tactiles d'un apprenant.
– *Knowledge :* permet de décrire les connaissances de l'apprenant.
– *Goal :* permet de décrire les buts de l'apprenant.
– *Center_Of_Interest :* permet de décrire les centres d'intérêt de l'apprenant.
– *Disability :* permet à l'apprenant d'indiquer un handicap, afin de déterminer la manière la plus adaptée de présentation des contenus.

Modélisation des dispositifs d'apprentissage :
Afin d'adapter un contenu pédagogique à la technologie mobile censée délivrer l'information, il est nécessaire de connaitre les propriétés de ces technologies. Ces dispositifs d'apprentissage sont caractérisés par différentes propriétés telles que le degré de portabilité du dispositif, la résolution de l'écran, la puissance du processeur, la connectivité, etc. et ils ont pour objectif de proposer une interface d'interaction entre le système et l'utilisateur.
De nombreux travaux de recherches se réfèrent aujourd'hui à des spécifications standards pour la description des dispositifs dans un système, tels que WURFL [6] (sérialisation

6. http ://wurfl.sourceforge.net/

en XML), CC/PP [7] (sérialisation en RDF) et FIPA Device Ontology [8] (sérialisation en OWL). Afin de modéliser notre dimension dispositif, nous proposons d'exploiter la FIPA Device Ontology qui est plus en adéquation avec notre ontologie.

Sous le concept *Device_Specifications*, nous proposons de décrire un dispositif selon trois principales catégories d'informations : des informations générales sur le dispositif (*Device_General_Description*), des informations sur le hardware (*Hardware_Description*) et des informations sur le software (*Software_Description*) (figure 5.9).

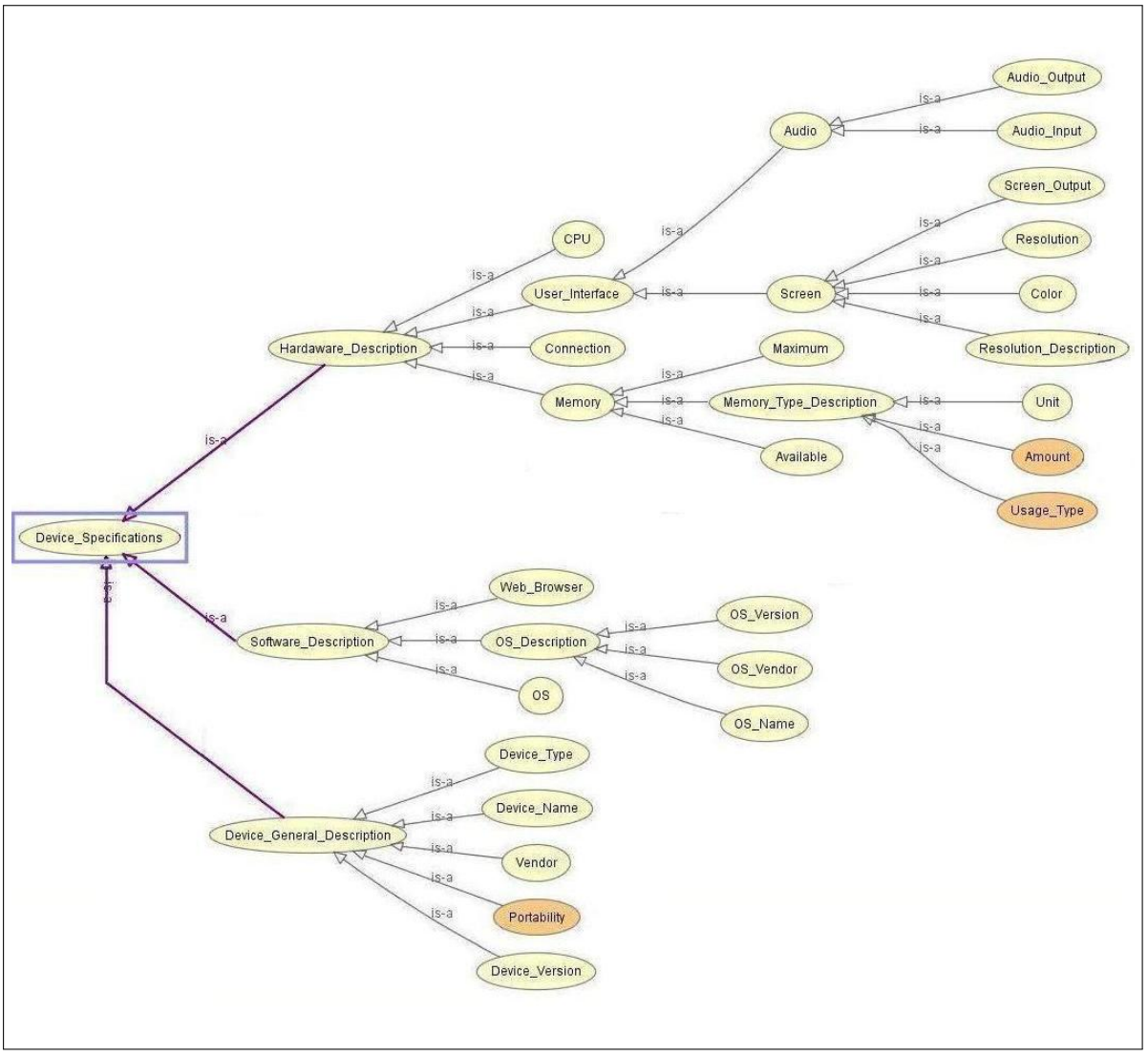

FIGURE 5.9 – Aperçu simplifié des concepts de la dimension dispositif

Nous décrivons dans ce qui suit les principaux concepts utiles à l'adaptation des contenus aux contraintes matérielles du dispositif d'apprentissage :

7. http ://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/
8. http ://www.fipa.org/specs/fipa00091/PC00091A.html

Concept	**Description**		
Device_Name	Nom du produit en question		
Device_Type	Type du dispositif. Exemple, Smartphone, tablette, etc.		
Portability	Degré de portabilité du dispositif		
Vendor	Nom du fournisseur du dispositif		
Device_Version	Version du dispositif		
Connection	Indique le type de connexion utilisée par le dispositif (wifi, 3G, etc.).		
CPU	Indique le type de CPU dont le dispositif dispose.		
User_Interface	*Audio*	*Audio_Output*	Indique si le dispositif est capable de produire une sortie audio.
		Audio_Input	Indique si le dispositif est capable de recevoir une entrée audio.
	Screen	*Width*	Indique la largeur de l'écran.
		Height	Indique la longueur de l'écran.
		Screen_Unit	Indique l'unité utilisée pour mesurer la largeur et la hauteur de l'écran (mm, cm, pouce, etc.).
		Resolution	Décrit la résolution de l'écran.
		Color	Indique la gamme de couleurs utilisée par l'appareil mobile.
Memory	*Maximum*	Indique la quantité maximale de mémoire.	
	Available	Indique la quantité disponible de mémoire.	
	Unit	Indique l'unité utilisée pour exprimer la quantité de mémoire.	
OS	*OS_Name*	Indique le nom du système opérationnel du dispositif.	
	OS_Version	Indique la version du système opérationnel.	
	OS_Vendor	Indique le vendeur du système opérationnel.	
Web_Browser	Indique les propriétés du navigateur Internet.		

TABLE 5.1 – Concepts de la dimension dispositif

Vecteur contexte :
Afin de décrire un contexte apprenant nous choisissons un modèle vectoriel pour agréger nos contextes. Un vecteur contexte contient tous les objets (au sens RDF) appartenant à toutes les données de ce contexte. Il est définit comme le vecteur des vecteurs des données de chaque dimension du contexte.
Soit C un contexte. Un vecteur contexte est noté $V_C = (X_s, X_t, X_u, X_d)$ avec X_s, X_t, X_u et X_d les vecteurs des données, respectivement des dimensions spatiale, temporelle, utilisateur et dispositif.

Soit l'exemple de contexte illustré par la figure 5.10.
Le vecteur contexte de la figure est :

$$V_C = ((2014-10-24T13:40:29, "Monday", "1h"),$$
$$("Restaurant", "Public", "Noisy", "ChezLeon"),$$
$$("Level1", "French", 25), ("Smartphone", "3G", "Android")$$

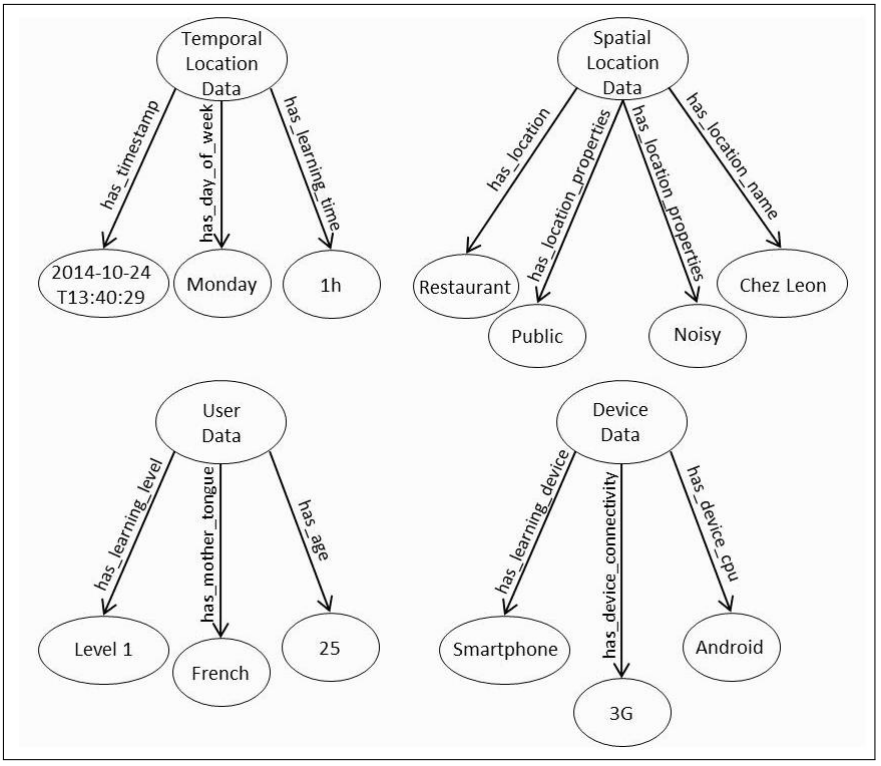

FIGURE 5.10 – Exemple de contexte

5.2.5/ MODÈLE DE CONTRAINTES

Basé sur l'informatique décisionnelle, le modèle de contraintes à pour objectif de décrire les stratégies d'apprentissage et les connaissances des experts métiers. L'informatique décisionnelle est un moyen qui a pour but de faciliter la définition et la mise en œuvre de stratégies gagnantes. Initialement à l'usage des décideurs et des dirigeants d'entreprises, elle s'insère aujourd'hui dans la modélisation des systèmes informatiques. Les systèmes décisionnels reposent sur un ensemble de règles métiers qui décrivent formellement les critères et les processus de décision. Ces règles capitalisent les connaissances d'une entreprise et traduisent sa stratégie en décrivant les actions à mener pour un processus donné. L'une des difficultés consiste à collecter et traduire les textes réglementaires qui n'existent souvent que sous la forme rédigée en un système cohérent et complet de règles formalisées dans un langage naturel contrôlé.

Dans le cas d'un système décisionnel pour le m-learning nous souhaitons collecter les connaissances et les stratégies des experts métier en formation à distance, ainsi que les pratiques des apprenants dans un environnement mobile, et ensuite les traduire sous forme de règles métier. Ces règles synthétisent les connaissances du domaine du e-

learning et des contraintes métiers qui doivent être respectées par le système pour une meilleure adaptation des LO dans un contexte mobile. Dans notre travail, l'ontologie du m-learning fournit un socle pour une modélisation robuste de ces règles.

Cette section décrit un ensemble de règles métiers rédigées pour l'adaptation des LO dans un contexte mobile. Elle décrit aussi la méthode développée pour éditer ces règles en un langage naturel contrôlé (SWRL), à partir de l'ontologie du m-learning, ce qui permet aux experts métier de modéliser des décisions dont la sémantique est formalisée en OWL, sans forcément avoir une bonne connaissance de ce langage.

Le langage SWRL :
SWRL[9] (Semantic Web Rule Language) est un langage de règles pour le Web sémantique, combinant les sous-langages OWL-DL et OWL-Lite avec le sous-langage unaire/binaire de RuleML (Rule Markup Language). La structure des règles SWRL est composée d'un antécédent ou corps (body) et d'un conséquent ou tête (head). Une règle signifie « Si les conditions de l'antécédent sont maintenues, alors les conditions du conséquent doivent également se tenir » :

If Antecedent Then Consequent
Antecedent (Body) → Consequent (Head)

La tête et le corps d'une règle sont composés d'une conjonction d'un ou de plusieurs atomes. SWRL définit des règles sur les instances, les concepts et les rôles OWL :

$$a(?x, ?y) \wedge b(?x, ?y) \wedge c(?x) \wedge \cdots \rightarrow n(?x, ?y)$$

a, b, n : prédicats binaires (rôles)
c : prédicat unaire (concept atomique)
x, y : x variables, instances ou littéraux (nombres, chaînes de caractères, etc.)

Supposons par exemple la règle suivante, si un apprenant *x* à comme langue maternelle *y*, alors *x* peut accéder à des LO *z* qui correspondent à cette langue maternelle. Cette règle exige l'existence des rôles *has_mother_tongue*, *has_language* et *has_access*, et des concepts *Learner* et *Learning_Object* dans l'ontologie OWL. Cette règle est exprimée en SWRL comme suit :

$$Learner(?x) \wedge has_mother_tongue(?x, ?y) \wedge Learning_Object(?z) \wedge has_language(?z, ?y) \rightarrow has_access(?x, ?z)$$

SWRL étend les logiques de descriptions et les axiomes du langage OWL. Il offre la possibilité de créer des règles utilisées pour augmenter les capacités de raisonnement afin de déduire de nouvelles connaissances à partir de l'ontologie OWL.

9. http ://www.w3.org/Submission/SWRL/

D'autres fonctionnalités du langage SWRL résident dans sa capacité à supporter une large gamme de compléments définis par l'utilisateur, appelés built-ins. Ces compléments permettent d'étendre significativement l'expressivité du langage SWRL. Certains compléments sont basés sur des fonctions de comparaison (equal, lessThan, etc.), des fonctions mathématiques (add, multiply, sin, etc.), des fonctions booléennes (booleanNot), des fonctions de traitement des chaines de caractères (stringLength, substring, etc.), etc.
Par exemple, la règle suivante intègre les compléments de comparaison greaterThan et lessThan. Cette règle permet de déterminer si un apprenant appartient à la bonne tranche d'âge pour accéder à un contenu. Notons l'utilisation de l'espace de nommage *swrlb* pour tous les compléments :

$$Learner(?x) \wedge has_age(?x, ?a) \wedge Learning_Object(?z) \wedge$$
$$has_typical_learning_age_min(?z, ?age1) \wedge has_typical_learning_age_max(?z, ?age2) \wedge$$
$$swrlb : lessThan(?a, ?age2) \wedge swrlb : greaterThan(?a, ?age1) \rightarrow$$
$$has_access(?x, ?z)$$

Nous proposons de classer l'ensemble des règles collectées dans le domaine de l'apprentissage en deux principales catégories, (1) des règles d'adaptation au contexte et (2) des règles de précédence : Les règles d'adaptation au contexte permettent de décrire le comportement du système dans un contexte particulier et de pondérer les LO (Le poids d'un LO représente l'intérêt de ce LO pour un apprenant dans un contexte particulier, voir chapitre 6). Les règles d'adaptation au contexte sont classées en deux sous-catégories : des règles génériques et des règles spécifiques au catalogue CrossKnowledge. Les règles de précédence quant à elles, permettent de décrire les contraintes organisationnelles entre les LO.

Exemple 1, règle générique d'adaptation au contexte : Si un apprenant se trouve dans un espace calme, alors proposer des contenus pédagogiques de forte densité sémantique.

$$Learner(?x) \wedge has_location(?x, ?y) \wedge has_location_properties(?y, calm) \wedge Learning_Object(?z) \wedge$$
$$has_semantic_density(?z, high) \rightarrow$$
$$has_access(?x, ?z) \wedge has_rules_weight(?z, ?w) \wedge Weight_DataProperty(?w, 10)$$

Exemple 2, règle d'adaptation au contexte spécifique au catalogue CorssKnowledge : Si un apprenant se trouve dans un espace calme, alors proposer des contenus pédagogiques de type *Session*, qui nécessitent un certain niveau de concentration.

$$Learner(?x) \wedge has_location(?x, ?y) \wedge has_location_properties(?y, calm) \wedge Learning_Object(?z) \wedge$$
$$has_product_name(?z, session) \rightarrow$$
$$has_access(?x, ?z) \wedge has_rules_weight(?z, ?w) \wedge Weight_DataProperty(?w, 10)$$

Exemple 3, règle de précédence : Un LO de type quizz (ici l'instance LOF1) doit être suivi d'un LO de type conclusion.

$$Learning_Object(LOF1) \wedge has_learning_type(LOF1, quizz) \wedge Learning_Object(?x) \rightarrow has_next(LOF1, ?x) \wedge has_learning_type(?x, conclusion)$$

Les règles métier sont définies par les experts du domaine de l'apprentissage. Comme ces experts n'ont pas nécessairement une connaissance du langage de règles SWRL et du langage ontologique OWL, nous avons développé un Web service que nous avons appelé un Système de Génération de Règles Métiers (SGRM) permettant de manipuler facilement les données de l'ontologie de domaine du m-learning (concepts, rôles et individus) et de générer automatiquement les règles en SWRL. L'interface de l'éditeur de règles est la suivante :

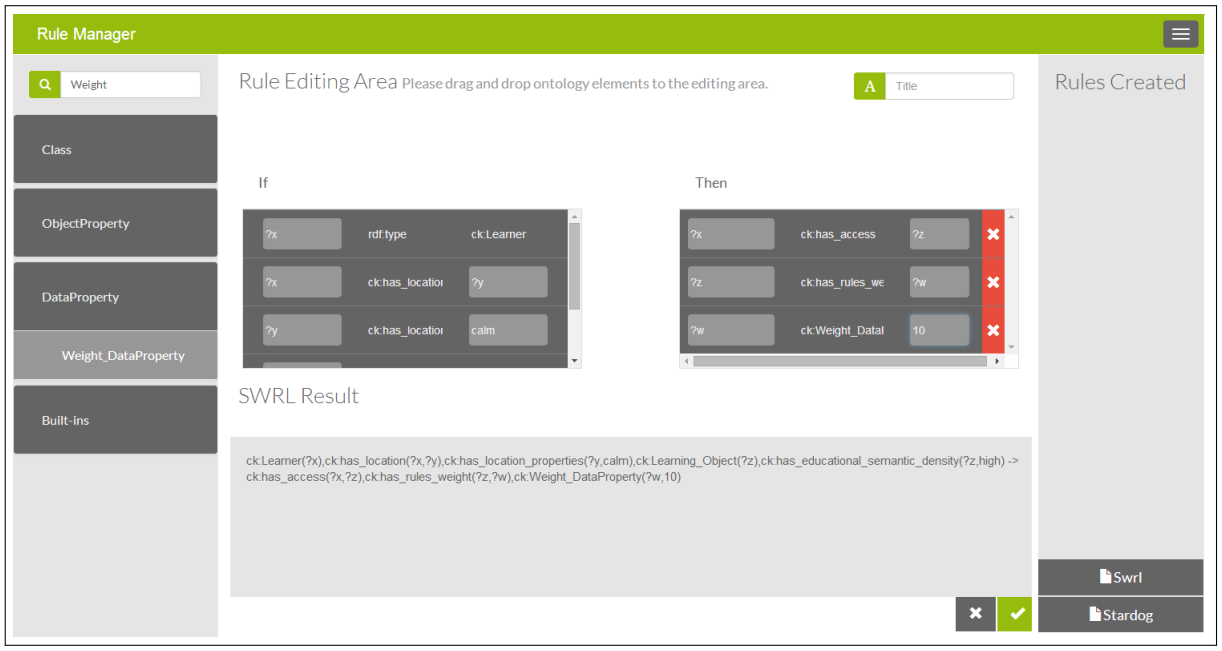

FIGURE 5.11 – Interface de gestion des règles

Nous avons les concepts et les rôles à gauche et la zone d'édition sur la droite. Il suffit de faire un Drag&Drop des éléments de la gauche vers la droite afin de créer une règle SWRL.

Implémentation du serveur sémantique :
Une fois le choix du modèle effectué, les règles de construction de l'ontologie définies et le formalisme de représentation identifié, nous avons créé l'ontologie du m-learning avec l'éditeur Protégé [10].

10. http ://protege.stanford.edu/

L'ontologie a été peuplée avec des LO venant de la base de données de CrossKnowledge en utilisant l'outil d'intégration de données Talend [11] pour créer un jeu de données réelles. Nous avons également peuplé cette ontologie par un jeu de données aléatoires. L'ontologie est sauvegardée dans un triple store de type Stardog [12] intégrant un moteur de raisonnement de type Pellet [13] et SL (voir annexe B).
L'ensemble des règles métiers créées avec le SGRM sont aussi rajoutées au triple store. L'objectif maintenant est d'appliquer sur la base de connaissance obtenue des techniques de raffinement et d'adaptation afin de livrer à l'apprenant un parcours de formation pertinent et optimisé selon son contexte.

5.3/ MODÈLE D'ADAPTATION

Dans le modèle d'adaptation sont implémentés les différents algorithmes utilisés pour proposer des recommandations aux utilisateurs. Les recommandations sont déduites à partir du modèle des LO, des vecteurs contextes et des règles métiers. La figure 5.12 illustre un aperçu simplifié de l'architecture du modèle d'adaptation.
L'objectif du modèle d'adaptation consiste à rechercher un parcours de formation adapté au contexte de l'apprenant. Pour ce faire, plusieurs questions se posent :

– *Quelle doit être la forme de la combinaison ? Autrement dit, de quel type d'éléments doit-elle être composée ?*
  Un LO représente le plus bas niveau de granularité d'un parcours de formation pouvant faire l'objet d'un suivi. Chaque parcours de formation contient ainsi un ensemble de LO destinés à être délivrés sans interruption, et respectant les règles de précédence de manière à former un ensemble cohérent qui va permette de remplir l'objectif de formation, et qui correspond à chaque étape au contexte de l'apprenant, notamment aux supports de délivrance qui sont à sa disposition.
– *Qu'est-ce qui détermine la pertinence d'une combinaison par rapport à une autre ?*
  La pertinence dépend du domaine d'application. Il n'est pas possible de définir une fonction générique pour déterminer la pertinence d'une combinaison par rapport à une autre. Dans le cas le plus simple, la fonction de pertinence peut être la somme des poids d'intérêts de l'utilisateur pour les LO de la combinaison. Mais cela peut être beaucoup plus complexe, elle peut être la composition de plusieurs fonctions.
– *Quelle méthode utiliser pour renvoyer la meilleure combinaison ou du moins s'en approcher ?*
  Cette question fait apparaître un problème d'optimisation combinatoire pour lequel il est nécessaire d'utiliser des algorithmes d'optimisation.

11. http ://fr.talend.com/
12. http ://stardog.com/
13. http ://clarkparsia.com/pellet/

Ces trois questions amènent à la définition de plusieurs concepts que nous présentons dans le chapitre suivant.

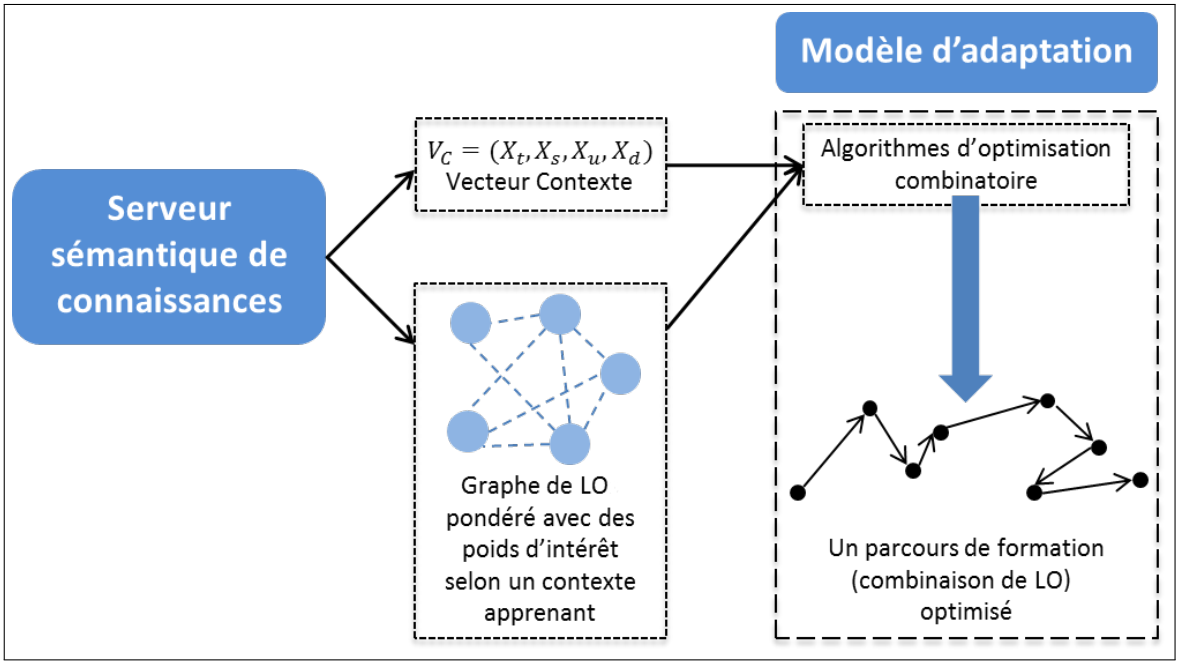

FIGURE 5.12 – Architecture du modèle d'adaptation

5.4/ CONCLUSION

Dans ce chapitre, nous avons présenté notre approche pour un système de recommandation appliqué au domaine du m-learning combinant des technologies du Web sémantique et des algorithmes d'optimisation combinatoire. L'architecture que nous proposons est composée de deux grandes parties : un serveur de connaissances et un modèle d'adaptation. Le serveur de connaissances permet aux formateurs et les experts métiers de représenter leur savoir-faire en utilisant des règles métier et une ontologie pour assurer une hétérogénéité des connaissances. Ensuite, dans un environnement de mobilité, le serveur de connaissance permet de prendre en compte les contraintes de l'environnement et les contraintes utilisateur. Le modèle d'adaptation correspond à des algorithmes d'optimisation combinatoire, dont l'objectif est de proposer à l'apprenant une combinaison pertinente de LO en fonction ses contraintes contextuelles. Dans le chapitre suivant, nous décrivons plus en détail le modèle d'adaptation et définissons plus clairement le problème d'optimisation combinatoire appliqué au domaine du m-learning.

6

PROBLÈME D'OPTIMISATION

Dans le chapitre précédent, nous avons présenté l'architecture d'un nouveau type de système de recommandation pour l'apprentissage mobile. L'objectif maintenant est de trouver un ensemble d'objets pédagogiques corrélés, formant un parcours optimisant l'expérience de formation de l'apprenant. La recommandation doit prendre en compte les objectifs de formation, mais également les supports disponibles pour dispenser cette formation, et le contexte d'apprentissage du moment. Dans ce cadre, nous désirons instancier cette architecture. Il est nécessaire alors d'identifier au préalable le problème d'optimisation qui est posé dans la recherche de combinaisons d'objets pédagogiques.
Ce chapitre s'articule en quatre parties. Dans la première partie, nous décrivons et définissons formellement le problème d'optimisation en apprentissage mobile. Ce problème ne pouvant être résolu de façon exacte par des algorithmes classiques en un temps raisonnable, la seconde partie aborde le domaine des métaheuristiques. Les métaheuristiques permettent d'approcher la solution optimale à un problème donné en un temps raisonnable. Les métaheuristiques les plus connues et les plus utilisées sont présentées dans la deuxième partie du chapitre. Cette partie permet d'identifier les méthodes les plus adaptées au problème d'optimisation qui nous est posé. Dans la troisième partie de ce chapitre, nous présentons les métaheuristiques candidates implémentées et réadaptées à notre problème de recommandation en m-learning. Nous présentons d'abord un algorithme simple de Hill-Climbing, puis des algorithmes plus complexes tels que l'algorithme de recuit simulé, l'algorithme tabou et l'algorithme basé sur une recherche à voisinage variable, pour avoir des solutions de meilleures qualités par rapport à une méthode simple. Dans la quatrième partie du chapitre nous proposons de comparer l'efficacité de ces différentes métaheuristiques. Des benchmarks de comparaison sont dressés.

Sommaire

6.1 Définition du problème d'optimisation combinatoire 128
6.1.1 Formalisation des LO 129

6.1.2 Pondération des LO 130
6.1.3 Graphe des LO 132
6.1.4 Pertinence d'une solution 135
6.2 Les métaheuristiques 137
6.2.1 Introduction aux métaheuristiques 137
6.2.2 Méthodes métaheuristiques approchées 138
6.2.3 Méthodes constructives 139
6.2.4 Méthodes de recherche locale / à solution unique 140
6.2.5 Méthodes évolutives / à population de solution 147
6.3 Implémentation des métaheuristiques 154
6.3.1 Construction d'une solution initiale par un algorithme de type glouton 154
6.3.2 Amélioration de recherche locale par un algorithme Hill-Climbing 156
6.3.3 Amélioration de recherche locale par un algorithme de recuit simulé 157
6.3.4 Amélioration de recherche locale par un algorithme tabou 158
6.3.5 Amélioration de recherche locale par un algorithme VNS 160
6.4 Résultats expérimentaux 163
6.4.1 Jeux de données 163
6.4.2 Paramétrage des algorithmes et résultats 164
6.5 Conclusion 166

6.1/ DÉFINITION DU PROBLÈME D'OPTIMISATION COMBINATOIRE

Dans le domaine de l'apprentissage électronique, un LO représente le plus bas niveau de granularité d'un parcours de formation, pouvant faire l'objet d'un suivi. Des règles de précédence entre les LO sont définies, pour préciser que certaines notions doivent impérativement être assimilées avant d'autres. A chaque fois que l'apprenant se connecte à la plateforme d'apprentissage, un parcours de session doit être proposé à l'apprenant, correspondant à son contexte actuel. Le parcours de formation global, est formé de l'ensemble des parcours de formation calculés aux différentes sessions.
Le parcours de formation global doit contenir un ensemble de LO destinés à être délivrés à l'apprenant, respectant les règles de précédence. Ces LO doivent aussi former un ensemble cohérent, qui va permettre de remplir tous les objectifs de la formation, et qui correspondent à chaque étape au contexte actuel de l'apprenant, notamment aux supports de délivrance qui sont à sa disposition. Le parcours de formation global créé doit optimiser l'expérience d'apprentissage. Cette optimisation intervient sur trois plans : premièrement, la pertinence des supports de formation par rapport au contexte actuel de l'utilisateur, deuxièmement la maximisation du gain de compétences, et finalement la minimisation de l'écart entre la durée totale prévue pour la formation et le durée autorisée par le formateur.

Dans cette partie du chapitre, nous formalisons le problème décrit ci-dessus. Ensuite nous décrivons la notion de pondération des LO, de graphe des LO et de pertinence d'une solution.

6.1.1/ FORMALISATION DES LO

Les apprenants sont représentés dans notre système par des profils. Ces profils utilisateurs sont stockés dans le modèle de contexte. Dans la suite, nous considérerons seulement un utilisateur, le processus pour produire une solution pouvant être réalisé indépendamment pour chaque utilisateur du système.

Soit Ty l'ensemble des types de support d'apprentissage disponibles pour un apprenant. Par exemple, $Ty = \{ordinateur,\ tablette,\ smartphone\}$. On associe à l'ensemble des types de support une numérotation linéaire suivant la bijection $N : Ty \rightarrow \{1, ..., |Ty|\}$. Soit ty_k le type de support $N^{-1}(k)$. Pour plus de simplicité dans la suite nous nommerons directement ce support le support k.

Soit $d_{i,max}$ la durée maximale de la i ème session de formation, indiquée par l'apprenant lors de sa connexion à l'application (le temps d'un trajet en bus, pendant 1h, etc.), et soit D_{max} la durée maximale autorisée d'un parcours de formation global indiqué par le formateur.

Le modèle des LO représente la connaissance sur les contenus d'apprentissage, grâce à l'utilisation d'une ontologie composée de concepts, relations entre concepts, et individus. Les LO proposables sont intégrés dans l'ontologie en tant qu'individus. Par commodité, nous réduisons le modèle des LO à l'ensemble LO des items proposables.
Un objet pédagogique $lo \in LO$ est une brique de formation dépendante d'un support d'apprentissage de type ty. Par exemple, un lo de type *tablette*, signifie que ce lo est accessible sur un support d'apprentissage de type $tablette$.
Les LO sont linéairement ordonnés de lo_1 à $lo_{|LO|}$. Un lo est représenté par un vecteur composé d'un identifiant, d'une durée, d'un type de support et d'un poids :

$$\begin{cases} \text{Soit } id_i \text{ l'identifiant du } lo_i. \\ \text{Soit } duree_i \text{ la durée du } lo_i. \\ \text{Soit } support_i \text{ le support sur lequel le } lo_i \text{ est accessible, avec } support_i \in Ty. \\ \text{Soit } W_{i,final} \text{ le poids pédagogique du } lo_i \text{ (Pour plus de précisions sur le calcul de cette donnée ,voir section 6.1.2).} \end{cases}$$

$$lo_i = < id_i,\ duree_i,\ support_i,\ W_{i,final} >,\ i \in [1, ..., |LO|]$$

Par exemple $lo_1 = < wo89,\ 15,\ tablette,\ 10 >$, représente l'objet pédagogique lo_1 d'identifiant $wo89$, d'une durée de 15 min, accessible sur $tablette$ et de poids pédagogique égal

à 10.

Nous ajoutons à l'ensemble des supports physiques Ty, trois types de supports virtuels destinés à représenter des situations particulières. Ces trois types de support virtuels sont : $objectif$, $initial$, et fin. Un lo de type $initial$ est utilisé pour démarrer un parcours de formation. Un lo de type fin est utilisé pour clôturer un parcours de formation. Finalement, un lo de type $objectif$ est utilisé pour valider un ensemble d'objets pédagogiques obligatoires à atteindre dans le parcours de formation.
Un lo de type $objectif$ ne représente pas une connaissance, mais il sert à la validation d'un ensemble de LO obligatoires pour atteindre un objectif de la formation. Soit $OBJ \subset LO$ l'ensemble des LO de type *objectif*. Ces LO ne sont pas ordonnés et ne doivent pas être réalisés dans un ordre imposé, prédéfini, mais il est nécessaire que tous les LO de type $objectif$ soient atteints pour valider la formation.
$\forall i$, un $lo_i \in OBJ$ à les caractéristiques suivantes : $duree_i = 0$, $support_i = objectif$, et $W_i = 0$ (un objet pédagogique de type $objectif$ étant neutre pédagogiquement, nous le pondérons à nul).

Nous décrivons dans la section suivante notre méthode pour pondérer les LO.

6.1.2/ Pondération des LO

Le poids du lo_i noté W_i, représente l'intérêt de ce LO pour l'apprenant dans un contexte particulier. Le calcul des poids est réalisé dans le modèle de connaissances à chaque changement de contexte selon l'algorithme 1. Cet algorithme se décompose en plusieurs étapes :

- Étape 1 : Les formateurs et les experts métiers attribuent une première pondération à l'ensemble des LO. Cette pondération représente l'intérêt d'un LO dans une formation précise. Plus le LO contient une connaissance pédagogique importante, plus son poids est important.
  $\forall i$, notons $W_{i,0}$ le poids initial attribué par l'expert au lo_i.
- Étape 2 : Les règles métier décrites dans le chapitre précédent sont utilisés pour modifier le poids des LO. L'idée est de réorganiser et filtrer les LO suivant les contraintes de l'apprenant dans un contexte particulier.
  Notons $W_{i,j}$ le poids évolué du lo_i après l'application d'une règle métier j, $j \in [1, \ldots, Nb_{regles}]$. Plus le poids attribué par une règle j à un lo_i est important, plus il exprime une forte importance de ce lo dans un contexte particulier.
  Plusieurs règles métiers peuvent être appliquées à un lo_i, par conséquent chaque nouvelle règle fait évoluer le poids de ce LO selon l'équation suivante :

$$W_{i,j} = W_{i,j-1}.\lambda_j \tag{6.1}$$

avec $\lambda_j = 0$ si lo_i n'est pas adapté au contexte, si non $\lambda_j \in [1, WMax]$ correspondant au poids attribué par la nouvelle règle ($WMax$ étant une valeur maximale fixée par l'expert métier).

Exemple de règles de pondération :

$$R1 : Learner(learner1) \wedge has_mother_tongue(learner1, francais) \wedge Learning_Object(LOFRANC) \wedge has_language(LOFRANC, francais) \rightarrow has_access(learner1, LOFRANC) \wedge has_rules_weight(LOFRANC, W10) \wedge Weight_DataProperty(W10, 10)$$

$$R2 : Learner(learner1) \wedge has_mother_tongue(learner1, francais) \wedge Learning_Object(LOANG) \wedge has_language(LOANG, anglais) \rightarrow has_access(learner1, LOANG) \wedge has_rules_weight(LOANG, W5) \wedge Weight_DataProperty(W5, 5)$$

$$R3 : Learner(learner1) \wedge has_mother_tongue(learner1, francais) \wedge Learning_Object(LORUS) \wedge has_language(LORUS, russe) \rightarrow has_rules_weight(LORUS, W0) \wedge Weight_DataProperty(W0, 0)$$

Dans cet exemple, la première règle exprime que si l'apprenant *learner1* a comme langue maternelle *français*, alors il pourra accéder à l'objet pédagogique *LOFRANC*, dont le contenu est exprimé en langue *française*. De plus l'objet pédagogique *LOFRANC* est pondéré par la valeur 10, exprimant une forte préférence, par rapport un objet pédagogique dont le contenu est exprimé en une autre langue : dans la deuxième règle l'objet pédagogique *LOANG* rédigé en langue anglaise est pondéré par la valeur 5 exprimant une moyenne préférence pour ce contenu, et dans la troisième règle l'objet pédagogique *LORUS* rédigé en langue russe est pondéré avec la valeur 0 exprimant que ce contenu n'est pas adapté au contexte.

– Étape 3 : La troisième étape de pondération est obtenue à partir des réponses de l'apprenant au questionnaire VAK, présenté dans le chapitre précédent. Suite à ce questionnaire, on peut déterminer les préférences de l'apprenant concernant les styles d'apprentissage, en récupérant les scores obtenus pour les différentes catégories de VAK. Ces scores, sont ensuite exploités pour pondérer l'ensemble des LO. L'objectif est d'augmenter le poids des LO correspondant au style d'apprentissage de l'apprenant.
Notons $Score_{i,VAK}$ le score VAK obtenu pour pondérer un lo_i. A la fin de cette étape nous obtenons un poids final $W_{i,final}$ selon l'équation suivante :

$$W_{i,final} = W_{i,Nb_{regles}}.Score_{i,VAK} \tag{6.2}$$

Pour résumer, le processus de pondération procède selon les étapes de l'algorithme décrit ci-dessous :

Dans la section suivant nous présentons la notion de graphe des LO et nous décrivons notre problème de construction d'un parcours de formation valide dans ce graphe.

Algorithm 1 : Algorithme de pondération

Input :
Un ensemble de LO non pondérés
Output :
Un ensemble de LO pondérés selon un contexte particulier

for $lo_i \in LO$ **do**
  $W_{i,0} \leftarrow$ poids initial attribué par l'expert au lo_i
  for $j = 1$ to Nb_{regles} **do**
    $W_{i,j} \leftarrow W_{i,j-1}.\lambda_j$
  end for
  $W_i \leftarrow W_{i,final}.Score_{i,VAK}$
end for

6.1.3/ GRAPHE DES LO

L'ensemble des LO accessibles dans le système sont reliés entre eux par différentes relations sémantiques (exemple, *has_next*, *has_previous*, *is_composed_of*, *is_form_of*, *needs*, etc.). Ces différentes relations sont définies par les experts métier, pour organiser les LO et pour préciser que certains LO doivent impérativement être assimilés avant d'autres. Nous proposons de représenter l'ensemble des LO recommandables par le système par un graphe pondéré.

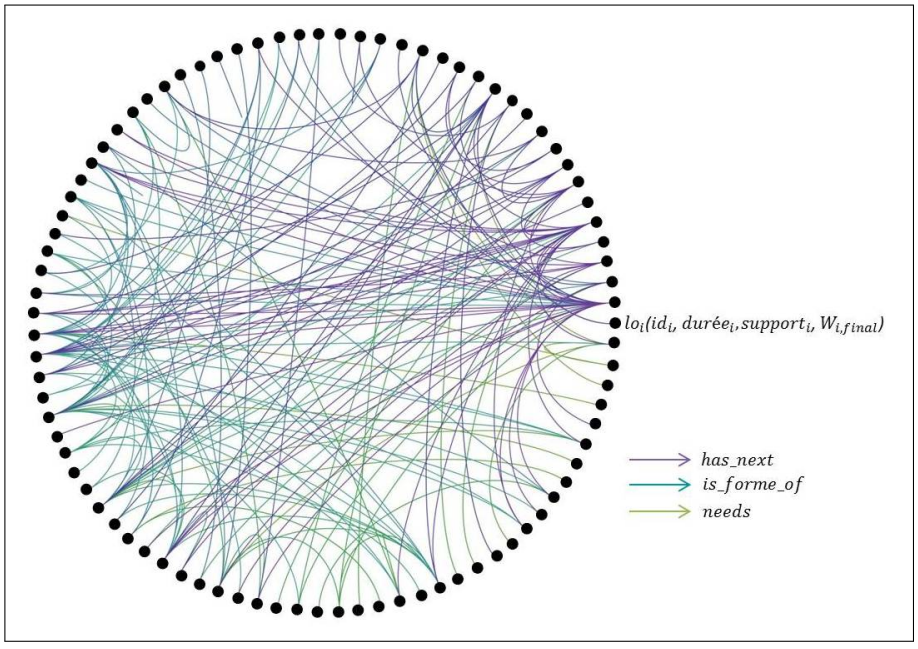

FIGURE 6.1 – Aperçu simplifié du graphe des LO

La figure 6.1 illustre un aperçu simplifié du graphe des LO :
– chaque LO recommandable est représenté par un nœud,
– chaque nœud possède un identifiant, une durée, un support et une pondération,

– chaque relation sémantique entre une paire de LO est représentée par un arc,
– et chaque type de relation sémantique par une couleur (exemple, dans la figure 6.1, trois types de relations sémantiques sont représentées).

De cet ensemble de LO et de relations nous construisons un graphe orienté pondéré $G = (LO, A)$ regroupant l'ensemble des parcours de formations possibles, où :
– LO est l'ensemble des sommets du graphe.
– A est un ensemble d'arcs, c'est-à-dire un ensemble de paires ordonnées de sommets : $A \subset LO \times LO$. Chaque arc du graphe représente une transition pédagogiquement autorisée entre deux objets pédagogiques.

Supposons qu'un apprenant n'ait à sa disposition qu'un seul type de support de formation de type k, et que tous les LO recommandables par le système sont accessibles sur ce support. Un parcours de formation valide, dans ce cas, serait un ensemble de LO enchaînés grâce aux transitions pédagogiques autorisées, et validant l'ensemble des objectifs de la formation. Dans ce cas le calcul d'un parcours de formation peut se ramener à un problème de recherche de cheminement dans le graphe orienté G.

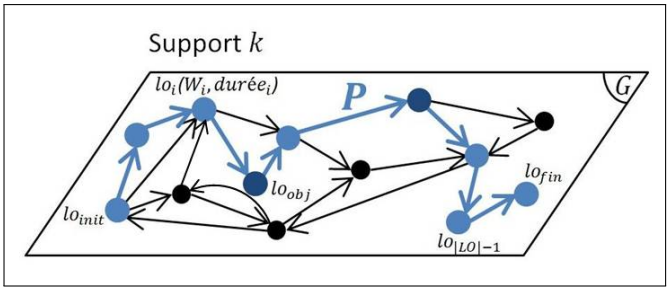

FIGURE 6.2 – Parcours mono-support

Un parcours de formation global P (concaténation des parcours de formations calculés à chaque session) est dans ce cas un chemin dans le graphe G. Un chemin est une suite de LO de G tels que deux objets pédagogiques consécutifs quelconques, lo_i et lo_{i+1}, sont reliés par un arc de G (figure 6.2). Ce parcours est dit valide s'il passe par tous les LO de type $objectif$. L'origine et l'extrémité du chemin P sont définies comme suit :
– $lo_{init} \in D$, origine du chemin. lo_{init} est un LO virtuel de support $initial$.
– $lo_{fin} \in OBJ$, extrémité du chemin. lo_{fin} est un LO virtuel de support fin.

$$P = (lo_{init}, \ldots, lo_{fin} \ / \forall 1 \leq i \leq |LO|,\ lo_i \in LO) \subset G$$
$$sous\ contrainte\ \forall lo_i \in OBJ,\ lo_i \in P$$

Soit V_{alid} l'ensemble des parcours valides.

Si chaque LO était accessible sur chaque type de support de formation, il serait aisé de choisir à tout instant le meilleur support permettant de délivrer l'enseignement de la manière la plus adaptée au contexte de l'apprenant. Les cas réels que nous avons étudiés nous ont montré au contraire une grande hétérogénéité des supports disponibles

selon les LO. Seulement les LO proposés ont une structure, une durée différente et un poids différent en fonction du type de support, ce qui interdit de changer de support de délivrance en cours de formation, sans risquer la redondance de certaines briques de contenu, ou la présence de contenus absents de l'objectif de la formation.
Le problème peut se ramener à un problème de recherche de chemin multimodal. La multimodalité est un problème étudié dans le domaine du transport. Il s'agit d'un processus de transport par le biais d'une interconnexion de réseaux, impliquant diverses combinaisons de modes de transport, dans lequel toutes les composantes sont liées pour former une coordination efficace. Ce problème difficile, très étudié ces dernières années (Gräbener et al., 2010), consiste à rallier un point B à partir d'un point A en empruntant divers moyens de transport, avec des temps de parcours, des itinéraires et des coûts de transport différents.
Nous pouvons faire le rapprochement avec notre problème, en considérant que le parcours de formation optimal est égal au cheminement pour rallier les objectifs de formation par différents moyens de transport (différents supports de formation), partant de lo_{init} jusqu'à arriver à lo_{fin} et passant par tous les LO objectifs. Tout comme deux trajets peuvent suivre des itinéraires différents selon le moyen de transport, deux parcours de formation peuvent comporter des LO différents. Tout comme le temps de parcours entre deux points varie en fonction du moyen de transport utilisé (exemple, un parcours à pied est plus long qu'en bus), le temps nécessaire pour parcourir un ensemble de LO peut varier en fonction du support de diffusion (exemple, une formation sur ordinateur est souvent plus longue qu'une formation sur Smartphone). Enfin, la disponibilité de chaque support de formation varie dans le temps, tout comme la disponibilité des moyens de transport.

Le réseau de transport multimodal est souvent modélisé par un graphe multi-niveaux. Dans ce type de structure, chaque niveau est généralement associé à un mode de transport et les niveaux sont reliés entre eux afin de représenter les possibles changements de mode. Parallèlement, l'ensemble des parcours de formations possibles peut être modélisé comme une superposition d'un ensemble de sous-graphes.
L'ensemble des objets pédagogiques LO est éclaté en plusieurs sous-ensembles, chacun regroupant les objets pédagogiques accessibles sur un support k : $LO = \cup LO_k, \forall k \in Ty$.
De même, l'ensemble A des arcs est éclaté en plusieurs sous-ensembles, chacun regroupant les arcs reliant les LO accessibles sur un support k.
Les différents niveaux du graphe sont reliés par un ensemble de liens qu'on note $A_{liaison}$.
L'ensemble des parcours de formation possibles est modélisé ainsi par un graphe multimodal comme suit : $G = (\cup LO_k, \cup A_k \cup A_{liaison}), \forall k \in Ty$.
Par exemple, dans la figure 6.3, nous pouvons considérer trois sous graphes : $G_s = (LO_s, A_s)$ représente l'ensemble des parcours de formations possibles sur Smartphone, $G_t = (LO_t, A_t)$ représente l'ensemble des parcours de formations possibles sur tablette et $G_o = (LO_o, A_o)$ représente l'ensemble des parcours de formations possibles sur ordinateur. Nous rajoutons une quatrième couche OBJ regroupant l'ensemble des LO de type *objectif*. Le graphe $G = (LO, A)$ représentant l'ensemble des parcours de formations

possibles multimodaux, est donc composé de G_s, G_t, G_o, OBJ et d'un ensemble d'arcs assurant la jonction entre ces différents niveaux.

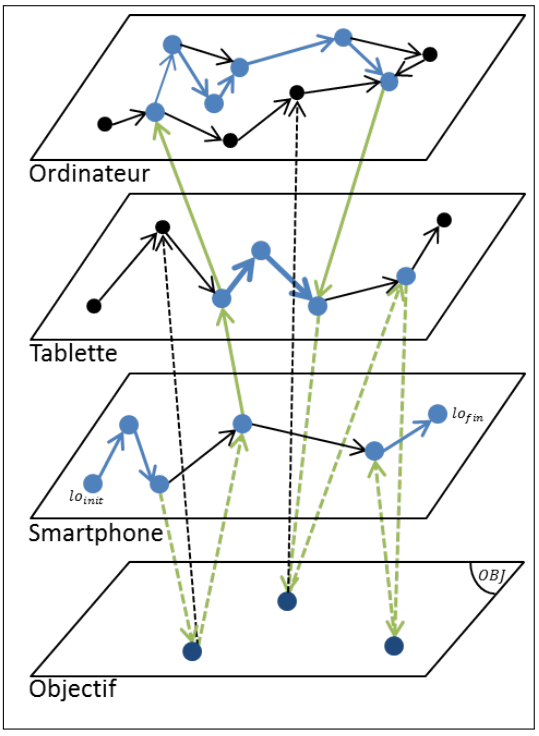

FIGURE 6.3 – Modélisation simplifiée multi-support

L'objectif est de trouver un chemin multimodal (concaténation de chemins monomodaux) entre deux nœuds lo_{init} et lo_{fin} du graphe G passant par tous les LO $objectif$ maximisant une fonction de *pertinence* que nous présentons dans la section 6.1.4.

6.1.4/ PERTINENCE D'UNE SOLUTION

Dans notre problème il s'agit de construire un parcours de formation global valide formé d'un ensemble de LO. Les LO choisis doivent maximiser le poids total de la formation globale pour maximiser le gain de connaissance et recommander les LO les plus adaptés au contexte, et minimiser l'écart entre la durée totale prévue et la durée D_{max} autorisée de formation.

En termes mathématiques, nous définissons les objectifs à optimiser par les deux équations suivantes :

– Maximiser la satisfaction de l'apprenant ⇒ maximiser le poids total du parcours de formation :

$$maximiser \sum_{i=1}^{|LO|} W_i \cdot x_i, \;\; x_i \in \{0,1\} \tag{6.3}$$

où x_i est égale à 1 si lo_i est choisi dans la solution ($lo_i \in P$), 0 sinon.

– Minimiser l'écart entre la durée totale prévue et la durée D_{max} autorisée de formation :

$$minimiser\ (D_{max} - \sum_{i=1}^{|LO|} durée_i \cdot x_i),\ x_i \in \{0, 1\} \tag{6.4}$$

où x_i est égale à 1 si lo_i est choisi dans la solution ($lo_i \in P$), 0 sinon.

Les deux objectifs que nous souhaitons optimiser sont contradictoires. D'après le principe de dualité (transformer une minimisation en une maximisation en multipliant la fonction objectif par -1), nous considérons que les fonctions objectifs à atteindre sont à maximiser.
Nous définissons finalement la fonction $Pert$ afin de calculer la pertinence d'une solution, cette pertinence étant l'agrégation des deux objectifs décrits précédemment :

$$Pert = \alpha \cdot \sum_{i=1}^{|LO|} W_i \cdot x_i - \beta \cdot (D_{max} - \sum_{i=1}^{|LO|} durée_i \cdot x_i),\ \alpha\ et\ \beta \in [0, 1] \tag{6.5}$$

L'optimisation du parcours de formation consiste donc à trouver un parcours de formation qui maximise la valeur de la fonction $Pert$:

$$maximiserPert(P), P \in V_{alid} \tag{6.6}$$

Une fois le parcours de formation global calculé, il est proposé à l'apprenant une section de ce parcours telle que :

$$\sum_{i=1}^{|LO|} durée_i \leq d_{max} \tag{6.7}$$

Les problèmes de cheminement dans un graphe sont des problèmes d'optimisation combinatoire bien connus. Le problème classique peut être résolu facilement avec une complexité polynomiale. Mais cette complexité augmente significativement dans notre cas, lorsque toutes les permutations des LO sont autorisées et lorsque le problème à résoudre passe d'un problème de cheminement classique à un problème d'optimisation multimodal. Le problème devient de complexité NP-difficile (Garey and Johnson, 1990). Ceci implique qu'une méthode exacte ne serait pas efficace pour résoudre notre problème, en particulier pour une application de décision en temps réel (Chen et al., 1999).
Compte tenu de ces difficultés, la plupart des experts de l'optimisation combinatoire ont orienté leurs recherches vers le développement de méthodes heuristiques.
Dans la deuxième partie de ce chapitre, nous présentons les métaheuristiques les plus connues pour résoudre des problèmes combinatoires difficiles.

6.2/ LES MÉTAHEURISTIQUES

Les métaheuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle pour résoudre des problèmes d'optimisation réputés difficiles. Une métaheuristique peut être définie comme étant une procédure exploitant au mieux la structure d'u problème, dans le but de trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible (Nicholson, 1971) (Widmer, 2001). Résoudre un problème d'optimisation combinatoire, c'est trouver l'optimum d'une fonction, parmi un nombre fini de choix, souvent très grand.
Nous proposons dans cette partie du chapitre de fournir une vue sur l'ensemble du domaine des métaheuristiques. En premier lieu nous introduisons brièvement quelques notions des métaheuristiques. Ensuite nous présentons les deux grandes classes des métaheuristiques existantes : les méthodes de recherche locale et les méthodes évolutives. Ce bref état de l'art sur les métaheuristiques est basé sur le rapport de (Autin, 2006) et le travail de (Picot-Clémente, 2011).

6.2.1/ INTRODUCTION AUX MÉTAHEURISTIQUES

En mathématiques, l'optimisation recouvre toutes les méthodes qui permettent de déterminer l'optimum d'une fonction. Un problème d'optimisation combinatoire est définit comme suit :
Chaque instance du problème est associé un ensemble de solutions S et un sous-ensemble X de S. X représente l'ensemble des solutions admissibles du problème. L'ensemble X des solutions admissibles est supposé fini et est en général défini par un ensemble C de contraintes. A cette instance est aussi associée une fonction f, appelée fonction objectif, permettant d'évaluer chaque solution admissible. f attribue à chaque solution $s \in X$ la valeur $f(s)$. Pour résoudre un problème d'optimisation combinatoire il faut trouver une solution $s \in X$ optimisant la valeur de la fonction objectif f. Dans le cas d'une maximisation le problème à résoudre est donc le suivant :

$$max f(s), s \in X \tag{6.8}$$

Une structure de voisinage (ou tout simplement un voisinage) est une fonction V qui associe un sous-ensemble de S à toute solution s. Une solution $s* \in V(s)$ est dite voisine de s. Formellement, on cherche donc $s* \in X$ tel que $f(s*) \geq f(s)$ pour tout $s \in X$.
Une solution $s \in X$ est dite optimum local relativement à la structure de voisinage V si $f(s*) \geq f(s)$ pour tout $s* \in V(s)$. Une solution $s \in X$ est dite optimum global si $f(s*) \geq f(s)$ pour tout $s* \in X$ (figure 6.4). De même, il est aussi bien possible de résoudre des problèmes de minimisation ; les principes de résolution restent les mêmes.

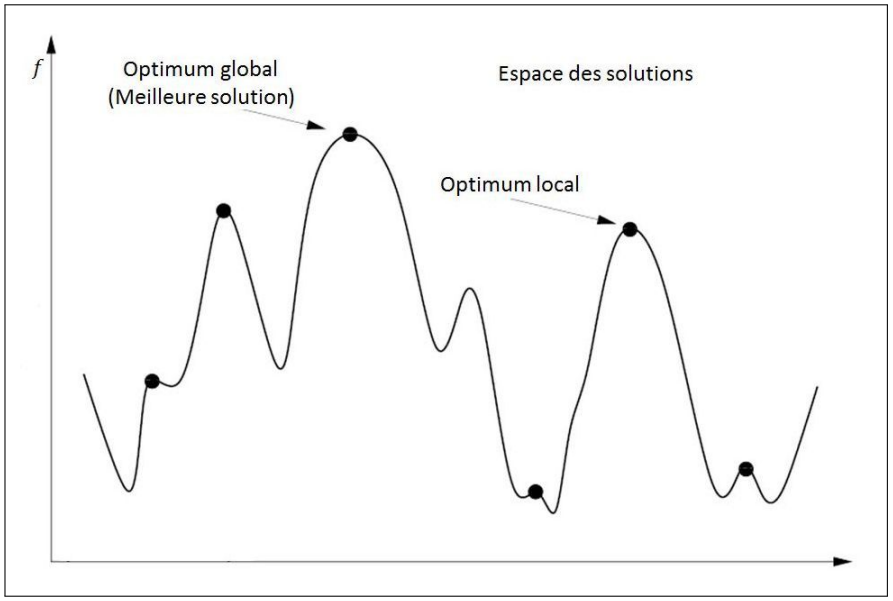

FIGURE 6.4 – Fonction de coût d'un problème combinatoire

La résolution des problèmes d'optimisation combinatoires est assez délicate lorsque le nombre fini de solutions admissibles croît avec la taille du problème, ainsi que sa complexité. La résolution de notre problème atteint rapidement une taille critique de X, au-dessus de laquelle même une énumération partielle des solutions admissibles devient très couteuse en temps de calcul. On parle alors d'optimisation difficile, ou de problèmes *NP-difficiles*. Compte tenu de cette difficulté, la plupart des spécialistes de l'optimisation combinatoire ont orienté leur recherche vers le développement de méthodes heuristiques permettant de résoudre un problème en un temps polynomial, tout en préservant une certaine qualité des solutions trouvées.

6.2.2/ MÉTHODES MÉTAHEURISTIQUES APPROCHÉES

On peut distinguer trois classes de métaheuristiques :

– *Les méthodes constructives :* Ces algorithmes construisent une seule solution par une suite de choix partiels et définitifs, c'est-à-dire sans retours en arrière. On les appelle méthodes gloutonnes. A chaque itération une méthode gloutonne cherche à faire le choix le plus avantageux parmi la liste des choix potentiels.

– *Les méthodes de recherche locale / à solution unique :* Ces algorithmes partent d'une solution initiale (obtenue de façon exacte, ou par tirage aléatoire) puis passent d'une solution à une autre dans l'espace des solutions admissibles jusqu'à ce qu'une solution considérée comme optimale soit trouvée ou que le temps imparti soit dépassé (figure 6.5). Dans cette catégorie, se rangent : la méthode Hill-Climbing, le recuit simulé, la méthode tabou et la recherche par voisinage variable.

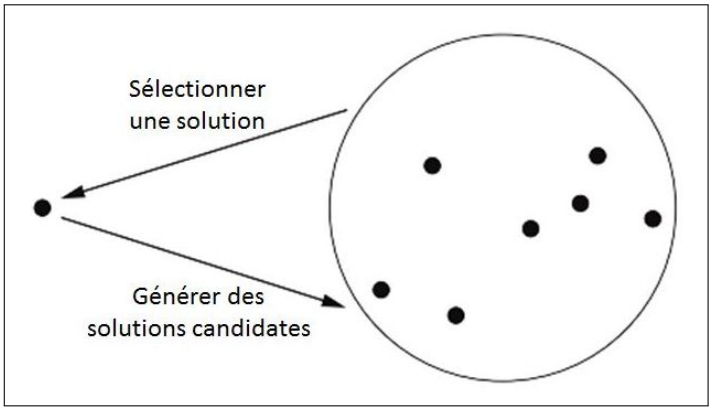

FIGURE 6.5 – Principes généraux d'une métaheuristique à base de solution unique

– *Les méthodes évolutives / à population de solutions :* Ces algorithmes, contrairement aux méthodes à solution unique, travaillent avec un ensemble de solutions en même temps, que la méthode fait évoluer graduellement. L'utilisation de plusieurs solutions simultanément permet d'améliorer l'exploration de l'espace des configurations (figure 6.6). Dans cette seconde catégorie, on recense : les algorithmes génétiques, les algorithmes à estimation de distribution et les algorithmes par colonies de fourmi. Ces métaheuristiques sont souvent plus gourmandes en calculs que les méthodes à solution unique.

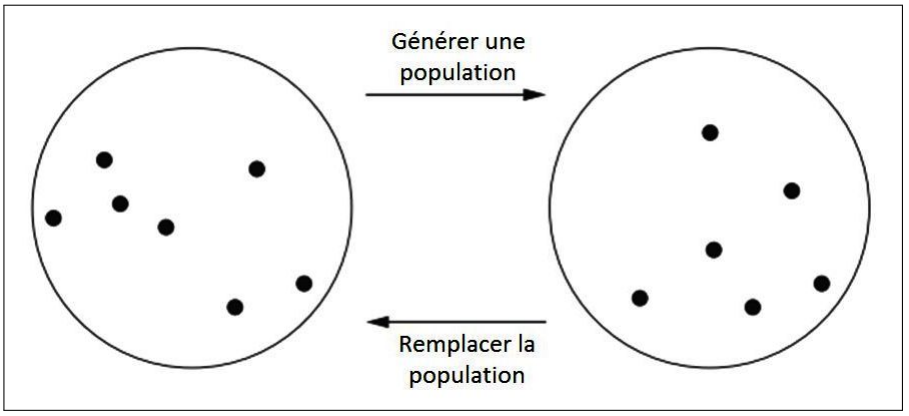

FIGURE 6.6 – Principes généraux d'une métaheuristique à population de solutions

Dans les sections suivantes nous développons plus en détail les caractéristiques de chacune de ces différentes classes de métaheuristiques.

6.2.3/ MÉTHODES CONSTRUCTIVES

Les méthodes constructives produisent des solutions admissibles en partant d'une solution initiale vide et en insérant à chaque étape une composante dans la solution partielle courante. Cette décision n'est jamais remise en question par la suite. La majorité des

méthodes constructives sont de type glouton. A chaque étape, la solution courante est complétée de la meilleure façon possible sans tenir compte de toutes les conséquences que cela entraîne au niveau du coût de la solution finale. Dans ce sens, les méthodes gloutonnes sont souvent considérées comme myopes (Widmer, 2001).

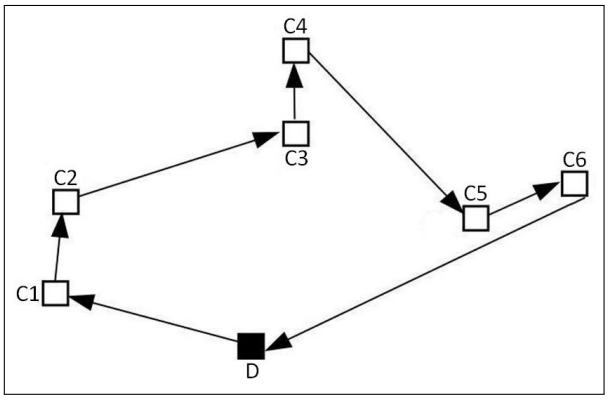

FIGURE 6.7 – Tournée du voyageur de commerce

Une méthode constructive peut être utilisée par exemple pour résoudre le problème d'un voyageur de commerce qui doit rendre visite à n clients. On peut ici proposer un algorithme glouton suivant : en partant d'un point de départ D, le voyageur de commerce va chez le client le plus proche (ici C_1). En quittant C_1, il va chez le client le plus proche de C_1 qu'il n'a pas encore rencontré. Ce processus est répété impérativement jusqu'à ce le voyageur ait visité tous ces clients. En quittant le dernier client (C_n), il rentre chez lui, formant ainsi un cycle hamiltonien. Il a ainsi construit la tournée : $D-C_1-C_2-\ldots-C_n-D$ (figure 6.7). Le dernier tronçon $C_n - D$ est souvent non optimal et peut ainsi nuire à la qualité globale de la solution.
Les méthodes constructives se distinguent par leur rapidité à générer une solution finale admissible sans avoir recours à des techniques hautement sophistiquées. Cependant, la solution finale trouvé n'est pas optimale. Aujourd'hui les méthodes constructives servent essentiellement à produire des solutions initiales à améliorer par la suite par des métaheuristiques de type recherche locale ou évolutives.

6.2.4/ MÉTHODES DE RECHERCHE LOCALE / À SOLUTION UNIQUE

Les méthodes de recherche locale sont des algorithmes itératifs qui explorent l'espace des solutions X en se déplaçant pas à pas d'une solution à une autre. Une méthode de ce type débute à partir d'une solution $s_0 \in X$ choisie arbitrairement ou alors obtenue par le biais d'une méthode constructive. Le passage d'une solution admissible à une autre se fait sur la base d'un ensemble de modifications élémentaires qu'il s'agit de définir au cas par cas. Une solution s' est obtenue à partir de s en appliquant une modification élémentaire. Le voisinage $V(s)$ d'une solution $s \in X$ est défini comme l'ensemble des

solutions admissibles atteignables depuis s en effectuant une modification élémentaire. Un tel processus d'exploration est interrompu lorsqu'un ou plusieurs critères d'arrêt sont satisfaits.

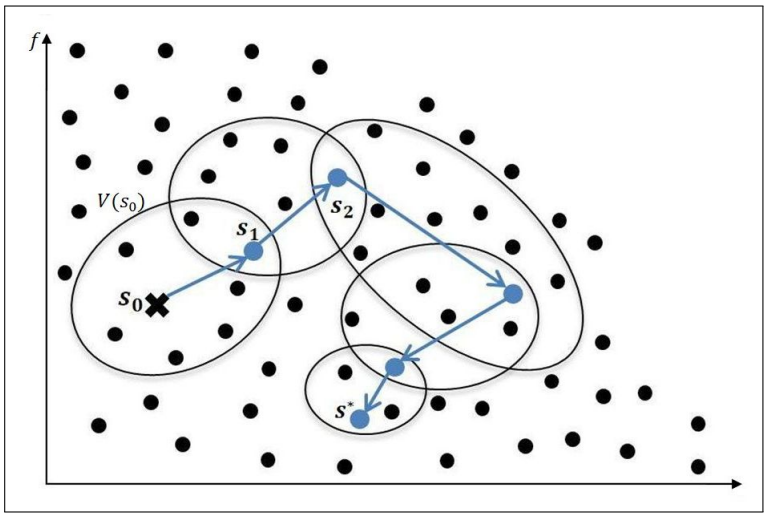

FIGURE 6.8 – Exploration de l'espace des solutions par la méthode de recherche locale

La figure 6.8 illustre la méthode d'exploration de l'espace des solutions par une méthode de recherche locale. Les passages successifs d'une solution à une solution voisine définissent un chemin au travers de l'espace des solutions admissibles. La modélisation d'un problème d'optimisation et le choix du voisinage doivent être effectués de telle sorte qu'il existe au moins un chemin entre la première solution s_0 et une solution optimale s^*. De nombreuses métaheuristiques ont été élaborées dans le cadre de la méthode de recherche locale.

Nous décrivons dans ce qui suit les méthodes de recherche locale les plus utilisées.

6.2.4.1/ LE HILL-CLIMBING

La méthode de recherche locale Hill-Climbing est proche de la méthode gradient (Cauchy, 1847). Le principe est de partir d'une première solution calculée aléatoirement, puis de se déplacer dans l'espace des solutions dans la direction de la plus forte pente : De nouvelles solutions candidates sont testées itérativement dans le voisinage du candidat courant, et elles sont adoptées si elles sont meilleures que la courante. Ce procédé est répété aussi longtemps que la valeur de la fonction objectif f augmente. La recherche s'interrompt dès qu'un optimum local de f est atteint. L'amélioration de la solution courante se base sur une approche stochastique (partiellement aléatoire) pour chercher de meilleurs candidats.
L'algorithme général de la méthode Hill-Climbing se présente comme suit :

La méthode Hill-Climbing a considéré parmi les méthodes heuristiques les plus populaires pour traiter les problèmes d'optimisation combinatoire. Toutefois, elle comporte

Algorithm 2 : Algorithme de Hill-Climbing

$s \leftarrow$ valeur aléatoire initiale
$Change = 0$
repeat
  $BestVoisin \leftarrow s$
  for $\forall s' \in V(s)$ solution voisine de s **do**
    if $f(s') > f(BestVoisin)$ **then**
      $BestVoisin \leftarrow s'$
    end if
  end for
  if $f(BestVoisin) > f(s)$ **then**
    $s \leftarrow BestVoisin$
    $Change = 1$
  end if
until $Change == 0$ ou le temps imparti est atteint
return s

deux obstacles majeurs qui limitent considérablement son efficacité :

– Suivant la taille et la structure du voisinage $V(s)$ considéré, la recherche de la meilleure solution voisine est un problème qui peut être aussi difficile que le problème initial.
– Une méthode Hill-Climbing est incapable de progresser au-delà du premier optimum local rencontré. Or les problèmes d'optimisation combinatoire comportent typiquement de nombreux optima locaux pour lesquels la valeur de la fonction objectif peut être fort éloignée de la valeur optimale.

Pour faire face à ces carences, des méthodes de recherche plus sophistiquées ont été développées au cours de ces dernières années. Ces méthodes acceptent des solutions voisines moins bonnes que la solution courante afin d'échapper aux minima locaux de la fonction f. En règle générale, seule une portion du voisinage courant est explorée à chaque étape.
Les méthodes que nous présentons dans les sections suivantes sont en général beaucoup plus performantes qu'une simple méthode Hill-Climbing, mais également beaucoup plus coûteuses en termes de ressources informatiques. Leur mise en œuvre doit généralement tenir compte du temps de réponse maximal autorisé par l'utilisateur du programme. Il convient aussi de signaler qu'un effort non négligeable est nécessaire pour ajuster convenablement les paramètres qu'elles font intervenir dans le but de guider efficacement la recherche au travers de l'ensemble des solutions X.

6.2.4.2/ LE RECUIT SIMULÉ

Le recuit simulé est un algorithme inspiré de l'industrie. Il est dérivé de l'algorithme Metropolis développé lors du projet Manhattan (Metropolis et al., 1953) (Aarts and Korst, 1989). Le projet a abouti à un algorithme permettant de simuler l'évolution d'un système

physique instable vers un état d'équilibre thermique. Plus précisément, cette méthode retrouve ses origines dans l'étude du recuit des métaux en métallurgie : on porte initialement le métal à très haute température, puis on le refroidit petit à petit. Lorsque la température diminue, les mouvements des atomes deviennent de moins en moins aléatoires, et le système aura tendance à se trouver dans des états à basse énergie. Un métal refroidi trop vite présente de nombreux défauts microscopiques, c'est l'équivalent d'un optimum local pour un problème d'optimisation combinatoire. Si on refroidit le métal lentement, les atomes se réarrangent, les défauts disparaissent, et le métal a alors une structure très ordonnée, équivalente à un optimum global.
L'idée du recuit simulé consiste à utiliser l'algorithme de Metropolis pour des valeurs décroissantes de la température t. Le refroidissement progressif d'un système de particules est simulé en faisant une analogie entre l'énergie du système et la fonction objectif du problème d'une part, et entre les états du système et les solutions admissibles du problème d'autre part.
Au début de l'algorithme, un paramètre t, apparenté à la température, est déterminé et décroît tout au long de l'algorithme pour tendre vers une température de gélification t_{gel}. Dans l'algorithme de Métropolis, on part d'une configuration initiale donnée, et on fait subir au système une modification élémentaire. Si cette perturbation a pour effet d'améliorer l'énergie E du système ($\triangle E < 0$), elle est acceptée. Sinon ($\triangle E \geqslant 0$), elle est acceptée avec une probabilité non nulle P :

$$P(t, \triangle E) = exp(\frac{- \triangle E}{t}). \tag{6.9}$$

En appliquant itérativement cette règle, on engendre une séquence de configurations qui tendent vers l'équilibre thermodynamique. A chaque étape, l'acceptation d'un nouvel état dont l'énergie n'est pas inférieure à celle de l'état courant est décidée en générant de manière aléatoire un nombre $r \in [0, 1]$. Si r est inférieur ou égal à $P(t, \triangle E)$, alors le nouvel état est accepté. Autrement l'état courant est maintenu. (Metropolis et al., 1953) ont montré que l'utilisation répétée d'une telle règle fait évoluer le système vers un état d'équilibre thermique. La répétition de cette règle dépend de la température de gélification.
Ainsi, par rapport à la méthode Hill-Climbing, l'algorithme du recuit simulé varie dans sa manière de choisir quand remplacer une solution s par une solution voisine s'. Si s' est meilleur que s, s' remplace toujours s, mais si s' est moins bonne que s, on ne peut remplacer s par s' qu'à une certaine probabilité P :

$$P(t, \triangle f) = exp(\frac{- \triangle f}{t}) \text{avec } \triangle f = f(s') - f(s) \tag{6.10}$$

L'algorithme général peut se représenter avec le pseudo-code suivant :

Algorithm 3 : Algorithme du recuit simulé

$s \leftarrow$ valeur aléatoire initiale
$t \leftarrow$ température, initialement une grande valeur
while $t > t_{gel}$ **do**
  Obtenir une solution voisine $s' \in V(s)$ par transformation élémentaire de s
  if $f(s') > f(s)$ **then**
    $s \leftarrow s'$
  else
    Générer un nombre réel aléatoire r dans $[0, 1]$
    if $r < P(t, \triangle f)$ **then**
      $s \leftarrow s'$
    end if
  end if
  Décroitre t
end while
return s

La figure 6.9 montre un exemple de variation de l'énergie de la combinaison courante au cours du temps. Au début, l'amplitude des variations de l'énergie est grande, car de nombreuses solutions sont acceptées étant donné la température du système. Puis, cette amplitude diminue au fil du temps, dû au refroidissement du système. Enfin, lorsqu'il n'y a plus de variation de cette énergie ($t_{gel} = 0$), le système est gelé et la meilleure solution trouvée est renvoyée.

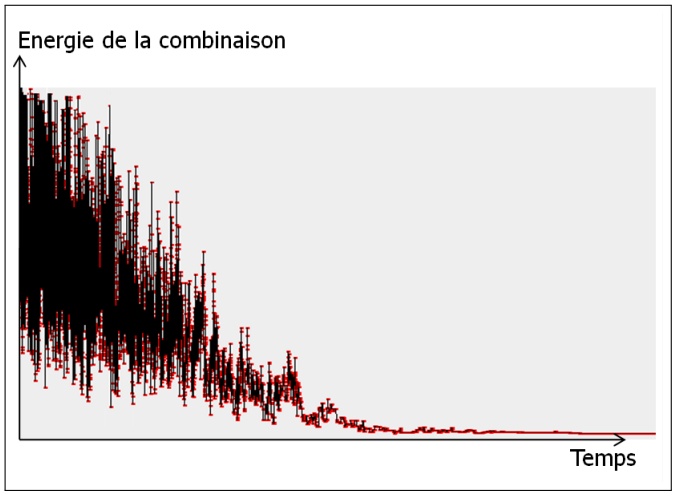

FIGURE 6.9 – Variation de l'énergie par l'algorithme de recuit simulé

La performance du recuit simulé dépend, entre autres, de la règle de refroidissement (c'est à dire la décroissance du paramètre t : (1) Un refroidissement trop rapide mènerait vers un optimum local pouvant être de très mauvaise qualité. (2) Un refroidissement trop lent serait très coûteux en temps de calcul : On constate qu'il faut une diminution de la température suffisamment lente, donc l'algorithme nécessite un certain nombre d'itérations pour voir la solution s'améliorer. Corrélativement, le temps de calcul devient

parfois excessif avec certaines applications.
Pour le bon fonctionnement de l'algorithme, un bon réglage des différents paramètres (température initiale, nombre d'itérations par palier de température et décroissance de la température) est nécessaire.

6.2.4.3/ LA MÉTHODE TABOU

Les principes de méthode tabou ont été proposés pour la première fois dans les années 80 par Fred Glover dans les années 80 (F. Glover, 1986). A l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine à chaque itération, la méthode tabou examine un échantillonnage de solutions voisines $V(s)$ et retient la meilleure solution voisine s' dans $V(s)$ même si $f(s') > f(s)$. De cette manière la recherche tabou ne s'arrête pas au premier optimum trouvé. Seulement lorsque $f(s') < f(s)$ l'algorithme risque de revenir à s immédiatement.
Pour éviter de revenir à chaque fois à la solution s, on crée une liste T appelée liste tabou, qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Le but est de donner assez de temps à l'algorithme pour lui permettre de sortir d'un optimum local.
Seulement, en s'interdisant de cette manière certaines permutations, un mouvement qui aboutirait à une solution meilleure que toutes celles déjà visitées risquerait d'être écarté. Pour éviter cela, une idée consiste à incorporer dans l'algorithme des critères d'aspiration, qui autorisent certains mouvements, bien qu'interdits par la liste tabou, parce qu'on suppose qu'ils vont améliorer la recherche. Généralement le critère d'aspiration consiste à autoriser une solution s si sa valeur $f(s)$ est meilleure que toutes les solutions rencontrées jusqu'ici.
Une condition d'arrêt de l'algorithme doit être définie. En général, on se donne un nombre maximum Nb_{max} d'itérations entre deux améliorations de la meilleure solution s^* rencontrée.
Soit $V_T(s)$ toutes les solutions qui ne sont pas taboues, et les solutions qui sont tabous mais dont le statut tabou a été levé par le critère d'aspiration : $V_T(s) = \{s' \in V(s)$ tel que $s' \notin T$ ou $f(s') < f(s*)\}$.
L'algorithme général de la méthode tabou peut être présenté avec le pseudo-code suivant :

La structure de l'algorithme de base de la méthode tabou est assez proche de celle du recuit simulé. L'avantage, par rapport au recuit simulé est le paramétrage plus simple de l'algorithme. En revanche, la méthode tabou présente un inconvénient par rapport aux stratégies de mémorisation des solutions tabous. Cette méthode devient de plus en plus lourde à mesure que l'on voudra raffiner le procédé de mémorisation des solutions, en mettant en place des stratégies plus complexes.

Algorithm 4 : Algorithme de la méthode tabou

$s^* \leftarrow$ valeur aléatoire initiale
$T \leftarrow \emptyset$
t taille de la liste tabou
Nb_{max} nombre maximum d'itérations
k itération courante
repeat
  Choisir s' la meilleure solution dans $V_T(s)$
  if $f(s') < f(s^*)$ **then**
    if $k < t$ **then**
      $s^* \leftarrow s'$
      Mettre à jour T
      $k \leftarrow k + 1$
    end if
  end if
  $s \leftarrow s'$
until Nb_{max} est atteint
return s

6.2.4.4/ LA RECHERCHE À VOISINAGE VARIABLE

La recherche à voisinage variable (Variable Neighborhood Search) est une métaheuristique récente pour la résolution de problèmes d'optimisation combinatoire, dont les premières descriptions ont été faites par (Mladenovic and Hansen, 1997). L'idée de base est de définir cette fois ci, non pas un, mais plusieurs voisinages possibles pour une solution. Cette méthode de recherche est motivée par trois principes :

– Un minimum local par rapport à un voisinage n'est pas forcément un minimum local par rapport à un autre.
– Un minimum global est un minimum local par rapport à tous les voisinages possibles.
– Pour de nombreux problèmes, les minimaux locaux par rapport à un ou à plusieurs voisinages sont relativement proches les uns des autres.

La méthode de base de la recherche à voisinage variable fonctionne en plusieurs étapes décrite par l'algorithme décrit comme suit :
Tout d'abord l'algorithme nécessite la définition de différentes structures de voisinage, c'est-à-dire un ensemble d'éléments noté $\{V_1, \ldots, V_k, \ldots, V_{max}\}, k \in \{1, \ldots, max\}$, généralement de portées k différentes, ordonnés par ordre bien défini. Cet ordre peut correspondre à une imbrication de voisinages, $V_1 \subset \ldots \subset V_k \subset \ldots \subset V_{max}$, ou simplement correspondre à la taille des voisinages : V_1 est alors le plus petit et V_{max} le plus grand, $V_1 < \ldots < V_k < \ldots < V_{max}$.
A partir d'une solution de départ s choisie aléatoirement, un voisin s' de s est sélectionné en utilisant la structure de voisinage V_1. Si une solution, noté s'', obtenue à partir de s' suite à l'application d'une méthode de recherche locale est meilleure que la solution s',

alors la recherche reprend à partir de cette solution. Sinon, une solution est choisie en utilisant un voisinage de portée supérieure, soit ici le voisinage V_2.
Cette méthode permet de sortir d'un optimum local si celui-ci est un optimum local pour une portée de voisinage plus petite que la valeur max. La recherche locale effectuée à chaque étape de l'algorithme peut être une simple méthode de descente, ou bien une méthode plus sophistiquée, comme le recuit simulé ou la méthode tabou. Une condition d'arrêt doit être définie. En général, on se donne un nombre maximum Nb_{max} d'itérations.
Le fait d'utiliser plusieurs voisinages permet de diversifier l'exploration de l'espace des solutions afin d'accéder à un plus grand nombre de régions intéressantes, ce qui conduit à une méthode plus robuste que le recuit simulé ou la recherche tabou.
L'algorithme général de la recherche à voisinage variable est décrit par le pseudo-code ci- dessous :

Algorithm 5 : Algorithme de recherche à voisinage variable

$s \leftarrow$ valeur aléatoire initiale
Nb_{max} nombre maximum d'itérations
$k \leftarrow 1$ où k itération courante
max portée maximale de voisinage
while $k < max$ **do**
  repeat
    Choisir s' aléatoirement dans $V_k(s)$
    Choisir s'' par une méthode de recherche locale à partir de s' dans $V_k(s)$
    if $f(s'') > f(s')$ **then**
      $s \leftarrow s''$
      Retourner au premier voisinage $V_1, k \leftarrow 1$
    end if
  until le nombre maximum d'itérations Nb_{max} est atteint
  $k \leftarrow k + 1$
end while
return s

6.2.5/ MÉTHODES ÉVOLUTIVES / À POPULATION DE SOLUTION

La méthode d'optimisation à base de stratégies d'évolution (SE) fut initialement proposée par (Rechenberg, 1965). Les méthodes évolutives aussi appelées méthodes à populations de solutions, font évoluer une population d'individus selon des règles bien précises. Ces méthodes s'inspirent généralement de la nature, et notamment du règne animal (Fogel, 1999). Cette analogie a donné lieu à plusieurs approches, parmi lesquelles les algorithmes génétiques, les algorithmes à estimation de distribution, les algorithmes de colonies de fourmis, etc.
Contrairement aux méthodes constructives et de recherche locale qui font intervenir une solution unique, les méthodes évolutives manipulent un groupe de solutions admissibles à chacune des étapes du processus de recherche. L'idée centrale consiste à utiliser

régulièrement les propriétés collectives d'un ensemble de λ solutions distinguables, appelé population, dans le but de guider efficacement la recherche vers de bonnes solutions dans l'espace X.
Chaque solution est liée à une valeur de la fonction objectif du problème, nommée *fitness*. Les algorithmes évolutionnaires font évoluer cette population par générations successives, en utilisant des opérateurs inspirés de la théorie de l'évolution ; la sélection, la reproduction et la mutation :
Après avoir généré une population initiale de solutions, aléatoirement ou par l'intermédiaire d'une méthode constructive, l'algorithme supprime toutes les solutions de la population sauf les μ plus forts, afin de transmettre les meilleurs μ solutions de la génération courante à la nouvelle génération (sélection). La sélection augmente la performance d'une population en retenant les meilleures solutions admissibles afin de guider la recherche vers des régions prometteuses. L'étape suivante consiste à faire croiser les solutions de μ, en combinant leurs caractéristiques, pour créer de nouvelles solutions. Ce croisement utilise ρ ($\rho \geq 2$) parents choisis aléatoirement pour créer un seul individu enfant. Au total, λ enfants sont créés (reproduction). Lors de la création d'une nouvelle solution certaines caractéristiques transmises par les parents peuvent changer aléatoirement avec une certaine probabilité (mutation). En troisième étape la population de la nouvelle génération est mise à jour en appliquant une sélection, soit uniquement aux enfants, soit à l'ensemble enfants + parents.
La sélection des solutions de la nouvelle population consiste à remplacer les parents par les μ meilleurs enfants. Dans ce cas l'algorithme est appelé (μ, λ)-SE. Dans le cas d'une sélection sur l'ensemble enfants + parents, on sélectionne les meilleurs μ individus parmi la population des parents et des enfants de la population courante pour créer la population parent de la nouvelle génération. L'algorithme est appelé $(\mu + \lambda)$-SE. Enfin, la procédure se réitère.

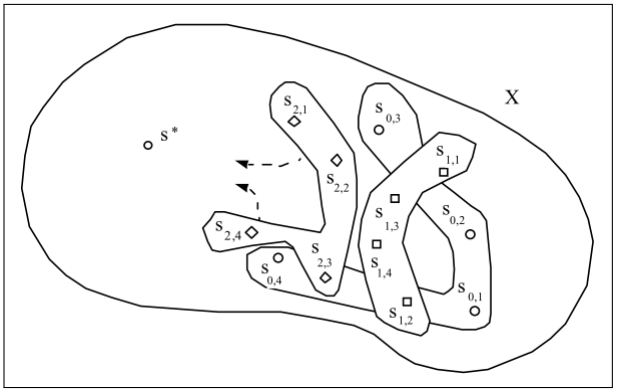

FIGURE 6.10 – Exploration par une approche évolutive

Le mécanisme de recherche de la méthode évolutive est représenté dans la figure 6.10.
La stratégie évolutive procède généralement suivant l'algorithme décrit par le pseudo-code ci-dessous :

Algorithm 6 : Algorithme de la stratégie évolutive

μ nombre de parents sélectionnés
λ nombre d'enfants générés par les parents
ρ nombre de parents à utiliser pour la reproduction
$P \leftarrow \{\}$ population initialement vide
$Best \leftarrow \emptyset$ meilleure solution
//Construire une population initiale de λ solutions
repeat
  $P \leftarrow P \cup \{$ une nouvelle solution choisie aléatoirement $\}$
until le nombre maximum λ de solutions à choisir atteint
repeat
  for $P_i \in P$ **do**
    Calculer $Fitness(P_i)$
    if $Best = \emptyset$ ou $Fitness(P_i) > Fitness(Best)$ **then**
      $Best \leftarrow P_i$
    end if
  end for
  //Sélection
  $F \leftarrow$ sélection de μ solutions de P dont les $Fitness()$ sont les meilleures
  $P \leftarrow \{\}$ dans le cas d'un algorithme $(\mu, \lambda) - SE$
  //ne pas vider P dans le cas d'un algorithme $(\mu + \lambda)$-SE
  for μ/ρ **do**
    $Enfant \leftarrow Croisement(\rho, F)$ //Reproduction
    $P \leftarrow P \cup \{Mutation(Enfant)\}$ //Mutation
  end for
  $P \leftarrow$ sélection de μ meilleures solutions
until $Best$ est la solution idéale ou le temps total imparti est atteint
return $Best$

Parmi les algorithmes évolutionnaires, la sous-classe d'algorithmes la plus populaire est les algorithmes génétiques que nous décrivons brièvement dans la section suivante.

6.2.5.1/ LES ALGORITHMES GÉNÉTIQUES

Les algorithmes génétiques sont des stratégies d'évolution qui s'inspirent fortement des mécanismes biologiques liés aux principes de sélection et d'évolution naturelle. Développés initialement par (Holland, 1975) pour répondre à des besoins spécifiques en biologie, les algorithmes génétiques ont été adaptés à des contextes très variés.
Le vocabulaire utilisé est le même que celui des stratégies d'évolution et de la génétique, on emploie le terme individu (solution potentielle), population (ensemble de solutions),

génotype (une représentation de la solution), chromosome (une partie du génotype), parent, enfant, sélection, reproduction, croisement, mutation, etc. Le fonctionnement des algorithmes génétiques est identique aux stratégies d'évolution (μ, λ)-SE. La principale différence est la façon dont la reproduction et la mutation sont faites :
L'opérateur de croisement a pour objectif de recombiner les chromosomes d'une paire d'individus sélectionnés (parents avec $\rho = 2$), afin de créer une nouvelle paire d'individus (enfants) qui héritent de certaines caractéristiques de leurs parents avec une probabilité p_c. Le croisement est mis en place pour que les nouveaux chromosomes gardent la meilleure partie des chromosomes anciens.
La mutation permet de transformer avec une probabilité p_m le codage d'un individu afin d'apporter une certaine diversité dans la population et empêcher que celle-ci converge trop vite vers un seul type d'individu parfait, incapable de sortir d'un minimum local.
La sélection est un opérateur essentiel pour améliorer la qualité d'une population. Son objectif est de retenir et multiplier les meilleurs individus qui contribueront à l'opération de croisement, et d'éliminer les plus mauvais, en accord avec la théorie de Darwin. Dans la littérature différentes méthodes de sélection sont proposées (que nous ne développons pas ici), comme celle par tournoi, par roulette, par rang, etc. (Harries and Smith, 1998) (Haupt and Haupt, 2004).
L'algorithme génétique est guidé par un certain nombre de paramètres fixés à l'avance. Par rapport à la stratégie évolutive, l'algorithme génétique fait intervenir deux nouveaux paramètres :

- La probabilité de croisement p_c : Elle dépend de la forme de la fonction de Fitness. Plus elle est élevée, plus la population subit des changements importants.
- La probabilité de mutation p_m : Ce taux est généralement faible, puisqu'un taux élevé risque de conduire à une solution sous-optimale, et à la perte de la population originale.

L'efficacité des algorithmes génétiques dépend fortement du réglage des différents paramètres caractérisant ces algorithmes, et qui sont parfois difficiles à déterminer.
Le pseudo-code de l'algorithme génétique est décrit ci-dessous :

Algorithm 7 : Algorithme génétique

t taille de la population désirée
$P \leftarrow \{\}$ population initialement vide
$Best \leftarrow \emptyset$ meilleure solution
repeat
  $P \leftarrow P \cup \{$ une nouvelle solution choisie aléatoirement $\}$
until le nombre maximum t de solutions à choisir atteint
repeat
  for $P_i \in P$ **do**
    Calculer $Fitness(P_i)$
    if $Best = \emptyset$ ou $Fitness(P_i) > Fitness(Best)$ **then**
      $Best \leftarrow P_i$
    end if
  end for
  $Q \leftarrow \{\{$
  for $t/2$ **do**
    $Parent_1, Parent_2 \leftarrow SelectionParent(P)$
    $Enfant_1, Enfant_2 \leftarrow Coisement(Parent_1, Parent_2)$ avec une probabilité p_c
    $Mutation_1 \leftarrow Mutation(Enfant_1)$ avec une probabilité p_m
    $Mutation_2 \leftarrow Mutation(Enfant_2)$ avec une probabilité P_m
    $Q \leftarrow Q \cup \{Mutation_1, Mutation_2\}$
  end for
  $P \leftarrow Q$
  $P \leftarrow Selection(P)$ des meilleures solutions pour la nouvelle génération
until $Best$ est la solution idéale ou le temps total imparti est atteint
return $Best$

6.2.5.2/ LES ALGORITHMES À ESTIMATION DE DISTRIBUTION

Les algorithmes à estimation de distribution EDA (Estimation of Distribution Algorithm) ont été conçus comme une variante des algorithmes évolutionnaires. Ils ont été introduits pour la première fois par (Baluja, 1994) puis repris par dans les travaux de (Mühlenbein and Paass, 1996). Les algorithmes à estimation de distribution reprennent les principes des algorithmes évolutionnaires mais utilisent des modèles probabilistes à la place des opérateurs de mutation et de croisement pour construire de nouveaux individus. Un modèle probabiliste ou la distribution de probabilité est estimée à chaque itération à partir des informations apportées par la précédente génération. L'estimation de la distribution de probabilité permet d'estimer les relations entre les différents individus.
Le principe général d'un algorithme à estimation de distribution est similaire à un algorithme évolutionnaire. On initialise tout d'abord une population P_0. Généralement la

population initiale est générée de façon aléatoire, et seules les meilleures combinaisons générées sont gardées. Ensuite, à l'itération k, un modèle probabiliste M est construit à partir des informations de la population P_{k-1}. Différents types de modèles probabilistes, plus ou moins simples, peuvent être considérés. Le modèle le plus simple, appelé PBIL (Baluja, 1994), est basé sur la probabilité d'apparition de chaque composant sans tenir compte d'éventuelles relations de dépendances entre les composants. Dans ce cas, on calcule pour chaque composant sa fréquence d'apparition dans la population et on définit la probabilité de sélection de ce composant proportionnellement à sa fréquence. Enfin, la nouvelle population P_k est créée aléatoirement à partir du modèle de distribution de probabilité M, généralement selon un principe glouton.
Le parallèle avec les algorithmes évolutionnaires est ici respecté, le couple croisement-mutation étant remplacé par l'estimation de la distribution.
Cette méthode est résumée par l'algorithme ci-dessous :

Algorithm 8 : Algorithme à estimation de distribution

$P_0 \leftarrow$ générer aléatoirement une population initiale
k itération courante
while critère d'arrêt n'est pas atteint **do**
  $k \leftarrow k + 1$
  Construction d'un modèle probabiliste M en fonction de P_{k-1}
  Génération de nouvelles combinaisons à l'aide de M
  Mise à jour de P_{k-1} en fonction des nouvelles combinaisons
end while
return la meilleure combinaison ayant appartenu à la dernière population

La principale difficulté de cette méthode est le choix et l'estimation de la distribution de probabilité. En effet, il faut choisir au préalable quel modèle de distribution va être utilisé par l'algorithme et à chaque itération estimer cette distribution.

6.2.5.3/ LES ALGORITHMES DE COLONIES DE FOURMIS

Les algorithmes de colonies de fourmis ACO (Ant Colony Optimization) ont été proposés par (Colorni et al., 1992) et appliqués la première fois au problème du voyageur de commerce (Dorigo and Gambardella, 1997). Il existe un parallèle entre l'optimisation par colonies de fourmis et les algorithmes par estimation de distribution. Ces deux approches utilisent un modèle probabiliste glouton pour générer des combinaisons, ce modèle évoluant en fonction des combinaisons précédemment construites dans un processus itératif d'apprentissage.
L'originalité et la contribution essentielle des algorithmes de colonies de fourmis est de s'inspirer du comportement collectif des fourmis pour faire évoluer le modèle probabiliste : Les fourmis utilisent pour communiquer une substance chimique volatile appelée phéromone. En quittant leur nid pour explorer leur environnement à la recherche de nourriture, les fourmis arrivent à élaborer des chemins qui s'avèrent fréquemment être

les plus courts pour aller du nid vers une source de nourriture. Chaque fourmi dépose alors une quantité de phéromones sur ces pistes qui deviendront un moyen de communication avec leurs congénères. Cette quantité de phéromone évolue par la conjugaison de deux mécanismes : un mécanisme de renforcement des traces de phéromone associées aux composants des meilleures chemins, visant à augmenter la probabilité de sélection de ces composants ; et un mécanisme d'évaporation, visant à privilégier les expériences récentes par rapport aux expériences plus anciennes. Les fourmis choisissent ainsi avec une probabilité élevée les chemins contenant les plus fortes concentrations de phéromones.

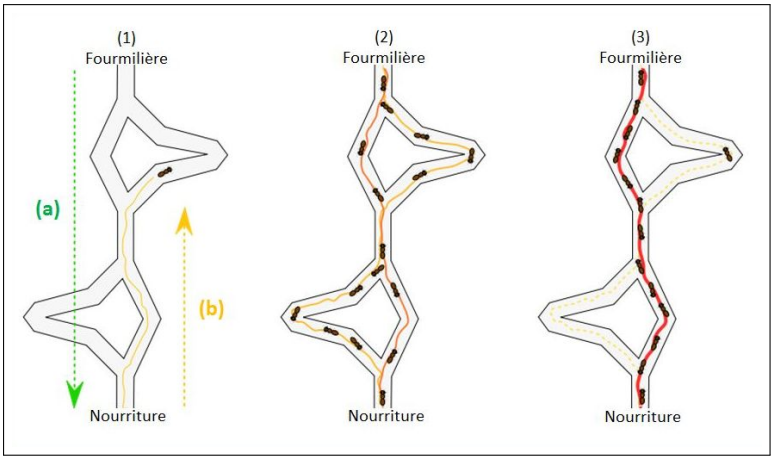

FIGURE 6.11 – Comportement d'une colonie de fourmis

La figure 6.11 illustre le comportement d'une colonie de fourmis : (1) La première fourmi trouve la source de nourriture, via un chemin quelconque (a), puis revient à la fourmilière en laissant derrière elle une piste de phéromone (b). (2) Les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement de la piste rend plus attractif le chemin le plus court. (3) Les fourmis empruntent le chemin le plus court, les portions longues des autres chemins perdent leur piste de phéromones.
Le principe général du fonctionnement de l'algorithme est décrit ci-dessous :

Algorithm 9 : Algorithme de colonies de fourmis

Initialiser les traces de phéromone à τ_{init}
while critère d'arrêt n'est pas atteint **do**
  Constructions de solutions (chemins) par les fourmis par construction gloutonne en tenant compte des phéromones
  Mise à jour des phéromones
  Mettre à jour la meilleure solution trouvée
end while
return la meilleure solution

L'algorithme de colonies de fourmis a été à l'origine surtout utilisé pour produire des solutions quasi-optimales au problème du voyageur de commerce, puis, plus généralement, aux problèmes d'optimisation combinatoire. Depuis ses débuts son emploi s'est étendu à plusieurs domaines, tel que dans le cadre de l'intelligence en essaim (Bonabeau et al., 1999) ou dans le cadre de l'optimisation continue (Dréo and Siarry, 2004).

6.3/ IMPLÉMENTATION DES MÉTAHEURISTIQUES

Dans cette partie du chapitre nous décrivons les différentes méthodes de recherche locale que nous avons utilisé pour la résoudre notre problème d'optimisation de parcours de formation en m-learning. Nous commençons par décrire l'algorithme glouton utilisé pour construire la solution initiale s_0.

6.3.1/ CONSTRUCTION D'UNE SOLUTION INITIALE PAR UN ALGORITHME DE TYPE GLOUTON

Un algorithme glouton est un algorithme qui suit le principe de faire, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global sans retour arrière. Nous passons en entrée un graphe G de LO pondérés représentant l'ensemble des parcours de formations possibles. L'algorithme commence son exploration à partir d'un LO de départ, $lo_{init} \in D$.
L'algorithme glouton doit construire pas à pas un chemin dans le graphe G. Il doit aussi s'assurer de construire un chemin qui passe par tous les LO $Objectif$ de la formation. Les LO $Objectif$ du graphe G ne respectant pas un ordre d'apprentissage particulier, l'algorithme va choisir aléatoirement le prochain objectif à réaliser dans le graphe G. Ainsi à chaque étape, l'algorithme choisit par une heuristique le prochain objectif aléatoirement dans l'ensemble des objectifs, puis il calcule le plus court chemin pour atteindre cet objectif, par un algorithme de type Dijkstra (en théorie des graphes, l'algorithme de Dijkstra est un algorithme traditionnel mis à jour par (Dijkstra, 1971) qui sert à résoudre le problème du plus court chemin dans un graphe).
L'algorithme s'arrête lorsque $lo_{fin} \in OBJ$ est atteint.
L'algorithme glouton est le suivant :

Algorithm 10 : Algorithme de construction de la solution s_0 avec une méthode gloutonne

Input :
Un graphe orienté G de LO pondérés
L'ensemble OBJ des LO de type objectif
Un LO de départ $lo_{init} \in D$ et un LO fin $lo_{fin} \in OBJ$

Output :
Une solution s_0 : une liste ordonnée de LO à suivre pour valider tous les objectifs

$s_0 = lo_{init}$
$lo_{courant} = lo_{init}$
$ObjNonValides = OBJ$ ensemble des LO objectif non validés
O LO de type objectif
while $lo_{courant} != lo_{fin}$ **do**
  $O = RechercherAleatoireObj(G, lo_{courant}, ObjNonValides)$
  $s'_0 = Dijkstra(G, lo_{courant}, O)$
  $s_0 = s_0 \cup s'_0$
  Mettre à jour $ObjNonValides = ObjNonValides \setminus \{O\}$
  Mettre à jour $lo_{courant}$
end while
Return s_0

Algorithm 11 : Fonction Dijkstra (G, lo_{init}, lo_{fin})

Input :
Un graphe orienté G de LO pondérés
lo_{init} LO de départ, lo_{fin} LO d'arrivé
Output :
s'_0 : une liste ordonnée de LO formant le plus court chemin entre lo_{init} et lo_{fin}
$dist$ tableau des distance entre les LO

for $i = 0$ to $|LO|$ **do**
  //on initialise les sommets autres que sdeb lo_{init} à infini infini
  $dist[lo_i] = \infty$
end for
// lo_{init} étant le point le plus proche de lo_{init} la distance est égale à 0
$dist[lo_{init}] = 0$
F = Ensemble de tous les LO non visités de type non objectif
while $F != \emptyset$ **do**
  //Trouver le LO dans F avec $dist$ le plus petit
  $lo1 = TrouverMin(F)$
  $F = F \setminus \{lo1\}$
  for chaque $lo2_{suivant}$ de $lo1$ **do**
    if $dist[lo2] > dist[lo1] + Pert(lo1)$ **then**
      $dist[lo2] = dist[lo1] + Pert(lo1)$
      $lo2_{precedent} = lo1$ //dit que pour aller à $lo2$ il faut passer par $lo1$
    end if
  end for
end while
$s'_0 = \emptyset$

$lo_{courant} = lo_{fin}$
while $lo_{courant} != lo_{init}$ **do**
  $s'_0 = Ajouter(s'_0, lo_{courant})$
  $lo_{courant} = lo_{precedent}$
end while
$s'_0 = Ajouter(s'_0, lo_{init})$
Return s'_0

6.3.2/ Amélioration de recherche locale par un algorithme Hill-Climbing

La méthode de recherche locale la plus simple est la méthode Hill-Climing. Le principe est très simple : on construit une solution initiale dans notre cas avec la méthode gloutonne, puis on mute cette solution (une modification élémentaire) en espérant qu'elle soit meilleure que la précédente. Une modification élémentaire dans notre cas consiste à remplacer le chemin reliant un lo_i et un lo_j de la solution courante par une seule arête reliant directement ces deux LO. Itérativement, de nouvelles solutions candidates dans le voisinage de la solution courante sont générées et testées.

Algorithm 12 : Algorithme Hill-Climbing

Input :
Une solution s_0 obtenue par un algorithme glouton
Output :
Une solution valide s amélioration de s_0

Nombre maximal d'itérations $MaxIter$
Nombre maximal total d'itérations $MaxTotalIter$
$NbIter = 0$, $NbTotalIter = 0$, $s_{actuelle} = s_0$
repeat
  $NbTotalIter = NbTotalIter + 1$
  repeat
    $NbIter = NbIter + 1$
    Générer $s_{voisine}$ par une transformation élémentaire de $s_{actuelle}$
    if $Pert(S_{voisine}) > Pert(S_{actuelle})$ **then**
      $s_{actuelle} = s_{voisine}$
      $NbIter = 0$
    end if
  until $NbIter == MaxIter$
until $NbTotalIter == MaxTotalIter$

$s = s_{actuelle}$
Return s

Ce procédé est répété aussi longtemps que la valeur de la fonction $Pert$ est améliorée. La recherche s'arrête dès qu'un optimum local est atteint ou que le nombre maximum d'itérations est atteint.
Nous avons choisi d'appliquer une variante du Hill-Climbing qui n'explore pas intégralement le voisinage mais qui choisit le premier voisin de meilleure qualité.
La méthode Hill-Climbing adaptée à notre problème est décrite par l'algorithme 12.

6.3.3/ AMÉLIORATION DE RECHERCHE LOCALE PAR UN ALGORITHME DE RECUIT SIMULÉ

Nous avons utilisé un algorithme basé sur le recuit simulé pour chercher une combinaison optimale de LO. Cet algorithme a la capacité de converger vers une bonne solution en un temps raisonnable pour une application quasi temps réel.
Par rapport à l'algorithme Hill-Climbing, l'algorithme de recuit simulé varie dans sa manière de choisir quand remplacer une solution s par une solution s' : si s' est meilleure que s alors s' remplace toujours s, mais si s' est moins bonne que s alors on ne peut remplacer s par s' qu'à une certaine probabilité $P(t, \triangle\ Pert(s, s'))$:

$$P(t, \triangle\ Pert(s, s')) = \frac{e^{\triangle Pert(s,s')}}{t}$$
$$avec\ \ \triangle\ Pert(s, s') = Pert(s') - Pert(s)\ et\ t \geq 0 \tag{6.11}$$

Si la température t est très grande, la probabilité P s'approche de 1. L'idée est d'initialiser t à une grande valeur pour bouger à chaque nouvelle solution créée sans quasiment prendre en compte leur qualité de départ, avec un parcours quasi aléatoire de l'espace des solutions. Puis t décroit lentement jusqu'à une température de gel.
Le taux de décroissance de la température t est appelé taux d'amortissement. Nous choisissons un taux d'amortissement fonctionnant par paliers comme suit :

$$t_{it} = g(t_{it-1}) = \lambda * t_{it-1} = \lambda^{it} * t_0$$
$$\text{où } it \text{ est l'ittération actuelle et } 0 < \lambda < 1 \tag{6.12}$$

L'algorithme de recuit simulé est le suivant :

Algorithm 13 : Algorithme de recuit simulé

Input :
Une solution s_0 obtenue par un algorithme glouton
Output :
Une solution valide s amélioration de s_0

Nombre maximal d'itérations $MaxIter$
Constante de refroidissement λ
$NbIter = 0$
$it = 0$
$s_{actuelle} = s_0$
Température initiale $t = t_{it}$

while $t > t_{gel}$ **do**
  repeat
    $NbIter = NbIter + 1$
    Générer $s_{voisine}$ par une transformation élémentaire de $s_{actuelle}$
    $\triangle\ Pert = Pert(s_{voisine}) - Pert(s_{actuelle})$
    if $\triangle\ Pert < 0$ **then**
      $s_{actuelle} = s_{voisine}$
    else
      $r = random(0, 1)$
      if $r < P(t, \triangle\ Pert)$ **then**
        $s_{actuelle} = s_{voisine}$
        $NbIter = 0$
      end if
    end if
  until $NbIter == MaxIter$
  $it = it + 1$
  $t_{it} = \lambda * t_0$
  $t = t_{it}$
end while
$s = s_{actuelle}$
Return s

6.3.4/ AMÉLIORATION DE RECHERCHE LOCALE PAR UN ALGORITHME TABOU

La méthode tabou consiste à explorer à partir d'une solution s donnée le voisinage que nous appelons $N(s)$, et à choisir la solution dans ce voisinage qui donne la meilleure

valeur de la fonction objectif $Pert$. Nous appelons cette meilleure solution m.
Il est essentiel de noter que cette opération peut conduire à dégrader la valeur de la fonction $Pert$ car l'algorithme accepte parfois des solutions qui n'améliorent pas toujours la solution courante : c'est le cas lorsque toutes les solutions du voisinage ont une valeur plus mauvaise de la fonction objectif. C'est à partir de ce mécanisme que l'on sort d'un optimum local.
Le risque cependant est qu'à l'étape suivante, on retombe dans l'optimum local auquel on vient d'échapper. C'est pourquoi il faut que l'algorithme ait de la mémoire : le mécanisme consiste à interdire dans une liste tabou les dernières solutions visitées ou certaines composantes des solutions. Dans notre cas nous proposons de conserver les dernières solutions déjà explorées dans une liste tabou T d'une taille donnée t, avec t un paramètre ajustable de l'algorithme. T doit conserver des solutions complètes, ce qui pour un jeu de données de grande taille, nécessite l'archivage d'une grande quantité de données. Nous contournons cet obstacle en ne gardant en mémoire que les t dernières solutions associés à la valeur de la fonction $Pert$. Quand le nombre t est atteint, chaque nouvelle solution sélectionnée remplace la plus ancienne dans la liste. La construction de la liste tabou est basée sur le principe FIFO (First In First Out), c'est-à-dire le premier entré est le premier sorti. Comme critère d'arrêt nous fixons un nombre maximum d'itérations.
L'algorithme tabou est le suivant :

Algorithm 14 : Algorithme tabou

Input :
Une solution s_0 obtenue par un algorithme glouton
Output :
Une solution valide s amélioration de s_0

Nombre maximal d'itérations $MaxIter$
Une liste tabou initialement vide $T = \emptyset$
t taille de la liste tabou
m meilleure solution
$NbIter = 0$
$s_{actuelle} = s_0$
$m = s_0$
$k = 0$ compteur sur la taille de la liste tabou
repeat
  $NbIter = NbIter + 1$
  Générer un ensemble $N(s_{actuelle})$ de solutions voisines à $s_{actuelle}$
  //Choisir $s_{voisine}$ le meilleur voisin de s tel que $s_{voisine} \notin T$ et $\forall s'_{voisine} \in N(s_{actuelle}), Pert(s_{voisine}) > Pert(s'_{voisine})$
  $s_{voisine} = best(N(s_{actuelle}))$
  if $Pert(s_{voisine}) > Pert(m)$ **then**
    if $k < t$ **then**
      $m = s_{voisine}$
      Mettre à jour T //ajouter $s_{voisine}$ à T
      $k = k + 1$
    else
      $k = 0$
    end if
  end if
until $NbIter == MaxIter$
$s = m$
Return s

6.3.5/ AMÉLIORATION DE RECHERCHE LOCALE PAR UN ALGORITHME VNS

La recherche à voisinage variable (Hansen et al., 2010) est une méthode récente proposée afin d'améliorer les performances les techniques classiques de recherche locale. Il s'agit cette fois ci de définir non pas un voisinage, mais plusieurs voisinages possibles pour une solution.
Avant tout, on a besoin d'une solution de départ (obtenue par un algorithme glouton). En-

suite, pour explorer un espace de solutions, il faut savoir comment passer d'une solution à une autre (figure 6.12).

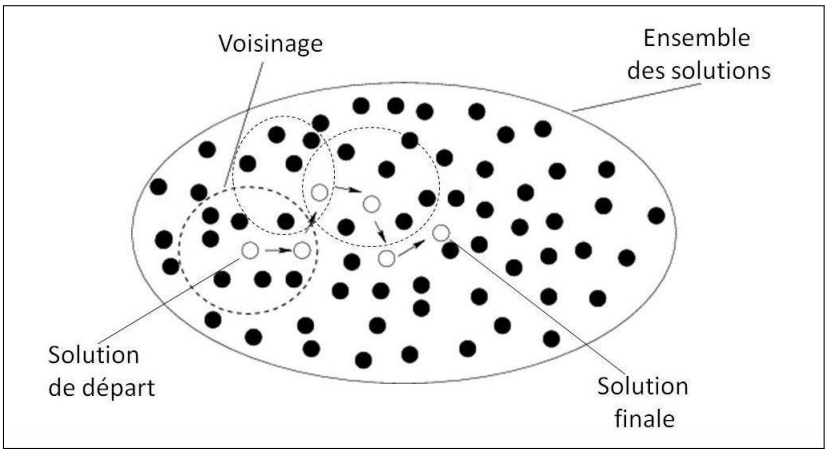

FIGURE 6.12 – Exploration de l'espace des solutions par une méthode VNS

L'algorithme procède comme suit : à partir d'une solution s' choisie aléatoirement dans le premier voisinage N_0 de la solution initiale s_0, l'algorithme effectue une recherche locale pour trouver une meilleure solution. Quand un optimum local est trouvé, l'algorithme relance le processus de recherche locale en considérant cette fois ci le second voisinage de la solution courante, et ainsi de suite jusqu'à trouver un optimum qui ne puisse pas être amélioré par aucun des voisinages définis.
La recherche locale effectuée à chaque étape de l'algorithme peut être une simple méthode de descente, ou bien une méthode plus sophistiquée comme la méthode de recuit simulé ou la méthode tabou. Plus le nombre de voisinage est élevé, plus la méthode se rapproche d'une recherche globale, et plus les temps d'exécution sont longs. Il s'agit donc de trouver un compromis en temps d'exécution et en qualité.

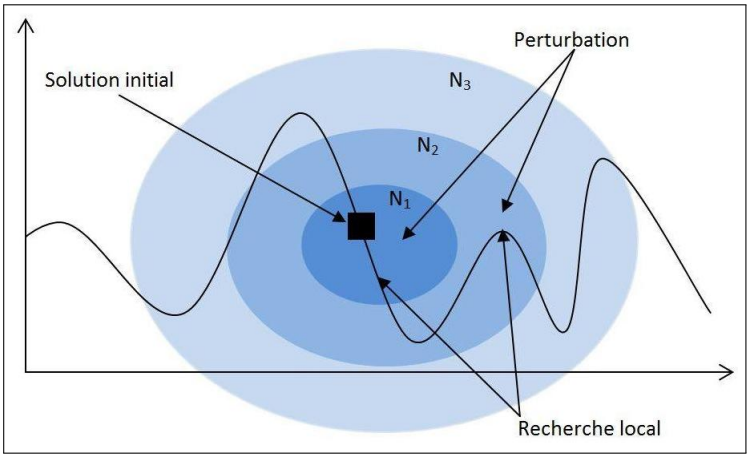

FIGURE 6.13 – Introduction de la perturbation dans la recherche à voisinage variable

Nous choisissons d'utiliser des voisinages de plus en plus larges. Ainsi le premier voisinage est obtenu de la même manière que pour les algorithmes présentés dans les sections précédentes : il s'agit de changer aléatoirement un chemin entre un lo_i et un lo_j par une seule arête reliant directement ces deux LO. L'algorithme comporte une phase de perturbation qui consiste à s'échapper d'un optimum local pour diversifier la recherche. Pour sortir des optimum locaux, nous introduisons de nouveaux voisinages N_n, avec $n \in \{1, 2, ..., n_{max}\}$. La perturbation développée correspond à modifier la solution courante en appliquant n transformations élémentaires induisant le voisinage N_n. n est systématiquement mis à jour (figure 6.13).
L'algorithme de recherche à voisinage variable procède comme suit :

Algorithm 15 : Algorithme de recherche à voisinage variable

Input :
Une solution s_0 obtenue par un algorithme glouton
Output :
Une solution valide s amélioration de s_0

Nombre maximal d'itérations $MaxIter1$
Nombre maximal d'itérations $MaxIter2$
n_{max} nombre maximum de transformations élémentaires
$NbIter = 0$, $s_{actuelle} = s_0$, $n = 1$
while $n < n_{max}$ **do**
  while $NbIter < MaxIter1$ **do**
    Générer un ensemble $N_k(s_{actuelle})$ de solutions voisines à $s_{actuelle}$
    $s_{voisine} = ChoixAleatoireDans(N_k(s_{actuelle}))$
    $s'_{voisine} = HillClimbing(s_{voisine}, MaxIter2)$
    if $Pert(s'_{voisine}) > Pert(s_{voisine})$ **then**
      $s_{actuelle} = s'_{voisine}$
      $n = 1$
    else
      $NbIter = NbIter + 1$
    end if
  end while
  $n = n + 1$
  $NbIter = 0$
end while
$s = s_{actuelle}$
Return s

6.4/ RÉSULTATS EXPÉRIMENTAUX

Après avoir sélectionné les heuristiques qui nous intéressent pour la résolution du problème de combinaison optimisée des LO, et après les avoir implémentées, nous effectuons des tests sur les différentes métaheuristiques. Quelques benchmarks ont été réalisés pour comparer l'efficacité des différents algorithmes présentés dans la section 6.3. L'objectif de ce comparatif est de déterminer quelle métaheuristique est la plus efficace pour améliorer la qualité des solutions par rapport aux résultats fournis par l'algorithme glouton. Nous présentons dans cette deuxième partie du chapitre, les jeux de données utilisés pour nos tests, les paramètres choisis pour les métaheuristiques, et les résultats expérimentaux de ces tests.

6.4.1/ JEUX DE DONNÉES

Nos benchmarks ont été réalisés à partir de jeux de données réelles et de données aléatoires :

– *Données réelles :* Dans le jeu de données réelles nous proposons un ensemble de LO accessibles sur trois types de supports de formations : ordinateur, tablette et smartphone. Cet ensemble est composé en total de 404 LO, incluant 22 objectifs de formations obligatoires à atteindre. Chaque LO possède une durée et une pondération. Les LO sont reliés entre eux par des arcs. Nous comptons dans ce jeux de données 2178 transitions possibles entre les LO. Ce jeu de données est extrait du catalogue de CrossKnowledge.
– *Données aléatoires :* Dans le jeu de données aléatoires nous proposons un ensemble de LO accessibles sur trois types de supports de formations : ordinateur, tablette et smartphone. Cet ensemble est composé en total de 3488 LO, incluant 75 objectifs de formations obligatoires à atteindre. Les durées et les poids des LO sont aussi définis aléatoirement. Les LO sont reliés entre eux par des arcs. Nous comptons dans ce jeu de données 19705 transitions possibles entre les LO.

Afin de simuler au mieux une utilisation réelle, nous disposons de plusieurs profils apprenants décrits par des préférences différentes et des contextes différents. Pour nos tests expérimentaux, nous avons généré 3 profils apprenants :

– *Appr1 :* L'apprenant 1, a une préférence pour des contenus de type lecture de densité sémantique moyenne non interactifs de difficulté de niveau 3 et de préférence en langue française. Il est dans son bureau et souhaite utiliser son ordinateur pour se former. En cours de formation cet apprenant se déplace

dans un autre bureau et souhaite maintenant utiliser sa tablette.
– *Appr2 :* L'apprenant 2, a une préférence pour des contenus de type audio de densité sémantique faible interactifs et de difficulté de niveau 1. Il est dans un bus et souhaite utiliser son Smartphone pour se former, si non une tablette. Arrivé chez lui, il souhaite reprendre sa formation mais sur son ordinateur.
– *Appr3 :* L'apprenant 3, suit une formation de niveau de difficulté 2, intégrant des quizzs et des exercices récapitulatifs. Il est dans un espace de travail ouvert et a à sa disposition un ordinateur, une tablette pour se former. Il se reconnecte à sa plateforme d'apprentissage le week-end pour reprendre sa formation et a à sa disposition une tablette et un Smartphone pour se former.

Dans le chapitre 6, nous avons défini la fonction $Pert$ comme étant l'agrégation de deux objectifs. Les valeurs choisies de α et β sont laissées à l'appréciation de l'utilisateur final. α et β peuvent varier entre [0,1] pour mettre un des deux objectifs en avant, ou leur donner la même importance. Dans notre cas nous faisons varier α et β pour chacun des profils apprenants définis précédemment comme suit : *Pert1*($\alpha = 0.1, \beta = 0.9$), *Pert2*($\alpha = 0.7, \beta = 0.3$) et *Pert3*($\alpha = 0.5, \beta = 0.5$).
Ainsi nous obtenons 9 instances à tester : *Appr1Pert1*, *Appr1Pert2*, *Appr1Pert3*, *Appr2Pert1*, *Appr2Pert2*, *Appr2Pert3*, *Appr3Pert1*, *Appr3Pert2* et *Appr3Pert3*.

6.4.2/ PARAMÉTRAGE DES ALGORITHMES ET RÉSULTATS

Dans cette section nous rapportons les résultats obtenus en utilisant un simple algorithme de type glouton, obtenus respectivement sur des données réelles et des données aléatoires pour calculer un parcours de formation global.
Nous rapportons aussi les résultats obtenus par les algorithmes glouton (G), Hill-Climbing (HC), de recuit simulé (RS), tabou (T), et de recherche à voisinage variable (VNS), obtenus respectivement sur des données réelles et des données aléatoires pour calculer un parcours de formation global. L'objectif est de justifier l'utilisation d'algorithmes complexes de type métaheuristiques par rapport un une méthode simple de type glouton.
Les résultats reportés dans ce tableau 6.1 présentent pour chaque algorithme les valeurs de la fonction objectif $Pert$ pour les 9 instances obtenus en combinant les apprenants *Appr1*, *Appr2* et *Appr3* avec les variations *Pert1*, *Pert2* et *Pert3*. Pour chaque algorithme nous rapportons les valeurs moyennes de la fonction objectif $Pert$ pour 1000 exécutions de chaque algorithme, ainsi que le temps moyen d'exécution en secondes.

Nous paramétrons les différents algorithmes comme suit :
– Pour le Hill-Climbing, nous fixons $MaxTotalIter = 1000$ et $MaxIter = 5000$.

– Pour le recuit simulé, nous fixons $MaxIter = 4000$, $\lambda = 0.6$, $t_0 = 10000$ et $t_{gel} = 100$.
– Pour le tabou, nous fixons $MaxIter = 5000$ et $t = 500$.
– Pour la recherche à voisinage variable, nous fixons $MaxIter1 = 500$, $MaxIter2 = 150$ et $n_{max} = 3$.

Ces tests ont été effectués sur une machine virtuelle de type Intel(R) Xeon(R) CPU E5-2430l 0 @ 2.20 Ghz.

Instance	**Jeu de données réelles**				
	G	**HC**	**RS**	**T**	**VNS**
Appr1Pert1	32.36	36.86	40.21	57.55	**67.46**
Appr2Pert1	35.96	46.78	53.06	58.46	**79.20**
Appr3Pert1	60.26	61.16	68.36	71.16	**81.92**
Appr1Pert2	147.33	149.79	157.19	158.42	**164.36**
Appr2Pert2	152.71	158.30	159.32	161.15	**166.78**
Appr3Pert2	150.05	153.37	156.11	159.77	**165.38**
Appr1Pert3	113.82	115.32	129.77	131.79	**135.97**
Appr2Pert3	122.79	123.80	125.83	129.83	**136.97**
Appr3Pert3	111.37	118.46	125.33	126.28	**136.00**
Instance	**Jeu de données aléatoires**				
	G	**HC**	**RS**	**T**	**VNS**
Appr1Pert1	35.00	53.90	69.55	81.73	**161.1**
Appr2Pert1	155.75	191.73	196.24	197.14	**204.40**
Appr3Pert1	154.88	158.48	187.26	194.47	**205.98**
Appr1Pert2	257.60	259.32	265.32	267.11	**272.75**
Appr2Pert2	201.63	217.28	221.20	228.27	**240.57**
Appr3Pert2	261.67	263.79	264.13	268.65	**272.79**
Appr1Pert3	105.59	122.75	130.37	149.87	**168.97**
Appr2Pert3	165.42	175.48	182.48	186.00	**218.48**
Appr3Pert3	191.11	195.40	198.13	200.21	**206.14**

TABLE 6.1 – Valeurs moyennes de $Pert$ pour les algorithmes G, HC, RS, T et VNS

Le tableau comparatif 6.1 montre que les quatre algorithmes HC, RS, T et VNS améliorent considérablement la valeur moyenne de la fonction $Pert$ par rapport à un simple algorithme de type glouton aussi bien pour le jeu de données réelles que pour le jeu de données aléatoires.
Le tableau montre aussi que la recherche à voisinage variable randomisé est la méthode qui fournit les résultats les plus proches de l'optimum (les plus grandes valeurs moyennes de la fonction $Pert$, aussi bien pour le jeu de données réelles

que pour le jeu de données aléatoires). En deuxième position l'algorithme tabou, fournit des résultats moins bons que la recherche à voisinage variable randomisé, mais meilleurs que l'algorithme de recuit simulé et l'algorithme Hill-Climbing. Finalement, les valeurs moyennes de la fonction $Pert$ des combinaisons trouvées par l'algorithme Hill-Climbing sont les moins bonnes (les plus petites valeurs moyennes de la fonction $Pert$, aussi bien pour le jeu de données réelles que pour le jeu de données aléatoires).

Algorithme	**Temps (s) : Jeu de données réelles**
G	5.01
HC	5.23
RS	5.36
T	5.56
VNS	5.58
Algorithme	**Temps (s) : Jeu de données aléatoires**
G	6.12
HC	6.21
RS	6.37
T	6.52
VNS	6.56

TABLE 6.2 – Temps moyen d'exécution pour les algorithmes G, HC, RS, T et VNS

En ce qui concerne le temps d'exécution, et pour répondre aux contraintes industrielles pour le développement de l'application, nous limitons le temps d'exécution des différents algorithmes. Le temps d'exécution de chaque algorithme doit rester borné entre 0 et 10.
Dans le tableau 6.2 et pour chaque algorithme nous rapportons le temps moyen d'exécution en secondes. L'algorithme glouton reste le plus rapide. L'algorithme de recherche à voisinage variable met un peu plus de temps d'exécution. Mais aucun algorithme ne dépasse les 10s (inférieur à 6s pour un jeu de données réelles et inférieur à 7s pour un jeu de données aléatoires).

C'est finalement l'algorithme de recherche à voisinage variable qui nous fournit le meilleur compromis entre qualité des résultats et temps d'exécution.

6.5/ CONCLUSION

Ce chapitre est constitué de quatre grandes parties :
Dans la première partie nous avons défini le problème d'optimisation combina-

toire en m-learning. L'objectif est de trouver un ensemble d'objets pédagogiques corrélés, formant un parcours de formation valide optimisant l'expérience de formation de l'apprenant. Cette optimisation doit prendre en considération deux objectifs : maximiser le poids total de la formation globale pour maximiser le gain de connaissance, et minimiser l'écart entre la durée totale prévue et la durée autorisée de formation. La solution établie doit prendre en considération plusieurs contraintes dont la disponibilité des supports de formations et le contexte d'apprentissage du moment.

Dans la deuxième partie nous avons dressé un état de l'art des métaheuristiques. Nous avons présenté les principales métaheuristiques utilisées pour résoudre des problèmes d'optimisation. Elles ont été regroupées en deux grandes familles, les métaheuristiques de recherche locale et les métaheuristiques évolutives. La première famille recherche la meilleure solution en parcourant l'espace des solutions en passant d'une solution à une autre de manière itérative. Les différences principales entre ces méthodes se situent au niveau du choix de la solution voisine et au niveau du critère d'arrêt. La deuxième famille recherche la meilleure solution en se basant sur un échantillon de solutions candidates au lieu d'une seule solution candidate. Les différences entre ces méthodes résident principalement dans les techniques de sélection des individus de la population, de croisement entre ces individus, de mutations et de l'intégration des nouveaux individus dans l'échantillon de population suivant.

Les métaheuristiques basées sur une population sont généralement plus coûteuses, en temps de calcul, que les métaheuristiques de recherche locale pour trouver la meilleure ou une bonne solution. Cependant, un de leurs intérêts est qu'elles engendrent une population de solutions tendant vers l'optimum alors que celles à état simple ne proposent qu'une seule solution.

Pour le problème qui nous est posé, et surtout pour l'application à fournir à l'utilisateur final, il n'est pas envisagé, dans un premier temps, de lui fournir plusieurs combinaisons possibles. De plus dans un contexte mobile la rapidité de réponse est exigée : le temps de calcul des métaheuristiques basées sur une population étant plus long, nous préférons utiliser des métaheuristiques de recherche locale, surtout que l'optimisation de plusieurs objectifs et la multimodalité du problème rajoutent de la complexité.

Dans les deux dernières parties du chapitre, nous avons tout d'abord présenté l'implémentation de différentes métaheuristiques pour la résolution du problème d'optimisation combinatoire, appliqué au domaine du m-learning. Ces métaheuristiques ont été testées sur un jeu de données réelles et un jeu de données aléatoires, et des benchmarks de comparaison de performances suivant la fonction de pertinence et le temps de réponse ont été réalisés.

Un algorithme simple de type glouton et quatre algorithmes plus complexes ont

été testés pour la génération de combinaisons de LO. Afin d'améliorer la solution obtenu par un algorithme glouton nous avons utilisé un algorithme basé sur le recuit simulé, l'algorithme Hill-Climbing, l'algorithme tabou et l'algorithme de recherche à voisinage variable : ces différents algorithmes ont été comparés et il en ressort que la recherche à voisinage variable est supérieur en terme de qualité de la solution finale.

IV

CONCLUSION ET TRAVAUX FUTURS

CONCLUSION ET TRAVAUX FUTURS

CONCLUSION

Le travail de recherche présenté dans cette thèse définit une nouvelle approche pour un système de recommandation sensible au contexte appliqué au domaine du m-learning combinant les technologies du Web sémantique et des algorithmes d'optimisation combinatoire.
Ce système est composé d'une partie statique représentant à la fois les contenus d'apprentissage et le contexte de l'apprenant et une partie adaptative et dynamique contenant des règles de comportement dans un contexte de mobilité et des métaheuristiques d'optimisation combinatoire.
Notre objectif est de proposer à l'apprenant une combinaison constituée de plusieurs items suivant un contexte particulier. Les items correspondent ici à des briques de formation appelés objets pédagogiques, ou LO pour Learning Object. Notre approche permet aux formateurs de représenter leur savoir-faire en utilisant des règles métier et une ontologie de domaine du m-learning pour assurer une hétérogénéité des connaissances. Ensuite, dans un environnement de mobilité, elle permet de prendre en compte les contraintes de l'environnement et les contraintes utilisateur. Enfin, la partie métaheuristique de notre proposition permet une combinaison dynamique d'objets pédagogiques pour former un parcours de formation en fonction de ces contraintes.

En étudiant l'architecture des systèmes hypermédia adaptatifs nous avons pu identifier un ensemble minimum de trois couches pour modéliser notre système. Ses couches adaptées à notre architecture sont le modèle des LO, le modèle de contexte et le modèle d'adaptation :

– Le premier modèle, le modèle des LO, permet de présenter les contenus sous formes d'instanciation d'objets pédagogiques. L'idée fondamentale derrière la création des objets pédagogiques est la possibilité de construire un parcours de formation autour de composants de petite taille sémantiquement hétérogènes qui peuvent être réutilisés plusieurs fois dans différents contextes d'apprentissage.

– Le deuxième modèle, le modèle de contexte, permet de présenter le contexte d'apprentissage.
– Finalement dans le modèle d'adaptation sont implémentés les différents algorithmes utilisés pour proposer des recommandations aux utilisateurs.

Nous avons bâti notre proposition sur cet ensemble de couches en regroupant dans une partie nommée serveur sémantique de connaissances le modèle des LO et le modèle de contexte. Nous avons ajouté à ce serveur sémantique de connaissances un modèle de contraintes permettant de prendre en compte le savoir-faire du fournisseur des contenus dans les propositions du système de recommandation sous forme de règles métier.

L'instanciation de chaque couche de l'architecture pour le système m-learning a été décrite en détail. Nous avons mis au point des techniques permettant d'apporter des solutions à des verrous technologiques comme l'hétérogénéité et la portabilité des objets pédagogiques et la prise en compte d'un contexte variable et dynamique. Le processus de création d'une ontologie de domaine du m-learning a été détaillé. Cette ontologie permet de décrire en partie le contexte de l'apprenant et les contenus pédagogiques en se basant sur la base de données de CrossKnowledge, des normes de standardisation en apprentissage électronique et les connaissances d'experts. L'ontologie est enrichie par un ensemble de règles dans le but d'effectuer de l'assistance aux décisions de l'utilisateur et de décrire le comportement du système dans un contexte mobile. Ces règles sont aussi utilisées pour apporter une pondération aux objets pédagogiques, donnant les intérêts des apprenants pour les différents objets pédagogiques dans différents contextes. Le processus de recherche combinatoire du modèle adaptatif va rechercher parmi la multitude de combinaisons possibles d'objets pédagogiques, une solution convenable optimale si possible, à proposer à l'apprenant.

L'architecture du système de recommandation combinatoire a été appliquée au domaine du m-learning pour l'entreprise CrossKnowledge. Le but de cette application est de proposer à l'apprenant un ensemble de parcours de formation. Chaque parcours de formation contient ainsi un ensemble d'objets pédagogiques destinés à être délivrés sans interruption, et respectant les règles de précédence de manière à former un ensemble cohérent qui va permette de remplir l'objectif de formation, et qui correspond à chaque étape au contexte de l'apprenant, notamment aux supports de délivrance qui sont à sa disposition. Ce problème est défini formellement comme un problème d'optimisation combinatoire. Afin de tendre vers une bonne solution en un temps raisonnable, nous avons proposé d'utiliser des algorithmes de type métaheuristiques. Nous avons présenté l'implémentation de différentes métaheuristiques pour la résolution du problème d'optimisation combinatoire, appliqué au domaine du m-learning. Ces métaheuristiques ont été

testées sur un jeu de données réelles et un jeu de données aléatoires, et des benchmarks de comparaison ont été réalisés, il en ressort que la recherche à voisinage variable est supérieur en terme de qualité de la solution finale.

TRAVAUX FUTURS

Nous avons identifié plusieurs pistes pour poursuivre nos travaux de recherche :

– Intégrer les travaux de l'équipe de recherche CIMEOS [1] (Laboratoire en Sciences de l'Information et de la Communication de l'Université de Bourgogne) :
  L'objectif est de faire évoluer notre proposition de système m-learning vers une solution socio-technique en intégrant les résultats des travaux de recherche de l'équipe CIMEOS. L'équipe de recherche CIMEOS travaille sur l'analyse et la compréhension du rôle de la socialisation électronique d'une part, et de l'ergonomie des interfaces d'autre part.
  Des travaux sont en cours sur l'organisation d'une expérimentation psychosociologique et ergonomique permettant de tester in situ les usages et les pratiques se développant autour de la manipulation du prototype de système m-learning que nous proposons dans notre architecture. Dans ce projet, il convient d'observer l'appropriation du prototype via des méthodologies éprouvées des sciences sociales (e.g. observation in situ, entretiens semi-directifs) et d'en intégrer les conclusions dans le prototype final (conception assistée par l'usage).
– Intégration de la dimension sociale dans l'apprentissage et évolution vers un système de recommandation hybride :
  L'apprentissage social (social learning ou s-learning) [2] est un mode d'apprentissage qui permet de partager, de construire, de collaborer avec d'autres à distance via des outils collaboratifs (wiki, chat, forum, blog, etc.) ou des réseaux sociaux d'entreprises, d'universités, etc. L'apprenant n'est plus uniquement destinataire du savoir dispensé par le formateur mais il devient acteur de la plateforme d'apprentissage. Il participe à son propre apprentissage et collabore avec le formateur et les autres apprenants. Enfin, il partage son savoir et ses connaissances. On parle de communauté d'apprentissage.
  Notre proposition de systèmes de recommandation se classe parmi les systèmes de recommandation basés sur le contenu. Nous pouvons ainsi enrichir notre proposition actuelle de système de recommandation basé sur le contenu en lui ajoutant des méthodes collaboratives. Par exemple, il est pos-

1. http ://cimeos.u-bourgogne.fr/
2. http ://www.elearning-cegos.fr/notre-approche-e-learning/social-learning/

sible d'identifier des clusters d'apprenants, de sorte à relier les apprenants entre eux selon leur profil et les notes données aux objets pédagogiques ou combinaisons d'objets pédagogiques. Il faudrait alors ajouter de nouvelles techniques pour prendre en compte ces groupements d'utilisateurs. Elles engendreraient une modification de la pondération des objets pédagogiques selon ces groupements. Le système hybride résultant donnerait des résultats plus pertinents pour les apprenants.

– Automatisation du processus de génération des règles métier :
  Le modèle de contraintes de notre architecture de système de recommandation a pour objectif de décrire les stratégies d'apprentissage et les connaissances des experts en utilisant des règles métier. Pour le moment, dans notre modèle ces règles sont créées et générées manuellement. Pour une application industrielle ce processus de génération des règles doit se faire automatiquement, sans besoin de faire intervenir un expert pour chaque changement de contexte de l'apprenant. Une idée est de créer un ensemble de règles génériques qui puissent s'adapter dynamiquement en fonction des chargements de contexte.

V

ANNEXES

A

LES HYPERMÉDIA ADAPTATIFS

L'hypermédia, obtenu par la fusion des techniques de l'hypertexte et du multimédia, présente plusieurs avantages dans le cadre de l'apprentissage électronique. En effet, d'une part, la composante multimédia améliore l'aspect visuel et ludique de l'apprentissage, et ainsi renforce l'intérêt de l'apprenant par rapport au système d'apprentissage (Hoogeveen, 1995). D'autre part, la composante hypertexte améliore la qualité de l'apprentissage grâce à sa structure non linéaire, en aidant l'apprenant à construire sa connaissance (Delestre et al., 1998). Cependant, un hypermédia ne peut pas offrir des contenus personnalisés. Les apprenants ont alors accès au même ensemble de contenus sans prendre en compte leurs différences : niveau de connaissances, intérêts, motivations, objectifs, etc. Différentes recherches dans le domaine des hypermédia adaptatifs essayent de comprendre les relations entre le profil de l'apprenant, le contexte dans lequel se déroule l'apprentissage, et les contenus, à des fin d'adaptation selon les besoins de l'apprenant.

La première partie de ce chapitre est divisée en plusieurs sections. La première section présente les avantages et les limites des systèmes hypermédia classiques. La deuxième section décrit les nombreuses architectures attachées au domaine des systèmes hypermédia adaptatifs. Cette étude nous permet d'identifier une architecture pivot qui nous servira pour construire notre proposition. La troisième section présente les spécificités des systèmes hypermédia adaptatifs dans le domaine éducatif.

A.1/ LES HYPERMÉDIA CLASSIQUES

La notion d'hypertexte fait son apparition en 1945 avec le projet MEMEX, lorsque (Bush, 1945) a constaté que le nombre de rapports de recherche augmente rapidement et qu'il devenait de plus en plus difficile de consulter des rapports. Il décrit alors un projet MEMEX (MEMoryindEX) dans lequel il propose une machine mul-

timédia qui permet de sauvegarder des milliers d'informations sous forme de microfilms pour une meilleure collecte et consultation des documents : «*a MEMEX is a device in which individual stores all his books, records and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility*». Il remarque toutefois, que pour avoir un meilleur usage de cette machine, il faut savoir chercher et retrouver l'information désirée parmi les milliers d'informations répertoriées. Il définit alors un mécanisme d'indexation des documents par mots-clefs et propose un mécanisme qui permet d'associer les documents et ainsi de naviguer d'un document à un autre. Malheureusement ce projet ne verra pas le jour, car sa vision était trop en avance par rapport aux technologies de l'époque.
Le terme hypertexte est introduit pour la première fois en 1965 par Nelson. Il imagine un projet, nommé XANADU, dont l'objectif est de créer une structure permettant de relier toute la littérature du monde dans un réseau de publications hypertextué, universel et accessible à tous (Nelson, 1965). Au début de ses travaux, Nelson a tenté d'appliquer sa vision à l'apprentissage assisté par ordinateur et a proposé l'hypertexte comme nouvelle approche. Il dit qu'il faut laisser à l'apprenant le soin de choisir ce qu'il désire apprendre. Il faut donc lui donner une variété de ressources intéressantes pour ce faire. Dans ces circonstances, les apprenants sont motivés pour atteindre un niveau d'accomplissement personnel supérieur à ce qu'ils peuvent accomplir dans le cadre d'un apprentissage traditionnel. La lente mise au point de XANADU s'est heurtée à un concept moins puissant, mais simple, ouvert, libre et utilisable immédiatement, le World Wide Web. Cependant, si Xanadu a failli dans sa réalisation, il a joué un rôle important dans l'évolution des systèmes hypertextes [1].
En 1968, Engelbart présente le premier système informatique fonctionnant sous mode d'hypertexte, le NLS (oN Line System), une base de données textuelle qui facilite le travail en collaboration et où tous les intervenants sont reliés en réseau. Ce système fut par la suite intégré dans le projet AUGMENT. Ce dernier propose un environnement en réseau pour la collecte de documents, des notes et des rapports de recherche tout en fournissant des moyens de planification, d'analyse, de résolution de problèmes entre collaborateurs (Engelbart, 1984).
Dès 1990, plusieurs systèmes utilisant les techniques hypertextes apparaissent sur le marché et le domaine qui tire le plus grand bénéfice de cette technique est le Web.

Mais qu'est-ce qu'un hypertexte ? Un hypertexte est un système qui permet de présenter différentes informations de façon non linéaire. Les hypertextes sont constitués de nœuds et de liens : Les nœuds, ou pages de l'hypertexte, sont

1. http ://fr.wikipedia.org/wiki/Projet_Xanadu

constitués d'informations textuelles ; Les liens forment des ponts entre les pages (Balasubramanian, 1994). Les nœuds et les liens forment ainsi un réseau hypertexte. Les hypermédias se différencient des hypertextes par le contenu des nœuds. Les nœuds ne contiennent plus seulement des données textuelles, mais peuvent aussi être composés de média divers, tels que des images, des vidéos, etc.

Les systèmes hypertextes/hypermédias proposent deux types d'interactions permettant d'aborder l'information de façons différentes : l'interrogation et la navigation. La méthode d'accès par interrogation existait déjà dans les systèmes de bases de données et dans les systèmes de recherche d'information, par contre la méthode d'accès pas navigation est lié à la naissance de la technologie hypertexte. (1) La méthode d'accès par interrogation suppose que l'utilisateur sait ce qu'il cherche (un nœud particulier ou un ensemble de nœuds du réseau). Pour y accéder, il aura à formuler une requête pour récupérer l'information. (2) La méthode d'accès par navigation consiste à explorer librement les nœuds en utilisant les liens pour atteindre d'autres nœuds.

La méthode d'accès par interrogation et la méthode d'accès par navigation introduisent ainsi de la flexibilité et de la simplicité pour l'accès à l'information.

Mais au-delà de ces avantages, les hypermédias véhiculent principalement deux problèmes, à savoir la *désorientation* et la *surcharge cognitive* :

– La désorientation est définie comme « *un effet cognitif produit chez l'usager qui perd la liaison entre son projet de navigation, et les informations qu'il est en train de lire* » (Rheaume, 1993). En d'autres termes, ce phénomène provient de la perte de repères au sein de la structure du réseau hypertexte. L'utilisateur risque alors de se poser des questions du type : où suis-je ?, pourquoi suis-je là ? et que dois-je faire ?. C'est cet effort de concentration nécessaire pour maintenir différentes navigations en même temps qui entraîne une surcharge cognitive de l'utilisateur du système.
– Un utilisateur est en surcharge cognitive quand il se trouve submergé par l'abondance d'informations et par l'excès de traitement à réaliser (Platteaux, 2002). Cette surcharge correspond à la diminution d'attention et de concentration sur l'objectif principal due à notre mémoire humaine, qui n'est pas capable de tout retenir (Rheaume, 1993).

En effet, les systèmes hypermédia classiques sont conçus de façon à ce que les mêmes nœuds et les mêmes liens soient présentés à tous les utilisateurs sans prendre en considération leurs besoins, leurs expériences, leurs connaissances, etc. En conséquence, en empruntant des chemins inintéressants pour lui, l'utilisateur peut facilement se perdre dans l'hyperespace et être surchargé cognitivement.

Afin de minimiser les aspects négatifs des hypermédia classiques, les travaux de recherche se sont focalisés sur une nouvelle approche des hypermédias : les hypermédia adaptatifs. L'objectif de ces systèmes est d'adapter la présentation de l'information et d'aider l'utilisateur à se diriger dans l'hyperespace.

A.2/ LES HYPERMÉDIA ADAPTATIFS

Les systèmes hypermédia adaptatifs sont devenus très populaires au début des années 1990, afin de permettre aux utilisateurs d'accéder à de l'information personnalisée. Selon (Brusilovsky, 1996) « *un système hypermédia adaptatif est un hypertexte ou un hypermédia qui reflète certaines caractéristiques particulières d'un individu dans un modèle d'utilisateur, et applique ce modèle pour adapter des aspects visibles de ce système à cet utilisateur* ». De cette définition, on peut déduire qu'un système hypermédia adaptatif doit remplir les critères suivants :

– Être un système d'information dont l'interface est un hypermédia,
– être capable de collecter des informations sur un utilisateur particulier et de construire un modèle utilisateur (profil utilisateur),
– être capable d'utiliser ce modèle pour adapter l'hypermédia à cet l'utilisateur.

Les systèmes hypermédia adaptatifs sont utiles dans tout domaine où les applications sont utilisées par des utilisateurs aux objectifs et aux connaissances différents. Afin de guider la création de tels systèmes, une méthodologie proposée par (Brusilovsky, 1996) consiste à répondre à un ensemble de questions. Nous répondons à ces questions dans la section suivante.

A.2.1/ MÉTHODE ET TECHNIQUES D'ADAPTATION

En empruntant la terminologie proposée par (Brusilovsky, 1996), une méthode d'adaptation désigne une idée au niveau conceptuel, une manière d'envisager l'adaptation, alors qu'une technique d'adaptation correspond aux moyens mis en œuvre pour implémenter une méthode (représentations des connaissances, algorithmes d'adaptation, etc.). Pour concevoir un système hypermédia adaptatif, il est nécessaire de répondre à un certain nombre de questions, afin de déterminer les méthodes et techniques nécessaires à cette conception (Picot-Clémente, 2011) :

Que peut-on adapter ?
La représentation d'un hypermédia sous la forme d'un réseau, permet d'identifier deux méthodes d'adaptations : soit adapter les nœuds du réseau, soit adapter les liens du réseau. De ce fait, les méthodes d'adaptation correspondent respec-

tivement, soit à l'adaptation du contenu, soit à l'adaptation de la navigation entre les contenus.
L'adaptation du contenu se décline en deux sous méthodes : l'adaptation de texte, ou l'adaptation de média. Pour le texte, l'adaptation consiste à utiliser des variantes des textes, à rajouter des explications, etc. L'adaptation de média est beaucoup moins développée. Certains systèmes permettent la substitution de média, mais les fichiers audio ou vidéo ne sont pas modifiés en fonction de l'utilisateur.
L'adaptation de navigation consiste à aider l'utilisateur à se repérer dans l'hyperespace en modifiant les liens qui lui sont proposés et en l'obligeant à utiliser certains liens plutôt que d'autres.

À quels éléments le système peut-il s'adapter ?
Il existe quatre catégories de données auxquelles un système hypermédia peut s'adapter : Premièrement, le système peut s'adapter aux connaissances de l'utilisateur concernant un domaine. Deuxièmement, il peut s'adapter aux buts de l'utilisateur (que doit-il apprendre ?, quelle tâche souhaite-t-il réaliser ?). Troisièmement, il peut s'adapter aux expériences et aux compétences de l'utilisateur. Finalement, le système peut s'adapter aux préférences de l'utilisateur concernant la présentation d'un contenu (exemple, taille des caractères, couleurs, etc.).

Quelles méthodes et techniques d'adaptation peut-on employer ?
L'adaptation peut s'appliquer soit aux contenus soit à la navigation suivant plusieurs méthodes et techniques que nous illustrons dans la figure A.1.
Nous commençons par décrire les différentes méthodes d'adaptation des contenus :

- *L'ajout d'explications :* consiste à fournir des explications supplémentaires à certaines catégories d'utilisateurs.
- *Les explications pré-requises* consistent à ajouter des explications introductives au début de chaque contenu présentant un sujet donné. Ces explications sont relatives aux pré-requis nécessaires pour aborder le sujet en question.
- *Les variantes d'explications :* consistent à proposer plusieurs variantes des explications sur un même sujet.
- *Les explications comparatives :* consistent à mettre en place des liens entre des concepts similaires. Si un concept similaire à celui déjà connu par l'utilisateur est présenté, ce dernier aura alors une explication comparative.
- *Le tri de fragments :* consiste à trier les fragments composant un contenu selon leur pertinence par rapport au profil de l'utilisateur.

Pour chacune de ces méthodes d'adaptation du contenu, il existe des techniques pour les implémenter :

– *Le texte conditionnel :* est une technique qui permet de proposer des informations supplémentaires. Ceci peut être réalisé par l'association de conditions aux informations supplémentaires. Ces conditions expriment généralement des critères requis pour y accéder. Par comparaison avec les valeurs affectées à l'utilisateur pour ces critères, le système décide de montrer ou non l'information supplémentaire.
– *Le texte allongé :* est une technique basée sur un principe d'expansion ou de réduction d'un texte dans un document hypermédia. Une partie du texte est associée à une information additionnelle, qu'il est possible de faire apparaître. Le système choisit ensuite de dévoiler ou non l'information additionnelle en fonction des spécifications données par l'utilisateur.
– *La technique de fragments :* est basée sur le choix de contenus alternatifs. Ceci est réalisable de deux façons différentes : La première consiste à créer plusieurs versions d'un contenu. Au moment de l'affichage d'un contenu, le système sélectionne la version qui correspond le mieux à l'utilisateur ; La seconde technique adopte un principe similaire mais à un niveau de granularité plus fin, en créant différentes versions de fragments du contenu. Une sélection de la version adéquate est opérée pour construire une page à présenter à l'utilisateur.

De même, la navigation peut être adaptée suivant plusieurs techniques :

– *Le guidage direct :* est la technique la plus utilisée car simple à mettre en œuvre. Elle est basée sur l'ajout d'un lien hypertexte, nommé suivant, qui permet d'accéder à la page en adéquation avec les objectifs de l'utilisateur. Pour être réellement efficace, cette technique est utilisée avec au moins une des techniques décrites dans ce qui suit.
– *L'ordonnancement des liens :* consiste à afficher les liens hypertextes suivant un ordre, définissant l'intérêt ou l'importance des contenus cibles. Les liens les plus adéquats sont disposés de manière à réduire le nombre d'actions à effectuer pour atteindre l'information souhaitée.
– *Le masquage des liens :* consiste à limiter les possibilités de navigation en supprimant des liens hypertextes qui sont en inadéquation avec les objectifs de l'utilisateur.
– *L'annotation des liens :* part du principe que l'utilisateur doit savoir où il va avant d'activer un lien. Il faut donc joindre à chaque lien des explications, textuelles ou graphiques, en fonction du profil de l'utilisateur.
– *Les cartes adaptatives :* permettent de présenter à l'utilisateur, l'organisation de l'hyperespace, à l'aide de liens, soit sous forme textuelle (une représentation

hiérarchique de l'hyperespace), soit sous forme graphique. Dès lors, il est possible de présenter à l'utilisateur une organisation plus ou moins simplifiée, en fonction de son profil.

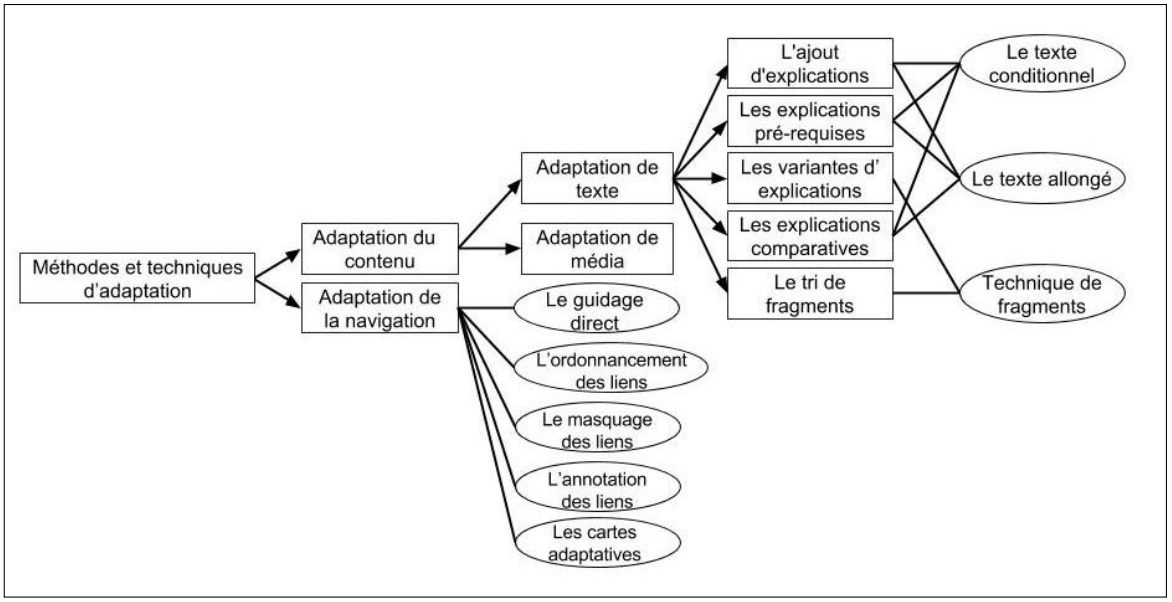

FIGURE A.1 – Méthodes et technologies des hypermédia adaptatifs

A.2.2/ ARCHITECTURE DES HYPERMÉDIA ADAPTATIFS

Plusieurs architectures ont été proposées depuis l'avènement du domaine des systèmes hypermédia adaptatifs. Dans cette section nous présentons d'abord, les architectures et les modèles les plus connus et influents du domaine, puis, nous comparons et déduisons les points communs de ces architectures. Les architectures que nous présentons, ont été essentiellement appliquées au domaine du e-learning, d'où l'appellation d'hypermédia adaptatifs éducatifs.

Modèle Dexter :
Le modèle Dexter (Halasz and Schwartz, 1994) est considéré comme un modèle de référence, et le premier essai pour unifier et clarifier les technologies de l'hypermédia.
La principale caractéristique de ce modèle, est la division de l'application hypermédia en trois couches, qui sont : la couche contenu de composants (within component layer), la couche stockage (storage layer) et la couche exécution (runtime layer). Ces couches sont interconnectées entre elles à l'aide de deux interfaces : spécification de présentation (presentation specifications) et ancrage (anchoring). L'architecture du modèle Dexter est présentée dans la figure A.2.

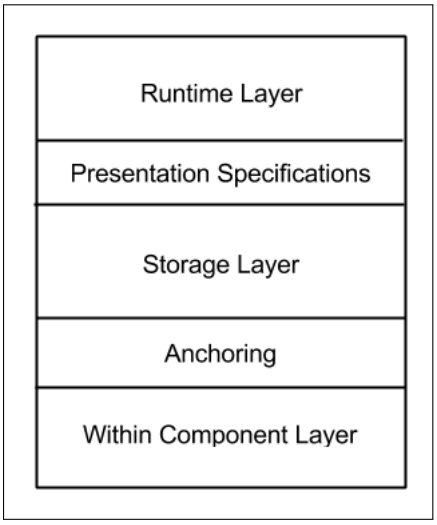

FIGURE A.2 – Modèle Dexter (Halasz and Schwartz, 1994)

La couche contenu de composants représente le contenu comme un ensemble d'éléments indépendants. La couche de stockage représente les liaisons hypertextes entre les éléments de l'application. La couche exécution représente le fonctionnement de l'application.
Cette proposition ne comporte pas de modèle utilisateur et elle ne permet pas d'adaptation, par conséquent elle ne répond donc pas aux contraintes des systèmes hypermédia adaptatifs. Les modèles apparus quelques années plus tard répondent à cette limitation et sont une amélioration du modèle Dexter.

Modèle AHAM :
AHAM (Adaptive Hypermedia Application Model) (Bra et al., 1999) est un modèle de référence pour les hypermédia adaptatifs, basé sur le modèle Dexter, et appliqué au domaine du e-learning. Ce modèle diffère du modèle Dexter par la couche stockage. Cette couche est organisée en trois composants qui sont le modèle de domaine (domain model), le modèle de utilisateur (user model) et le modèle d'adaptation (teaching model). L'architecture du modèle AHAM est présentée dans la figure A.3.
Le modèle de domaine représente le domaine d'application de l'hypermédia. Le modèle utilisateur contient des informations pertinentes sur l'utilisateur et utiles pour l'adaptation. Le modèle d'adaptation contient les règles utilisées pour l'adaptation de l'hypermédia. L'adaptation se fait en combinant les informations présentes dans le modèle de domaine et le modèle utilisateur.
Le modèle AHAM permet l'adaptation de contenus (adaptation de présentation) et de liens (adaptation de navigation) et de mettre à jour le modèle de l'utilisateur en fonction du comportement de ce dernier dans le système. L'adaptation de l'hypermédia est réalisée par un moteur d'adaptation formé d'un sélecteur

de pages (choisit les fragments à présenter à l'utilisateur) et un constructeur de pages (détermine la présentation à appliquer à ces fragments).

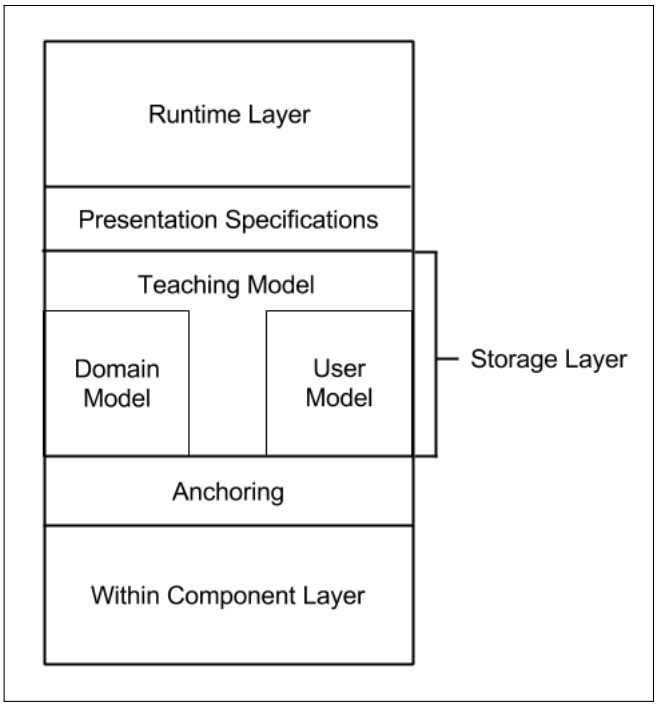

FIGURE A.3 – Modèle AHAM (Bra et al., 1999)

Le modèle AHAM est aujourd'hui une architecture de référence pour la création d'hypermédia adaptatifs.

Modèle Munich :
Les deux premiers modèles Dexter AHAM ont permis de fixer l'architecture générale d'un système hypermédia adaptatif. Le modèle Munich (Koch and Wirsing, 2002) est basé sur le modèle Dexter et partiellement sur le modèle AHAM. Il a aussi été appliqué au domaine du e-learning. Le principal apport de ce modèle est qu'il utilise un langage graphique pour décrire les différents composants d'un système hypermédia adaptatif. L'architecture en couches est remplacée par un diagramme de package UML. De même, la description des modèles de domaine et d'utilisateur est réalisée par des diagrammes de classes UML. L'architecture du modèle Munich est présentée par la figure A.4.
Le modèle utilise aussi le formalisme OCL (Object Constraint Language) pour définir formellement les contraintes posées sur les différents éléments du modèle. L'apport du modèle Munich par rapport au modèle AHAM consiste à pouvoir relier les composants du modèle de domaine par des liens tel que, "part de", "pré-requis de", etc. et non plus seulement par des liens de navigation. Le modèle utilisateur (*user meta-model*) inclut un gestionnaire d'utilisateur, et un modèle pour chaque

utilisateur du système, composé d'attributs (exemple, connaissances de l'utilisateur, ses préférences, son parcours professionnel, etc.) et de valeurs. Le modèle d'adaptation (*adaptation meta-model*) est basé sur un ensemble de règles : des règles de construction d'un contenu, des règles d'adaptation à l'utilisateur et des règles de mise à jour du profil utilisateur.

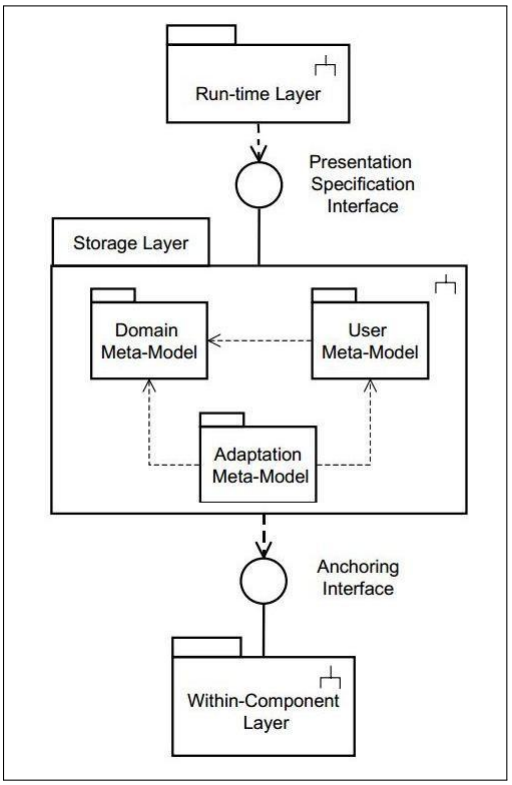

FIGURE A.4 – Modèle Munich (Koch and Wirsing, 2002)

Le modèle Munich présente l'avantage d'être formalisé de manière plus standard et plus précise. Le modèle Munich n'est pas utilisé seulement pour les hypermédia éducatifs, mais aussi dans d'autre domaines.

Modèle LAOS :
Le modèle LAOS (Cristea and Andmethods, 2003), est un modèle pour la création de systèmes hypermédia adaptatifs dans le domaine du e-learning. Il est basé sur le modèle AHAM. Il décompose les systèmes hypermédia adaptatifs en un plus grand nombre de couches (figure A.5) : un modèle de domaine (DM), un modèle de buts et contraintes (GM), un modèle utilisateur (UM), un modèle d'adaptation (AM) et un modèle de présentation (PM).
Le modèle de domaine est composé de schémas conceptuels composés de concepts, relations et attributs attachés aux concepts. Ce modèle représente les ressources pour l'apprentissage et leurs caractéristiques. Le modèle de buts et

des contraintes filtre, regroupe et restructure le modèle de domaine en fonction des objectifs d'instruction. L'interprétation réelle de cette structure est assurée par le modèle d'adaptation. Le modèle utilisateur représente l'utilisateur sous la forme d'un schéma. Le modèle d'adaptation est découpé en trois niveaux : un niveau dit, bas niveau, permettant d'agir directement sur les éléments à adapter par les techniques courantes d'adaptation ; un niveau intermédiaire permettant de grouper les éléments du niveau précédent pour créer des mécanismes d'adaptation et de construction, il en résulte un langage de programmation ; un niveau dit, haut niveau, permettant l'utilisation de fonctions. Enfin, le modèle de présentation prend en compte les propriétés physiques et l'environnement de présentation pour la génération réelle de code sur différentes plateformes. Dans sa construction, le modèle de présentation est similaire au modèle utilisateur. Il combine une sur-couche sur le modèle de domaine, le modèle de buts et des contraintes, avec des variables indépendantes et leurs valeurs respectives. Cette structure permet d'attacher à chaque concept un certain type de représentation à l'écran, tout en conservant en même temps des types de représentation indépendants associés aux valeurs courantes du modèle de l'utilisateur.

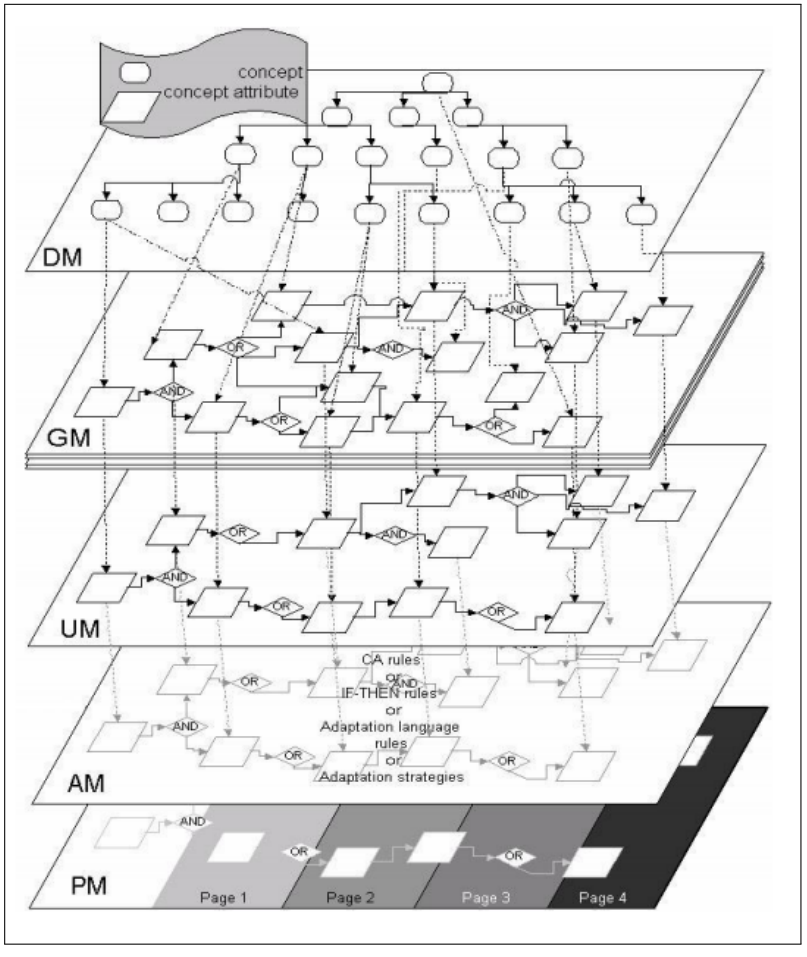

FIGURE A.5 – Modèle LAOS (Cristea and Andmethods, 2003)

Modèle ALEM :
Le modèle ALEM (Adaptive Learning Environment Model) (Tadlaoui et al., 2010) est une extension du modèle de référence Munich. Le modèle Munich, est qualifié comme modèle de base à cause de son approche de modélisation. Quoique l'architecture du modèle ALEM contient les mêmes trois couches existantes dans le modèle Munich, elle étend leurs fonctionnalités pour une meilleure modélisation des systèmes e-learning.

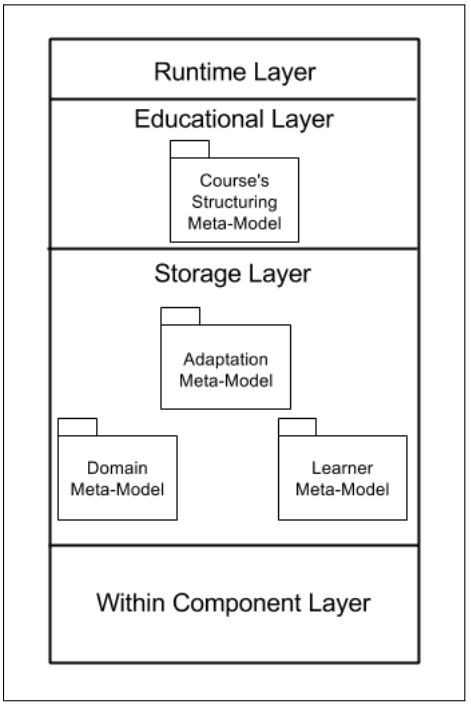

FIGURE A.6 – Modèle ALEM (Tadlaoui et al., 2010)

Le modèle intègre une couche supplémentaire qui est la couche pédagogique (educational layer). Les différentes couches du modèle ALEM, représentées dans la figure A.6, sont les suivantes : La couche contenu de composant (within component layer) contient le contenu et la structure des nœuds de l'hypermédia. La couche de stockage (storage layer) stocke les informations sur la structure de l'hypermédia. Cette couche est composée de trois méta-modèles : le méta-modèle du domaine (domain meta-model) décrivant le domaine d'application de l'hypermédia, le méta-modèle de l'apprenant (learner meta-model) décrivant les caractéristiques de l'apprenant utiles pour la personnalisation de l'hyper-espace, et finalement le méta-modèle de l'adaptation (adaptation meta-model) décrivant les stratégies et les règles d'adaptation. La couche pédagogique est une représentation abstraite du cours. Cette couche contient le modèle de structuration des cours. La dernière couche est la couche d'exécution (runtime layer) décrit la manière de présenter les nœuds. Cette couche est aussi responsable

de l'interaction avec l'apprenant, l'acquisition des données de l'apprenant et la gestion des cours.
L'apport essentiel de ce modèle est la modélisation du cours et du processus pédagogique, afin de proposer des cours personnalisés à l'apprenant.

Synthèse des architectures hypermédia adaptatifs :
D'après la description de ces différents modèles de référence, nous remarquons qu'on ne peut toujours pas définir un modèle générique des systèmes hypermédia adaptatifs, car il n'y a pas encore unanimité quant à l'architecture idéale de ces systèmes. Mais nous pouvons à partir de ces différents modèles extraire une architecture minimale. Cette architecture minimale se compose essentiellement de trois parties :

– *Le modèle de domaine :* Ce modèle permet de mettre à la disposition d'un utilisateur particulier des données adaptés à ses besoins. Ces données auxquelles l'utilisateur peut accéder sont souvent hétérogènes (page Web entière, fragments de page Web, un paragraphe, une image, etc.).
– *Le modèle utilisateur :* Comme expliqué précédemment, l'approche dans les hypermédia adaptatifs est axée principalement sur l'utilisateur. C'est le modèle utilisateur qui va aider à mettre en place la dynamique du système. Il faut donc que le système puisse conserver une certaine connaissance sur l'utilisateur à des fins d'adaptation.
– *Le modèle d'adaptation :* Afin de fournir aux utilisateurs des informations dynamiques et adaptées à leur profil, il faut mettre en place des mécanismes d'adaptation permettant de lui présenter les informations stockées dans le modèle de domaine. Ces mécanismes d'adaptation décrivent comment le système hypermédia adaptatif va se comporter par rapport un profil utilisateur. Le modèle d'adaptation peut être défini comme un ensemble de règles d'adaptation ou comme un ensemble d'algorithmes, déterminant le comportement du système vis-à-vis d'un profil utilisateur.

La spécification de la présentation est générée à partir du modèle d'adaptation et plus particulièrement des informations sélectionnées du modèle de domaine. Elle permet de définir, à partir de la couche de sauvegarde, comment organiser et présenter les informations.
Enfin, la couche exécutive est spécifique à une application. Elle est responsable des mécanismes d'interactions avec l'utilisateur.
Cette division en couches et en modèles fournit une séparation claire des parties majeures des systèmes hypermédia adaptatifs. Cette architecture minimale est nécessaire et suffisante pour la création d'un système hypermédia adaptatif. Ce-

pendant, selon le niveau d'adaptation, de flexibilité et de portabilité désirée, il est possible d'ajouter des composantes supplémentaires à cette architecture.

A.3/ AVANTAGES ET INCONVÉNIENTS

Les hypermédia adaptatifs ont été appliqués à de nombreux domaines, mais principalement au domaine éducatif. Il est clair que les hypermédia adaptatifs représentent un avantage non négligeable par rapport aux hypermédias classiques. Comme l'explique (Delestre et al., 1998) : Tout d'abord, les différentes techniques utilisées permettent à l'apprenant d'être guidé dans son apprentissage. Ainsi sans toutefois annihiler la liberté de navigation intrinsèque aux hypermédias, l'apprenant est constamment guidé dans son cheminement. Ensuite, l'utilisation d'un système interactif multimédia peut améliorer l'aspect visuel et ludique, et par conséquent renforcer l'intérêt de l'apprenant.
Cependant, quelques problèmes des hypermédias persistent : Tout d'abord, l'accent a surtout été mis sur l'adaptation des liens, afin de guider l'apprenant dans sa navigation. Or la deuxième composante de l'adaptation, à savoir l'adaptation du contenu, a souvent été mise de côté. De plus, dans un système d'apprentissage à distance, il faut que le système puisse utiliser immédiatement toute nouvelle information rajoutée par les apprenants ou les formateurs, pour présenter une nouvelle connaissance. Or, dans un volume de données en constante évolution, on remarque que le problème de surcharge cognitive persiste toujours.
Un domaine d'application des hypermédia adaptatifs, permettant de résoudre ces problèmes, est le domaine des systèmes de recommandation.

B

LA LOGIQUE DE DESCRIPTION

La logique descriptive est une famille de formalismes utilisés pour représenter une base de connaissances d'un domaine d'application d'une façon structurée et formelle (Baader et al., 2003). Les logiques de description ont deux objectifs : Tout d'abord représenter les connaissances d'un domaine, c'est donc la partie description ; et ensuite raisonner à partir de ces connaissances, c'est la partie logique.
Dans cette annexe nous traitons la logique descriptive ainsi que les aspects de raisonnement logique. Leur utilisation conjointe permet la définition de bases de connaissances, puis à partir de ces connaissances la déduction de nouveaux faits.

B.1/ LES LANGAGES FORMELS ET LE WEB SÉMANTIQUE

Les logiques de descriptions sont utilisées pour de nombreuses applications, parmi elles on cite les domaines suivants : traitement automatique des langues, l'ingénierie logicielle (représentation de la sémantique des diagrammes de classe UML) et le Web sémantique pour la représentation d'ontologies. Le courant des recherches, qui s'est nourri d'études effectuées sur la logique des prédicats et les réseaux sémantiques a donné naissance à une famille de langages de représentation appelés logiques de descriptions $\mathcal{LD}$. Ces langages ont été introduits dans les années 80 dans le but de rendre la représentation de connaissances plus naturelle qu'en logique du premier ordre.
Une caractéristique fondamentale des logiques de description est qu'elles ont une sémantique descriptive formelle conforme au cadre des logiques de (Tarski, 1941). Les connaissances d'un domaine y sont représentées par des entités qui correspondent à une description syntaxique à laquelle est associée une sémantique :
(1) Les éléments du monde réel y sont représentés à l'aide de concepts, rôles et

individus. Les concepts ainsi que les rôles sont organisés de façon hiérarchique à l'aide de relations de subsomption. L'utilisation de ce type de logique permet la réalisation de taches de raisonnement.
(2) La sémantique des logiques de description est clairement définie et associée à la description des concepts ainsi que des rôles. Cette sémantique affecte la façon de manipuler ces éléments. Cette sémantique est définie comme suit :
Soit $CON = \{C1, C2, \ldots\}$ un ensemble fini de concepts atomiques, $ROL = \{R1, R2, \ldots\}$ un ensemble fini de rôles atomiques et $IND = \{a1, a2, \ldots\}$ un ensemble fini d'individus. Si CON, ROL, IND sont disjoints deux à deux, $\mathcal{S} = \langle CON, ROL, IND\rangle$ est une signature. Une fois qu'une signature $\mathcal{S}$ est fixée, une interprétation $\mathcal{I}$ pour $\mathcal{S}$ est défini comme un couple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$, où :

– $\Delta^{\mathcal{I}}$ est un ensemble non vide, le domaine de l'interprétation.
– $\cdot^{\mathcal{I}}$ est une fonction affectant :
  – un élément $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ à chaque individu $a_i \in IND$,
  – un sous-ensemble $C_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ à chaque concept atomique $C_i \in CON$, et une relation $R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ à chaque rôle atomique $R_i \in ROL$.

B.1.1/ LES BASES DE CONNAISSANCES

Une base de connaissances est composée d'une partie terminologique, qualifié de *TBox* (Terminology Box), et d'une partie assertionnelle, qualifié de *ABox* (Assertions Box). La *ABox* décrit les individus et leurs relations (quel individu appartient à quel concept nommé, quel individu est lié à quel autre à travers quel rôle). Typiquement, la base de connaissances standard utilisée par les logiques de description est définie de la manière suivante :
Étant donné un langage de description $\mathcal{L}$ et une signature $\mathcal{S}$, une base de connaissances Σ dans $\mathcal{L}$ est une paire $\Sigma = \langle T, A\rangle$ telle que :

– $\mathcal{T}$ la *TBox*, est un ensemble fini, qui peut être vide, d'expressions appelées GCI (General Concept Inclusion) de la forme $C \sqsubseteq D$ où C et D sont des concepts sans restriction [1]. $C \doteq D$ est une notation pour $C \sqsubseteq D$ et $D \sqsubseteq C$.
  Une interprétation $\mathcal{I}$ d'une *TBox* satisfait $C \sqsubseteq D$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ et elle satisfait $C \equiv D$ si $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$.
– $\mathcal{A}$ la *ABox*, un ensemble fini, qui peut être vide, d'expressions de la forme $C(a)$ ou $R(a, b)$, où C est un concept sans restriction, R est un rôle qui n'est pas nécessairement atomique, et a,b appartiennent à IND. Les formules de *A* sont appelées des assertions.
  Une interprétation $\mathcal{I}$ d'une *ABox* associe à chaque constante a un élément $a^{\mathcal{I}}$

1. http ://fr.wikipedia.org/wiki/Logique_de_description

de $\Delta^{\mathcal{I}}$. $\mathcal{I}$ est un modèle de la *ABox* si pour toute assertion $C(a) \Rightarrow a^{\mathcal{I}} \in C^{\mathcal{I}}$ et $R(a,b) \Rightarrow (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Un modèle d'une base de connaissance est une interprétation qui est à la fois un modèle de la *TBox* et de la *ABox*. Exemple, si $\mathcal{T} = \{C \sqsubseteq D, C \equiv \exists R.E\}$ et $\mathcal{A} = \{C(a), R(a,b), R(b,c), E(b), E(c)\}$ on peut avoir le modèle $\mathcal{I}$ suivant : $C^{\mathcal{I}} = \{a,b\}, D^{\mathcal{I}} = \{a,b\}, E^{\mathcal{I}} = \{b,c\}, R^{\mathcal{I}} = \{(a,b),(b,c)\}$ (pour simplifier on définit $a^{\mathcal{I}} = a, b^{\mathcal{I}} = b, \ldots$).

B.1.2/ LES FAMILLES DE LOGIQUES DE DESCRIPTION

Les $\mathcal{LD}$ forment une famille de langages de représentation de connaissances. Ces $\mathcal{LD}$ sont utilisés pour représenter les connaissances d'un domaine d'application d'une façon structurée et formelle. Ces $\mathcal{LD}$ sont caractérisés par un ensemble de symboles et par des règles de construction qui permettent d'obtenir des formules à partir de ces symboles et moyennant des constructeurs spécifiques au langage. Une $\mathcal{LD}$ possède une signature, qui est basée sur les types : concepts, rôles et individus.
Dans le formalisme des $\mathcal{LD}$, un concept représente un ensemble d'individus et un rôle représente une relation binaire entre individus. Un concept correspond à une entité générique d'un domaine d'application et un individu à une entité particulière, instance d'un concept.
Concepts, rôles et individus obéissent aux principes suivants :

- Un concept et un rôle possèdent une description structurée, élaborée à partir d'un certain nombre de constructeurs. Une sémantique est associée à chaque description de concept et de rôle par l'intermédiaire d'une interprétation.
- Les connaissances sont prises en compte selon plusieurs niveaux : la représentation et la manipulation des concepts et des rôles relèvent du niveau terminologique *TBox* alors que la description et la manipulation des individus relèvent du niveau factuel ou niveau des assertions *ABox*.
- La relation de subsomption permet d'organiser concepts et rôles par niveau de généralité : intuitivement, un concept *C* subsume un concept *D* si *C* est plus général que *D* au sens où l'ensemble d'individus représenté par *C* contient l'ensemble d'individus représenté par *D*. Une base de connaissances se compose alors d'une hiérarchie de concepts et d'une (éventuelle) hiérarchie de rôles.

Exemple : Cet exemple décrit les relations entre membres d'une famille :

- Concepts : *Femme*, *Homme*, etc.
- Rôles : *epousDe*, *pereDe*, etc.
- Individus : *marie*, *pierre*, *jean*, etc.

Un concept et un rôle possèdent une description structurée définie à partir d'un certain ensemble de constructeurs. Concepts, rôles et individus de l'exemple précédent sont décrits comme suit au niveau de la *TBOx* et de la *ABox* :

– *TBox*, décrit les concepts :

$$\textit{Femelle} \sqsubseteq \top \sqcap \neg\, \textit{Male}$$
$$\textit{Male} \sqsubseteq \top \sqcap \neg\, \textit{Femelle}$$
$$\textit{Animal} \equiv \textit{Male} \sqcup \textit{Femelle}$$
$$\textit{Humain} \sqsubseteq \textit{Animal}$$
$$\textit{Femme} \equiv \textit{Humain} \sqcap \textit{Femelle}$$
$$\textit{Homme} \equiv \textit{Humain} \sqcap \neg\, \textit{Femelle}$$

– *ABox*, fournit des instances des concepts et des rôles :

marie : Femme
pierre : Homme
jean : Homme
epouseDe(pierre, marie)
pereDe(jean, pierre)

Il existe plusieurs familles de langages de description des concepts, des rôles et des individus. La logique de description $\mathcal{AL}$ (Attribute Language) est la logique minimale, dite aussi logique attributive, et elle a été définie par (Schmidt-Schaub and Smolka, 1991). Elle est dite minimale car une logique d'expressivité inférieure n'aurait pas d'intérêt pratique (Baader et al., 2003).
Les descriptions possibles dans le langage $\mathcal{AL}$ sont présentées par le tableau A.1 (on suppose que A est un concept atomique et C et D sont des concepts atomiques ou complexes) :

Syntaxe	**Définition**
A	Concept atomique
$\top$	Le concept universel Top
$\bot$	Le concept vide Bottom
$\neg A$	Négation complète
$C \sqcap D$	Conjonction de concepts
$\forall R.C$	Quantificateur universel
$\exists R.\top$	Quantificateur existentiel non typé

TABLE B.1 – La grammaire du langage de description de concepts selon $\mathcal{AL}$

Le langage de base $\mathcal{AL}$ est défini à partir des éléments syntaxiques suivants :

– Le concept *Top* ($\top$) dénote le concept le plus général et le concept *Bottom* ($\bot$)

le concept le plus spécifique. Intuitivement, l'extension de Top inclut tous les individus possibles tandis que celle de Bottom est vide.
– Le constructeur *not* ($\neg$) correspond à la négation et ne porte que sur les concepts primitifs (atomiques). Le constructeur *and* ($\sqcap$) permet de définir une conjonction d'expressions conceptuelles.
– La quantification universelle *all* ($\forall R.C$) précise le co-domaine du rôle R et la quantification existentielle non typée *some* ($\exists R$) introduit le rôle R et affirme l'existence d'(au moins) un couple d'individus en relation par l'intermédiaire de R.

On obtient alors le langage $\mathcal{AL} = \{\top, \bot, A, \neg A, C \sqcap D, \forall r.C, \exists r\}$ qui peut être par la suite enrichi par plusieurs autres constructeurs.
La sémantique du langage $\mathcal{AL}$ fait appel à la théorie des ensembles. Essentiellement, à chaque concept est associé un ensemble d'individus dénotés par ce concept. Une interprétation suppose donc l'existence d'un ensemble non vide Δ qui représente des entités du monde décrit. Une fonction d'interprétation (décrite précédemment) associe à une concept C un sous-ensemble $C^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$ et à un rôle R un sous-ensemble $R^{\mathcal{I}}$ de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$(Gagnon, 2009).
La sémantique de $\mathcal{AL}$ est définit alors comme suit :

Syntaxe	**Sémantique**
A	$A^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
$\neg A$	$\Delta^{\mathcal{I}}\ C^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
$\exists R.\top$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}}\}$

TABLE B.2 – La sémantique du langage de description de concepts selon $\mathcal{AL}$

La logique $\mathcal{AL}$ n'est généralement pas suffisamment expressive pour représenter certaines connaissances. Il existe alors plusieurs autres constructeurs que l'on peut ajouter à ce langage pour le rendre plus expressif. Notamment, la logique $\mathcal{ALC}$ qui est l'extension de la logique $\mathcal{AL}$ par la négation complète C (négation de concepts non primitifs, définis). $\mathcal{ALC}$ constitue la logique descriptive de base : $\mathcal{ALC} = \mathcal{ALC} \cup \{\neg C\}$.
Les différentes logiques de description qui existent aujourd'hui sont des combinaisons des différents constructeurs du tableau A.3 avec la logique de base $\mathcal{ALC}$:

Lettre	Définition
$\mathcal{F}$	Rôles fonctionnels
$\mathcal{E}$	Quantification existentielles typé
$\mathcal{U}$	Disjonction de rôles
$\mathcal{C}$	Négation complète
$\mathcal{R}^+$	Transitivité de rôles
$\mathcal{S}$	Abréviation de la logique $\mathcal{ALC}$ avec la transitivité des rôles $\mathcal{R}^+$
$\mathcal{H}$	Hiérarchie de rôles
$\mathcal{O}$	Classes nominales ou énumération par individus
$\mathcal{I}$	Rôles inverses
$\mathcal{N}$	Restrictions de cardinalité sur les rôles
$\mathcal{Q}$	Restrictions de quantifieur de cardinalité sur les rôles
$\mathcal{R}$	Conjonction de rôles

TABLE B.3 – Constructeurs des logiques de description

Le tableau A.4 décrit la syntaxe et la sémantique des différents constructeurs des logiques de description :

Constructeurs	Syntaxe	Sémantique
$\mathcal{F}$	$\leq 1R$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid \leq 1\}$
$\mathcal{E}$	$\exists R.C$	$\{x \in \Delta^I \mid \exists y \in \Delta^I, (x,y) \in R^I \Rightarrow y \in C^I\}$
$\mathcal{U}$	$C_1 \sqcup C_2$	$C_1^I \cup C_2^I$
$\mathcal{C}$	$\neg C$	$\Delta^I \setminus \cap C^I$
$\mathcal{R}^+$	R^+	Plus petite relation transitive contenant R^I
$\mathcal{H}$	$R_1 \sqsubseteq R_2$	$R_1^I \subseteq R_2^I$
$\mathcal{O}$	$\{a_1, \ldots, a_n\}$	$\{x \in \Delta^I \mid x = a_i$ pour un $a_i\}$
$\mathcal{I}$	R^{-1}	$\{(x,y) \in R^I \mid (y,x) \in R^I\}$
$\mathcal{N}$	$\geq nR$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid \geq n\}$
$\mathcal{N}$	$\leq nR$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid \leq n\}$
$\mathcal{N}$	$= nR$	$\{x \in \Delta^I \mid y, (x,y) \in R^I \mid = n\}$
$\mathcal{Q}$	$\geq nR.C$	$\{x \in \Delta^I \mid y \in C^I, (x,y) \in R^I \mid \geq n\}$
$\mathcal{Q}$	$\leq nR.C$	$\{x \in \Delta^I \mid y \in C^I, (x,y) \in R^I \mid \leq n\}$
$\mathcal{Q}$	$= nR.C$	$\{x \in \Delta^I \mid y \in C^I, (x,y) \in R^I \mid = n\}$
$\mathcal{R}$	$R_1 \sqcap R_2$	$R_1^I \cap R_2^I$

TABLE B.4 – Constructeurs : syntaxe et sémantique

Nous pouvons ainsi obtenir d'autres langages plus expressifs, tels qu'avec l'utilisation des constructeurs suivants :

– Le constructeur $\mathcal{U}$ permet de définir la disjonction (and) de concepts, notée $C \cup D$. L'extension correspondante d'$\mathcal{ALC}$ est $\mathcal{ALCU} = \mathcal{ALC} \cup \{C \cup D\}$.
– Le constructeur $\mathcal{R}$ permet de définir la conjonction (or) de rôles, notée $R_1 \cap R_2$, les rôles R_1 et R_2 étant primitifs. L'extension correspondante d'$\mathcal{ALC}$ est $\mathcal{ALCR} = \mathcal{ALC} \cup \{R_1 \cap R_2\}$.
– La quantification existentielle typée $\mathcal{E}$, notée $\exists R.C$ (la quantification $\exists R.C$ introduit un rôle R de co-domaine C et impose l'existence d'(au moins) un couple d'individus (x, y) en relation par l'intermédiaire de R, où C est le type de y). L'extension correspondante d'$\mathcal{ALC}$ est $\mathcal{ALCE} = \mathcal{ALC} \cup \{\exists r.C\}$.
– Le constructeur $\mathcal{N}$, notée $\geq nR$ (atleast nR) et $\leq nR$ (atmost nR), fixent la cardinalité minimale et maximale du rôle auquel ils sont associés. L'extension correspondante d'$\mathcal{ALC}$ est $\mathcal{ALCN} = \mathcal{ALC} \cup \{\geq nR, \leq nR\}$.

Certaines logiques sont équivalentes, notamment $\mathcal{ALC}$ et $\mathcal{ALUE}$. Ces deux dernières logiques augmentées par $\mathcal{R}^+$ sont notées $\mathcal{S}$.

Toutes les logiques de description qui existent sont des combinaisons des différents constructeurs du tableau A.4. On obtient essentiellement les familles suivantes :

Composition	**Famille**
$\mathcal{AL}$	Logique de base
$\mathcal{ALC}$	Logique minimale
$\mathcal{ALC} + \mathcal{N}$	Logique $\mathcal{ALCN}$
$\mathcal{ALC} + \mathcal{Q}$	Logique $\mathcal{ALCQ}$
$\mathcal{ALC} + \mathcal{F}$	Logique $\mathcal{ALCF}$
$\mathcal{ALC} + \mathcal{R}^+$	Logique $\mathcal{S}$
$\mathcal{S} + \mathcal{H}$	Logique $\mathcal{SH}$
$\mathcal{SH} + \mathcal{I} + \mathcal{F}$	Logique $\mathcal{SHIF}$
$\mathcal{SH} + \mathcal{I} + \mathcal{Q}$	Logique $\mathcal{SHIQ}$
$\mathcal{SH} + \mathcal{O} + \mathcal{I} + \mathcal{N}$	Logique $\mathcal{SHOIN}$
$\mathcal{SH} + \mathcal{O} + \mathcal{I} + \mathcal{Q}$	Logique $\mathcal{SHOIQ}$

TABLE B.5 – Familles de logique de description

B.1.3/ LE LANGAGE ONTOLOGIQUE OWL

Le langage OWL, proposé par le consortium W3C, est le langage de représentation des connaissances permettant la définition d'ontologies. Il est basé sur les travaux ayant eu lieux dans le domaine de la logique de description précédemment introduite. Cela permet la mise en place de processus de

vérification de cohérence ainsi que de déduction de connaissances à partir des connaissances existantes.
Le langage OWL est composé de trois sous-langages : OWL-Lite, OWL-DL et OWL-Full. OWL-Lite correspond essentiellement à la famille $\mathcal{SHIF}$ alors que OWL-DL correspond essentiellement à la famille $\mathcal{SHOIN}$. En fait, plus précisément, il s'agit des familles $\mathcal{SHIF}(\mathcal{D})$ et $\mathcal{SHOIN}(\mathcal{D})$, parce qu'on y distingue deux types distincts de rôles : les rôles qui lient deux individus, tel que vu tout au long du présent rapport, et les rôles qui associent un individu avec un type primitif (entier, chaine de caractère, etc.), d'où le $\mathcal{D}$, pour Data Property. Les sous-langages OWL-Lite et OWL-DL ne sont pas des extensions de RDF, dans le sens qu'un triplet RDF n'est pas nécessairement valide dans ces deux sous-langages. C'est pour cette raison que le sous-langage OWL-Full a été ajouté, qui comprend tout OWL-DL, avec en plus tout RDF.
En 2009, le consortium W3C a officiellement lancé OWL 2, qui se distingue de la première version par un pouvoir expressif augmenté et une élimination de la décomposition en trois sous-langages. Ainsi OWL-Full n'existe plus, alors que OWL-Lite et OWL-DL sont considérés comme des profils de OWL 2 (on parle de profil lorsqu'on élimine de OWL 2 certains constructeurs limitant sont pouvoir expressif). Depuis 2012, OWL 2 est recommandé par W3C et permet l'expressivité de la logique $\mathcal{SROIQ}(\mathcal{D})$ (Gagnon, 2009).
L'évolution de l'expressivité depuis les logiques descriptives les moins expressives jusqu'aux logiques les plus complexes utilisées dans le domaine du Web sémantique est présentée dans la figure ci-dessous.

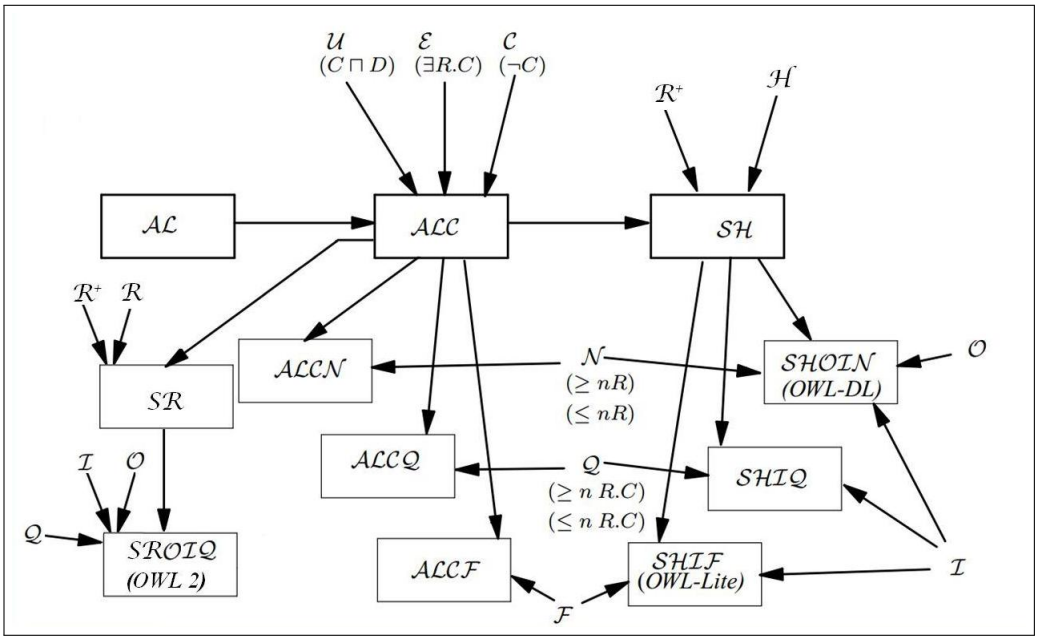

FIGURE B.1 – Évolution des logiques de description

Les langages OWL-DL et OWL-Lite se basent respectivement sur les logique $\mathcal{SHOIN}$ et $\mathcal{SHIF}$ qui sont une augmentation de l'expressivité de la logique $\mathcal{SH}$, elle-même basée sur la logique $\mathcal{S}$ à laquelle la gestion des hiérarchies de rôles a été ajoutée. Tandis que le langage OWL 2 a l'expressivité de la logique $\mathcal{SROIQ}$ qui est une augmentation de l'expressivité de la logique $\mathcal{SR}$ basé sur la logique $\mathcal{S}$ à laquelle contrairement la logique $\mathcal{SH}$ les inclusion de rôles R ont été ajoutés en plus des hiérarchies de rôles.

Constructeurs	**Syntaxe**	**Exemple**
intersectionOf	$C_1 \sqcap C_2$	$Humain \sqcap Homme$
unionOf	$C_1 \sqcup C_2$	$Docteur \sqcup Avocat$
complementOf	$\neg C$	$\neg Homme$
oneOf	$\{a_1 ... a_2\}$	$\{marie, pierre, jean\}$
allValueFrom	$\forall R.C$	$\forall possedeEnfant.Docteur$
someValueFromFrom	$\exists R.C$	$\exists possedeEnfant.Avocat$
hasValue	$\exists R.\{a\}$	$\exists citoyenDe.\{France\}$
minCardinality	$(\geq nR)$	$(\geq\ 2\ possedeEnfant)$
maxCardinality	$(\leq nR)$	$(\leq\ 1\ possedeEnfant)$
inverseOf	R^-	$possedeEnfant^-$

TABLE B.6 – Constructeurs OWL

Axiome	**Syntaxe**	**Exemple**
subClassOf	$C_1 \sqsubseteq C_2$	$Humain \sqsubseteq Animal \sqcap Bipede$
equivalentClass	$C_1 \equiv C_2$	$Homme \equiv Humain \sqcap Male$
subPropertyOf	$R_1 \sqsubseteq R_2$	$possedeFille \sqsubseteq possedeEnfant$
equivalentProperty	$R_1 \equiv R_2$	$Cout \equiv Prix$
disjointWith	$C_1 \sqsubseteq \neg C_2$	$Male \sqsubseteq \neg Femelle$
sameAs	$\{a_1\} \equiv \{a_2\}$	$\{president_hollande\} \equiv \{françois_hollande\}$
differentFrom	$\{a_1\} \sqsubseteq \neg\{a_2\}$	$\{marie\} \sqsubseteq \neg\{jean\}$
transitiveProperty	R rôle transitif	$estAncetreDe$ rôle transitif
functionalProperty	$\top \sqsubseteq (\leq 1R)$	$\top \sqsubseteq (\leq 1\ estMereDe)$
inverseFunctionalProperty	$\top \sqsubseteq (\leq 1R^-)$	$\top \sqsubseteq (\leq 1\ aPourMere^-)$
symmetricProperty	$R \equiv R^-$	$estMarieAvec \equiv estMarieAvec^-$

TABLE B.7 – Axiomes OWL 2

Les deux tableaux Ci-dessus présentent les constructeurs du langage OWL et les axiomes du langage OWL 2 ainsi que leur correspondance en logique de description.

Différentes syntaxes peuvent être utilisées afin de sérialiser une ontologie OWL, par exemple en Turtle ou en XML.
L'exemple 1 décrit la syntaxe XML pour la définition d'une classe étant l'intersection des classes *Homme* et *Humain* (*Humain* ⊓ *Homme*) :

```
<owl:Class>
       <owl:intersectionOf rdf:parserType="Collection">
              <owl:Class rdf:about="#Humain"/>
              <owl:Class rdf:about="#Homme"/>
       </owl:intersectionOf>
</owl:Class>
```

L'exemple 2 décrit la syntaxe XML pour la définition d'une restriction sur la cardinalité minimum d'un rôle (≥ *possede-enfant.Thing*) :

```
<owl:Restriction>
       <owl:OnProperty rdf:Ressource="#possede-enfant"/>
       <owl:minCardinality rdf:datatype="&xsd;NonNegativeInterger"> 2
       <owl:minCardinality>
</owl:Restriction>
```

B.2/ LES RAISONNEURS

La construction et l'implémentation d'une ontologie peut amener à la détection de contradiction dans le modèle sous-jacent. Une des tâches d'un raisonneur est de vérifier qu'il n'y a pas de contradiction et que le modèle associé à l'ontologie est bien cohérent. Un raisonneur peut invalider de différente manière une ontologie :

– Une ontologie peut être détectée comme inconsistante ou incohérence : c'est-à-dire qu'il n'existe aucune interprétation possible de l'ontologie qui soit un modèle pour cette ontologie.
– Une ontologie peut être détectée comme insatisfiable : c'est-à-dire qu'il existe une interprétation possible de cette ontologie qui soit un modèle, mais que ce modèle sous-jacent associe à certaines concepts un ensemble d'individus qui sera toujours vide. C'est-à-dire qu'aucun individu ne pourra jamais être une instance de ces concepts.

Les tâches de raisonnement peuvent être exprimées en termes de vérifications de consistance et de satisfiabilité.

Les raisonneurs offrent également des services d'inférence dont la charge principale du raisonnement dans une logique de description est de découvrir des connaissances implicites à partir des connaissances explicites par l'inférence en plus de la vérification de la consistance et de satisfiabilité.

B.2.1/ INFÉRENCE

L'inférence s'effectue dans les logiques descriptives au niveau terminologique, c'est à dire au niveau de la TBox, ou bien au niveau assertionnel en prenant en compte les individus de la base de connaissances au niveau de la ABox.

Inférence au niveau de la TBox :
D'après (Baader et al., 2003) quatre cas d'inférence au niveau terminologique peuvent être distingués :

- *Statisfiabilité* : Un concept C d'une terminologie TBox T est satisfiable si et seulement si il existe un modèle (une interprétation) $\mathcal{I}$ de T tel que $C^{\mathcal{I}} \neq \emptyset$. Ce qui signifie qu'il existe une ABox non vide pour le domaine d'interprétation $\delta^{\mathcal{I}}$ tel que le concept C d'une terminologie T possède au moins un individus pour le modèle d'interprétation $\mathcal{I}$. Par exemple, le concept $Homme \sqcap \neg Homme$ est insatisfaisable, puisque l'intersection d'un ensemble avec son complément est toujours l'ensemble vide. Il est donc impossible qu'une entité du monde appartienne à ce concept.
- *Subsomption* : Un concept C est subsumé par un concept D pour une terminologie T si et seulement si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de T. Par exemple, si une terminologie T contient l'axiome $Mere \equiv Femme \sqcap \exists possedeEnfant.Personne$, il est relativement facile de démontrer la subsomption suivante : $Mere \sqsubseteq Femme$. Autrement dit, si un individu appartient à l'ensemble des mères, alors il appartient aussi à l'ensemble des femmes.
- *Équivalence* : Un concept C est équivalent à un concept D pour une terminologie T si et seulement si $C^{\mathcal{I}} = D^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de T. Dans ce cas, on écrira $T \models C \equiv D$.
- Disjonction : Les concepts C et D sont disjoint par rapport à la terminologie T si et seulement si $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ pour tout modèle $\mathcal{I}$ de T. Dans ce cas, on écrira $T \models C \sqcap D \sqsubseteq \bot$. Par exemple, on peut démontrer que les concepts $Pere$ et $Mere$ sont disjoints. En effet, selon la définition du concept $Homme$, un homme fait partie du complément de l'ensemble des femmes. Ainsi, une entité ne peut être à la fois un homme et une femme, ce qui implique qu'il ne peut pas être à la fois un père et une mère, puisque par définition un père est aussi un homme, et une mère est aussi une femme.

En fait, la vérification de la satisfaisabilité de concept est une inférence principale. Dans (Baader et al., 2003) les autres inférences pour les concepts peuvent être réduites à la satisfaisabilité et inversement.

Inférence au niveau de la ABox :
D'après (Baader et al., 2003) quatre cas d'inférence au niveau assertionel peuvent être distingués :

– *Cohérence* : Une *Abox* A est cohérente par rapport à une *Tbox* T si et seulement si, il existe un modèle $\mathcal{I}$ de A et T.
– *Vérification d'instance* : Elle consiste à vérifier si un individu a d'une *ABox* A est une instance d'une description de concept donnée C ($a \in C^{\mathcal{I}}$). L'insertion est vérifiée si elle est vraie pour tout modèle $\mathcal{I}$ de la *Abox* A et de la *Tbox* T.

B.2.2/ APPROCHES DE RAISONNEMENT

A partir de la technique employée dans le développement des algorithmes de décision pour le problème d'inférence, nous pouvons les catégoriser en deux groupes :
Le premier groupe contient des algorithmes appelés algorithmes structurels. Ces algorithmes comparent la structure syntaxique des concepts, pour résoudre le problème de subsomption de concept dans quelques logiques de description primitives. Ces algorithmes, toutefois, ne sont pas applicables pour les logiques de description plus complexes avec la négation, la disjonction, etc. Pour de tels langages d'autres types d'algorithmes ont été créés, que nous nous catégorisons dans le deuxième groupe.
Le deuxième groupe contient des algorithmes appelés algorithmes de tableaux. Les algorithmes de tableaux ont été donnés la première fois par (Schmidt-Schaub and Smolka, 1991) et sont aujourd'hui l'outil principal pour la résolution des problèmes de satisfaisabilité et de subsomption de concept dans les logiques de description.
Dans ce qui suit nous décrivons brièvement ces deux catégories d'algorithmes.

Raisonnement basé sur la comparaison structurelle :
Les algorithmes de raisonnement basé sur la comparaison structurelle ne sont applicables que sur des logiques primitives de type $\mathcal{FL}^-$ (équivalent à $\mathcal{ALC}$ sans la négation des concepts atomiques). Les algorithmes du calcul de subsomption sont basés sur la comparaison structurelle entre les expressions de concepts. L'idée de cette approche est que si deux expressions de concept sont faites de

sous-expressions, alors elles peuvent être comparées séparément en comparant une sous-expression d'un concept avec toutes celles de l'autre.
Afin de vérifier la subsomption dans la logique $\mathcal{FL}^-$ l'algorithme s'exécute en deux phases : Premièrement, les concepts sont réécrits dans une forme normale (déplier les concepts et factoriser les rôles), et ensuite leurs structures sont comparées :

– Toutes les conjonctions emboîtées sont égalisées : $A \sqcap (B \sqcap C) \Leftrightarrow A \sqcap B \sqcap C$.
  Toutes les conjonctions de quantifications universelles sont factorisées : $\forall R.C \sqcap \forall R.D \Leftrightarrow \forall (C \sqcap D)$.
  Les concepts réécrits sont logiquement équivalents avec les précédents, donc la subsomption est préservée par cette transformation.
– Soient $C = C_1 \sqcap C_2 \sqcap \ldots \sqcap C_m$ et $D = D_1 \sqcap D_2 \sqcap \ldots \sqcap D_n$, alors D subsume C si et seulement si pour chaque D_i , il existe un C_j avec :
  (1) Si D_i est un concept atomique, ou est bien de la forme $\exists R$, alors $D_i = C_j$.
  (2) Si D_i est un concept de forme $\forall R.D'$, $C_j = \forall R.C'$ (le même rôle atomique R), alors $C' \sqsubseteq D'$.

Raisonnement basé sur l'algorithme de tableaux :
Cette technique de raisonnement est basée sur les calculs de tableaux pour la logique des prédicats du premier-ordre. Les structures de tableaux obtenues en raisonnant avec un langage donné de la logique de description sont soigneusement analysées, et les vérifications redondantes dans les tableaux sont éliminées afin de donner une limite supérieure stricte sur la complexité de la méthode.
L'idée principale du calcul de tableau est de vérifier si une formule donnée F est une conséquence logique d'une théorie donnée T. On essaye de construire, en utilisant des règles de propagation, le modèle le plus générique de T où F est faux. Si le modèle est construit avec succès, alors la réponse est NON (parce que F n'est pas une conséquence logique de T) ; si le modèle construit n'est pas un succès, alors la réponse est OUI (parce qu'il n'existe pas un modèle de T avec F faux, donc F est réellement une conséquence logique de T). Les règles de propagation viennent directement de la sémantique de constructeurs.
D'une manière générale, l'algorithme de tableaux appliqué dans une logique de description essaye de prouver la satisfaisabilité d'une expression de concept D en démontrant l'existence d'une interprétation $\mathcal{I}$ dans laquelle $D^{\mathcal{I}} \neq \emptyset$.
Cette technique de raisonnement permet aujourd'hui de proposer des algorithmes de décision de satisfaisabilité et de subsomption qui sont corrects et complets pour les langages très expressifs de la logique de descriptions.

B.2.3/ LES RAISONNEURS POUR OWL

Actuellement, il existe plusieurs moteurs d'inférence, la plupart conçus pour raisonner sur les logiques de description, mais qui acceptent en entrée des fichiers OWL/RDF(S). Parmi ceux-ci, on peut citer Fact++ [2], HermiT [3], RacerPro [4] et Pellet [5], etc.

	FaCT++	HermiT	RacerPro	Pellet
Expressivité	$\mathcal{SROIQ}(\mathcal{D})$	$\mathcal{SROIQ}(\mathcal{D})$	$\mathcal{SHIQ}$	$\mathcal{SROIQ}(\mathcal{D})$
Regles	-	+ (SWRL)	+ SWRL	+ (SWRL)
Language	C++	JAVA	JAVA	JAVA
Methode	tableau	hypertableau	tableau	tableau
Raisonnement sur la ABox	+	+	+	+
OWL API	+	+	+	+

TABLE B.8 – Moteurs d'inférence

Certains moteurs d'inférence ne peuvent raisonner qu'au niveau terminologique, alors que des moteurs comme Pellet et RacerPro permettent de raisonner aussi sur les instances deconcepts. Nous les présentons ci-dessous :

– *FaCT++ (Fast Clssification of Terminologies)* : est un raisonneur OWL DL($\mathcal{SHOIN}(\mathcal{D})$) supportant un sous ensemble de OWL 2 ($\mathcal{SHOINQ}(\mathcal{D})$). Il a été implémenté en C++ et utilise un algorithme de tableaux optimisé pour de meilleures performances.
– *HermiT* : est un raisonneur codé en JAVA, utilisant les hypertableaux. Il a une expressivité allant jusqu'à ($\mathcal{SHOINQ}(\mathcal{D})$).
– *RacerPro (Renamed Abox and Concept Expression Reasoner)* : est un raisonneur utilisant les hypertableaux calculus optimisé pour la logique de description $\mathcal{SHOIQ}$. Il peut traiter des ontologies OWL-Lite et DL, mais ignore les types définis par l'utilisateur et les concepts énumérées.
– *Pellet* : est le premier raisonneur à supporter entièrement OWL-DL. La nouvelle version, Pellet 2.0 est compatible avec le langage OWL 2 et intègre diverses techniques d'optimisation, y compris pour les nominaux, les requêtes conjonctives, etc. Pellet est codé en JAVA et fonctionne sur la base de tableaux.

Le tableau A.8 résume les propriétés de ces raisonneurs.

2. http ://owl.man.ac.uk/factplusplus/
3. http ://hermit-reasoner.com/
4. http ://franz.com/agraph/racer/
5. http ://clarkparsia.com/pellet/

B.3/ CONCLUSION

Dans cette annexe nous avons traité en premier temps des différents langages de la logique de description, puis nous avons examiné les points les plus importants concernant les problèmes de raisonnement dans la construction des systèmes de représentation de connaissances basés sur la logique de description. Le raisonnement se concentre sur généralement sur la *TBox* avec des algorithmes de tableaux.

C

ÉTUDE COMPARATIVE DES TRIPLESTORES

Un système d'information peut être perçu selon trois axes : les données, les traitements et les communications. Depuis une cinquantaine d'années, l'ensemble de ces axes fait l'objet de recherches intensives pour améliorer les temps de traitements, la gestion de l'espace physique, la recherche et la transmission d'informations. Depuis ces 20 dernières années, certains identifient un 4 ème axe de représentation du système d'information qui est l'axe sémantique. Cet axe, indépendant des contraintes physiques de gestion de l'information tente de résoudre les problèmes d'hétérogénéité sémantique On parle alors de graphe sémantique, d'ontologie et de triplestore.

Le modèle relationnel des bases de données s'est rapidement imposé et occupe une position dominante depuis 40 ans. Seulement, l'apparition du Web a montré limites de ce modèle. Dès le début, Tim Berners-Lee a mis en avant une vision du Web comme une base de données géante, et l'inadéquation du modèle relationnel pour gérer ces données. Il a très tôt proposé un nouveau modèle, le modèle sémantique, basé sur un format qui allait rapidement devenir le RDF.

Un triplestore est une base de données spécialement conçue pour le stockage et la récupération de données RDF. Tout comme une base de données relationnelle, un triple store stocke des données et il les récupère via un langage de requête. Mais contrairement à une base de données relationnelle, un triplestore ne stocke qu'un seul type de données, le triple, sous forme de (sujet, prédicat, objet). Un triplestore n'a donc pas besoin de phase d'initialisation pour enregistrer de nouvelles données, c'est-à-dire il n'a pas besoin de créer des tables comme dans une base de données relationnelle. De plus, un triple store est optimisé pour le stockage d'un grand nombre de triplets et pour la récupération de ces triplets à l'aide du langage de requête SPARQL [1].

L'objectif de cette annexe est, dans la première partie, de positionner la notion de

1. http ://dbpedia.org/page/Triplestore

triplestore par rapport à une logique de stockage des données du système d'information. Puis, dans la seconde partie, nous présenterons l'ensemble des outils relatifs au stockage de données sémantiques de type RDF. Dans la troisième partie, nous proposerons une classification multicritère de ces outils. Des tableaux de synthèse permettront de mettre en évidence les tendances principales, les points communs ou les originalités de ces outils. Cette étude nous permet de justifier l'utilisation du triplestore Stardog pour la réalisation de notre système.

C.1/ ÉVOLUTION DES MODÈLES DE DONNÉES

Dans cette section, nous présenterons un aperçu des différentes évolutions des modèles de données. Les concepts abordés dans cette section sont nécessaires à la compréhension de ceux décrivant les triplestores. Cette section présente quatre modèles qui ont précédé le modèle sémantique : le modèle hiérarchique, le modèle réseau, le modèle relationnel et le modèle objet. Chaque modèle est décrit brièvement, et ses avantages ainsi que ses limites sont mis en évidence. A la fin de cette section nous décrivons le modèle sémantique.

C.1.1/ LE MODÈLE HIÉRARCHIQUE

Le modèle hiérarchique voit le jour au milieu des années 60, en même temps que le concept de base de données. Dans ce modèle, l'information est organisée de manière arborescente et accessible uniquement à partir de la racine de cette arborescence. Les différents enregistrements sont reliés par des pointeurs.
Ce modèle présente des avantages : Sa structure arborescente est adaptée à un grand nombre d'organisations économiques et sociales, il est simple et son implémentation est facile. Par contre ce modèle présente quelques défauts : L'impossibilité de modéliser des liens multiples fait que le chemin d'accès à une information est unique et oblige à créer de la redondance. Cette redondance rend complexe les opérations élémentaires. Par exemples, l'insertion d'une information impose la création d'un segment parent, la mise à jour d'une information doit s'effectuer sur toutes ses occurrences, et la suppression d'un nœud entraîne la disparition des descendants.

C.1.2/ LE MODÈLE RÉSEAU

Le modèle réseau est en mesure de lever de nombreuses difficultés du modèle hiérarchique. Dans ce modèle, comme dans le modèle hiérarchique, les différents

enregistrements sont reliés par des pointeurs. Mais la structure arborescente descendante est remplacée par des liens maillés. Les liens maillés permettent de modéliser les liens multiples, de type n-m, entre les données. On élimine ainsi les redondances et la complexité qu'elles induisent. On multiplie par la même occasion les chemins d'accès à une même information.
En revanche, ce modèle n'apporte pas d'amélioration du point de vue des stratégies d'accès.

C.1.3/ LE MODÈLE RELATIONNEL

Dès le milieu des années 70, des travaux de recherche en mathématique vont permettre la conception d'un nouveau modèle afin remédier aux défauts des deux précédents modèles, concernant les stratégies d'accès. Il s'agit du modèle relationnel. Le modèle relationnel, comme son nom l'indique, est basé sur la notion de relations. Ces relations correspondent à des entités du monde réel. Elles sont très simplement liées entre elles par un système de clé primaire et clé étrangère. Une clé primaire permet d'identifier une ligne, et les autres lignes vont utiliser cette clé primaire, sous forme de clé étrangère, pour faire référence à cette ligne. La création de nombreux standards va libérer les entreprises de la dépendance à un système particulier de base de données. De nombreuses techniques de conception vont améliorer la mise en œuvre de ce modèle. Il s'est révélé être un excellent outil pour gérer des données structurées. Ce modèle s'est imposé comme le modèle principal depuis 40 ans. De nombreux systèmes de gestion de base de données relationnelle sont aujourd'hui matures.

C.1.4/ LE MODÈLE OBJET

Les progrès de la recherche sur le premier axe du système d'information, les données, ont abouti à la généralisation du modèle relationnel. Parallèlement, les recherches sur le deuxième axe, les traitements, aboutissaient dans les années 90 à la programmation objet, suivi par l'apparition de systèmes de gestion de base de données objet (SGBDO).
Dans le modèle objet les informations sont regroupées sous forme d'objets. Un objet représente une entité du monde réel. Il contient un conteneur logique qui englobe des informations sur l'entité ainsi que les traitements qui lui sont propres. Les objets sont créés dynamiquement lors de l'exécution d'un programme écrit en langage de programmation orienté objet. Chaque objet est créé à partir d'une classe. Une classe peut être dérivée d'une autre classe, dans ce cas elle héritera des caractéristiques de la classe source. Chaque classe contient des attributs

qui correspondent aux informations sur l'objet. Les classes comportent aussi des méthodes, des traitements propres à cet objet, qui manipulent les valeurs des attributs.
Les techniques de conception des systèmes d'information développées pour le modèle relationnel ont été adaptées pour être compatibles avec la programmation objet. La persistance dans la programmation objet est le plus souvent réalisée par un mapping objet-relationnel vers une base de données relationnelle.
Le modèle objet a eu un impact faible sur le monde de la persistance des données et n'a pas remis en cause la dominance du modèle relationnel.

C.1.5/ LE MODÈLE SÉMANTIQUE

Nous avons vu que le modèle relationnel s'était rapidement imposé, et ce pour longtemps. Son succès tient à sa parfaite capacité à gérer des données de manière structurée, et à l'existence de nombreux standards et techniques de conception. Seulement le développement du Web a entrainé l'explosion du volume des données accessibles. Le Web peut être vu comme une gigantesque source de données non structurées. Le modèle relationnel s'avère inadapté pour la gestion de ces données.
L'exploitation des ressources sur le Web passe par la conception de nouvelles technologies. L'objectif de ces technologies est de rendre l'information contenue dans les documents du Web utilisable par des programmes et agents logiciels, grâce à un système de métadonnées formelles. L'ensemble de ces technologies et les données qu'elles structurent s'appelle le Web sémantique.
Les créateurs du Web ont anticipé dès le début la nécessité de concevoir un nouveau modèle pour gérer les informations non structurées du Web. Ils ont (1) créé un nouveau modèle, le modèle sémantique (1) l'ont basé sur un langage, RDF (2) l'ont doté d'un langage d'interrogation, le SPARQL et (3) l'ont équipé d'une fonctionnalité originale, l'inférence.
(1) RDF est un modèle de graphe destiné à décrire de façon formelle les données non structurées du Web. Un document respectant la structure RDF se compose d'un ensemble de déclarations sous forme de triplets : (Sujet, prédicat, Objet) (figure B.1).

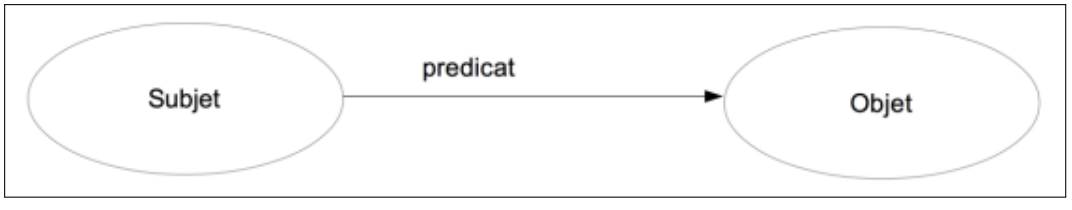

FIGURE C.1 – Triplet RDF

Le sujet représente la ressource ; Le prédicat représente une propriété applicable à cette ressource ; L'objet représente une donnée ou une autre ressource. Sujets, prédicats et objets sont souvent des URI, mais l'objet peut aussi être un littéral. L'exemple de la figure ci-dessous illustre la page Web du W3C sur la syntaxe RDF [2], en utilisant les triplets :

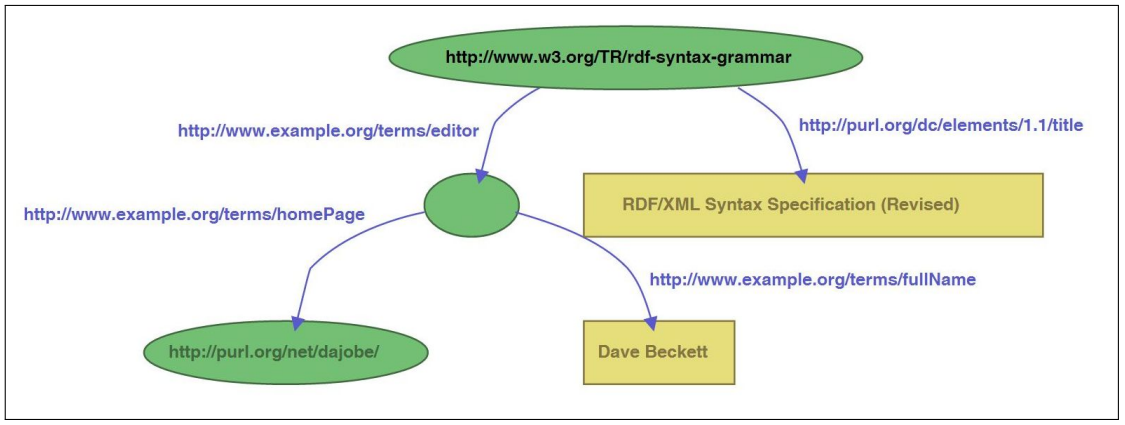

FIGURE C.2 – Exemple de graphe RDF

La principale syntaxe de RDF est le RDF/XML. D'autres sérialisations de RDF sont apparues ensuite, cherchant à rendre la lecture plus compréhensible. Voilà la représentation du graphe précédent en RDF/XML :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
              xmlns:dc="http://purl.org/dc/elements/1.1/"
              xmlns:ex="http://example.org/stuff/1.0/">
       <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
              dc:title="RDF/XML\ Syntax Specification (Revised)">
              <ex:editor>
                     <rdf:Description ex:fullName="Dave Beckett">
                            <ex:homePage rdf:resource=
                            "http://purl.org/net/dajobe/" />
                     </rdf:Description>
              </ex:editor>
       </rdf:Description>
</rdf:RDF>
```

(2) Pareil qu'un modèle relationnel, le modèle sémantique a aussi son langage d'interrogation : le SPARQL. Le langage d'interrogation SPARQL pour RDF est conçu pour satisfaire aux cas d'utilisation et aux exigences décrites par le groupe

2. http ://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

de travail RDF Data Access [3]. Ci-dessous un des exemples de la page du W3C dédiée au SPARQL :

Données (en syntaxe Turtle) :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy" .
_:b foaf:mbox <mailto:peter@example.org> .
_:c foaf:mbox <mailto:carol@example.org> .
```

Requête SPARQL :

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
  { ?x foaf:name ?name .
    ?x foaf:mbox ?mbox }
```

Résultat :

name	**mbox**
”Johnny Lee Outlaw”	<mailto :jlow@example.com>
”Peter Goodguy”	<mailto :peter@example.org>

On retrouve la structure des déclarations sous forme de triplets dans la clause WHERE. Dans cette requête, on recherche le nom et l'adresse mail de tous les sujets ayant un nom et une adresse mail. Le W3C a publié une recommandation qui propose un standard de protocole d'accès SPARQL. Tous les triplestores étudiés dans la suite de cette annexe proposent un accès SPARQL conforme à ce protocole.

(3) Nous avons vu que le modèle sémantique est basé sur le langage RDF et qu'il possède un langage d'interrogation, le SPARQL. Ce modèle a aussi été doté d'une fonctionnalité originale, l'inférence. Les moteurs d'inférence, ou encore raisonneurs (voir annexe A) sont ici utilisés comme processus de déduction de nouveaux triplets. À partir de déclarations existantes et de règles, un moteur d'inférence va générer de nouveaux triplets. L'inférence n'est pas une fonctionnalité obligatoire des triplestores. Seule la moitié des triplestores étudiés dans cette annexe possèdent un moteur d'inférence.

3. http ://www.w3.org/TR/rdf-dawg-uc/

C.2/ LES SOLUTIONS TRIPLESTORES

Un triplestore peut être considéré comme une base de données conçue pour le stockage et la récupération de triplets. La notion de triplestore peut être étendue plus généralement à tout outil qui permet de stocker et de récupérer des triplets RDF. Les différents outils triplestore présentés dans cette section ont tous la capacité de manipuler et stocker des graphes RDF. Dans cette section nous allons étudier les triplestores les plus connus selon les syntaxes RDF supportées, les langages d'interrogation supportés, l'inférence, les langages supportés pour l'écriture de clients, les langages des sources et les licences.

C.2.1/ SYNTAXES RDF SUPPORTÉS

Nous avons vu dans la première partie que le modèle sémantique est basé sur RDF. Les syntaxes utilisées pour écrire les déclarations RDF sont nombreuses. Chaque triplestore en supporte un certain nombre. Les différentes syntaxes supportées par les triplestores retenus sont : RDF/XML, RDF/XML-ABBREV, RDFa, RDF/JSON, N3, N3-PP, N3-Triple, N-Triple, N-Quads, Turtle, TriX, TriG, et GRDDL.
Après une brève description de ces syntaxes, nous présenterons, dans cette section, un tableau qui permet d'identifier les syntaxes supportées par chaque triplestore.

- RDF/XML est la syntaxe de référence du W3C pour le RDF.
- RDF/XML-Abbrev est proposé par la librairie Jena : parmi les multiples façons d'exprimer en RDF/XML un même ensemble de déclarations RDF, ce format est le plus compact et le plus léger en octets.
- RDFa permet d'intégrer dans une page Web des données structurées. Il n'est pas interprété par les navigateurs.
- RDF/JSON permet de sérialiser les triplets sous la syntaxe JSON.
- N3 (ou Notation-3) a été conçu pour être plus accessible à un humain.
- Les syntaxes N3-XX, N-Triple et Turtle sont des sous-ensembles de N3.
- N-Quads est une extension de N-Triple qui lie un contexte à chaque déclaration.
- TriX et TriG sont des extensions de Turtle.
- GRDDL n'est pas exactement une syntaxe, c'est un mécanisme qui permet d'extraire la partie RDF d'un document XML et de l'exporter dans une syntaxe particulière.

4. http ://www.4store.org/
5. http ://4store.org/trac/wiki/5store
6. http ://www.franz.com/agraph/allegrograph/

Triplestore	RDF/XML	RDF/XML-ABBREV	RDFa	RDF/JSON	N3	N3-PP	N3-Plain	N3-Triple	N-Triple	N-Quads	Turtle	TriX	TriG	GRDDL
4store [4]	√		√	√					√	√	√		√	√
5store [5]	√		√	√					√	√	√		√	√
AllegroGraph [6]	√				√				√		√			
ARC2 [7]	√			√					√		√			
Bigdata [8]	√		√		√				√	√	√	√	√	
Jena [9]	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Mulgara [10]	√	√	√	√	√	√	√	√	√	√	√	√	√	√
OpenAnzo [11]	√										√			
OntoBroker [12]	√				√				√		√			
Oracle Semantic [13]	√	√	√	√	√	√	√	√	√	√	√	√	√	√
OWLIM [14]	√		√		√				√		√	√	√	
Parliament [15]	√	√	√	√	√	√	√	√	√	√	√	√	√	√
RDF : :Trine [16]	√		√	√					√	√	√			
Redland [17]	√								√		√			
Semantics Platform [18]	√								√		√			
SemWeb-DotNet [19]	√				√			√			√			
Sesame [20]	√		√		√				√		√	√	√	
Soprano [21]	√		√		√			√	√	√	√	√	√	
Virtuoso [22]	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Stardog [23]	√								√	√	√	√	√	

TABLE C.1 – Formats RDF supportés par chaque les triplestores

7. https ://github.com/semsol/arc2/wiki
8. http ://www.systap.com/bigdata.htm
9. http ://jena.sourceforge.net/index.html
10. http ://www.mulgara.org/
11. http ://www.openanzo.org/
12. http ://www.ontoprise.de/en/products/ontobroker/
13. http ://www.oracle.com/technetwork/database/options/semantic-tech/index.html
14. http ://www.ontotext.com/owlim
15. http ://parliament.semwebcentral.org/
16. http ://search.cpan.org/dist/RDF-Trine/
17. http ://librdf.org/
18. http ://www.intellidimension.com/
19. http ://razor.occams.info/code/semweb/
20. http ://www.openrdf.org/
21. http ://soprano.sourceforge.net/
22. http ://virtuoso.openlinksw.com/

Tous les triplestores supportent le format de référence RDF/XML, et le format Turtle. La plupart supportent aussi le format N-Triples.

C.2.2/ LANGAGES D'INTERROGATION

Nous avons vu dans la première partie que le modèle sémantique est doté d'un langage d'interrogation : le SPARQL. Certains triplestores utilisent d'autres langages d'interrogation. Ces langages sont : SeRQL, RDQL, iTQL et Prolog. Nous présentons dans cette section un tableau qui nous permet d'identifier les langages supportés par les différents triplestores.

Triplestore	SPARQL	SeRQL	RDQL	iTQL	Prolog
4store	√				
5store	√				
AllegroGraph	√	√			√
ARC2	√				
Bigdata	√	√			
Jena	√				
Mulgara	√			√	
OpenAnzo	√				
OntoBroker	√				
Oracle Semantic	√	√			
OWLIM	√	√			
Parliament	√	√			
RDF : :Trine	√		√		
Redland	√		√		
Semantics Platform	√				
SemWeb-DotNet	√				
Sesame	√	√			
Soprano	√	√	√		
Virtuoso	√	√	√		
Stardog	√				

TABLE C.2 – Langages de requêtes supportés

On voit que tous les triplestores supportent le langage SPARQL, recommandé par le W3C, et qui s'est imposé comme le langage d'interrogation de référence des triplestores.

23. http ://stardog.com/

C.2.3/ INFÉRENCE

Un triplestore permet d'utiliser des moteurs d'inférences indépendants. Les ensembles de références sont RDFS, OWL Lite, OWL DL, EL, QL, RL, SL, SKOS, DAML, NRL. Il permet aussi d'utiliser des moteurs d'inférences externes qui sont Pellet et TrOWL. Nous présentons dans cette section un tableau qui nous permettra d'identifier les triplestores dont le moteur d'inférence supporte les règles de type SWRL.

Triplestore	**Moteurs internes**										**externes**		**Règles utilisateur**
	RDFS	**OWL Lite**	**OWL DL**	**EL**	**QL**	**RL**	**SL**	**SKOS**	**DAML**	**NRL**	**Pellet**	**TrOWL**	
4store													
5store													
AllegroGraph	√	√											
ARC2	√	√											
Bigdata	√	√											
Jena	√	√	√						√		√	√	√
Mulgara	√	√	√										√
OpenAnzo													
OntoBroker	√	√	√										
Oracle Semantic	√	√	√					√					
OWLIM	√	√	√										
Parliament	√	√											
RDF : :Trine													
Redland													
Semantics Platform													
SemWeb-DotNet	√												
Sesame	√												
Soprano	√									√			
Virtuoso	√	√											√
Stardog	√			√	√	√	√						√

TABLE C.3 – Moteurs d'inférence

Presque tous les triples stores supportent RDFS. Mais seul Stardog supporte OWL 2 (EL, QL, RL) et le des règles en SWRL (SL).
Dans le cadre de notre travail, Stardog est le plus adapté pour inférer sur notre ontologie de domaine OWL 2 et nos règles métiers.

C.2.4/ LANGAGES SUPPORTÉS POUR LES CLIENTS

Nous présentons dans cette section un tableau décrivant pour chaque triplestore les langages dans lesquels les clients peuvent être écrits.

Triplestore	C/C++	C#	Clojure	Django	Java	Jena	Sesame	Javascript	Common Lisp	Perl	PHP	Python	Ruby	Scala	Tcl
4store				√	√						√	√	√		
5store				√	√						√	√	√		
AllegroGraph		√	√		√	√	√		√	√	√	√	√	√	
ARC2											√				
Bigdata					√		√				√				
Jena					√	√									
Mulgara					√	√	√			√	√				
OpenAnzo								√							
OntoBroker					√										
Oracle Semantic					√	√	√				√				
OWLIM					√		√				√				
Parliament	√				√	√	√				√				
RDF : :Trine										√					
Redland		√			√					√	√	√	√		√
Semantics Platform	√														
SemWeb-DotNet															
Sesame					√		√				√				
Soprano	√				√		√				√				
Virtuoso		√			√	√	√			√	√	√	√		√
Stardog					√	√	√								

TABLE C.4 – Langages de requêtes supportés

C.2.5/ LICENCES

Les triplestores sont protégés par différentes licences, commerciales ou open source. Nous présentons dans cette section un tableau permettant d'identifier les triplestores protégés par une licence et ceux protégés par une licence open source.

Triplestore	Commerciale	Open-source	Description
4store		√	GNU General Public License
5store	√		
AllegroGraph	√		
ARC2		√	GNU General Public License, W3C Software License
Bigdata		√	GNU General Public License, évaluation/research license
Jena		√	http ://jena.sourceforge.net/license.html
Mulgara		√	Open Software v3.0, Apache 2.0
OpenAnzo		√	Eclipse Public License 1.0.
OntoBroker	√		
Oracle Semantic	√		
OWLIM	√	√	GNU Lesser General Public License
Parliament		√	Berkeley Sofware Distribution
RDF : :Trine		√	Mozilla Public License, GNU General Public License
Redland		√	GNU Lesser General Public License GPL Apache
Semantics Platform	√		
SemWeb-DotNet	√		
Sesame		√	Berkeley Sofware Distribution
Soprano		√	GNU General Public Licence
Virtuoso	√		
Stardog	√	√	Clark & Parsia LLC

TABLE C.5 – Licence des triplestore

C.3/ CONCLUSION

Le Web constitue une gigantesque base de données non structurée. Le modèle relationnel s'est imposé depuis une quarantaine d'années pour sa parfaite capacité à gérer des données structurées, mais il s'avère inadapté à la gestion de cette source de données non structurées. Un nouveau modèle a été conçu pour gérer ces nouvelles ressources. Il se base sur le langage RDF, le langage d'interrogation SPARQL et la fonctionnalité d'inférence. Les triplestores sont des outils permettant de stocker et de récupérer des données RDF. A travers les différents tableaux d'analyse des triplestores les plus connus, nous avons identifié Stardog comme étant le triplestore le plus adapté à notre projet.

RAPPORTS TECHNIQUES

≪M-learning 3.0≫, Fayrouz Soualah Alila, novembre 2011.

≪La recommandation en apprentissage mobile≫, Fayrouz Soualah Alila, décembre 2011.

≪La contextualisation pour l'apprentissage mobile≫, Fayrouz Soualah Alila, janvier 2012.

≪État de l'art de l'apprentissage mobile≫, Fayrouz Soualah Alila, février 2012.

≪IADIS Mobile Learning 2012≫, Fayrouz Soualah Alila, mars 2012.

≪Projet MEMORAe≫, Fayrouz Soualah Alila, mars 2012.

≪Langages ontologiques et normes e-learning≫, Fayrouz Soualah Alila, mars 2012.

≪Une nouvelle approche d'application des technologies du Web sémantique pour un apprentissage centré sur l'étudiant≫, Fayrouz Soualah Alila, avril 2012.

≪SCORM 2004≫, Fayrouz Soualah Alila, mai 2012.

≪Aperçu sur le catalogue CrossKnowledge≫, Fayrouz Soualah Alila, mai 2012.

≪Rapport de recherche ANR, num 1≫, Fayrouz Soualah Alila, juin 2013.

≪Résumé 1 ère année de thèse≫, Fayrouz Soualah Alila, septembre 2012.

≪Formalisation en logique de description≫, Fayrouz Soualah Alila, décembre 2012.

≪Une ontologie de domaine du m-learning≫, Fayrouz Soualah Alila, janvier 2013.

≪Sesame, Triple Store et règles≫, Fayrouz Soualah Alila, juin 2013.

≪Rapport de recherche ANR, num 2≫, Fayrouz Soualah Alila, juin 2013.

≪Résumé 2 ème année de thèse≫, Fayrouz Soualah Alila, septembre 2013.

≪Modélisation mathématiques du problème d'optimisation combinatoire en m-learning≫, Fayrouz Soualah Alila, janvier 2014.

≪Algorithmes pour la résolution du problème d'optimisation combinatoire en m-learning, version 1≫, Fayrouz Soualah Alila, février 2014.

≪Algorithmes pour la résolution du problème d'optimisation combinatoire en m-learning, version 2≫, Fayrouz Soualah Alila, février 2014.

≪Règles métier pour le m-learning≫, Fayrouz Soualah Alila, février 2014.

PUBLICATIONS

Revues

≪Semantic and Contextual Approach for the Recommendation of Learning Modules in Mobility≫, Fayrouz Soualah Alila, Christophe Nicolle, Florence Mendes, IEEE Learning Technology Newsletter, vol.4, issue.2, pp.33-35, 2012.

≪A Context-Based Adaptation in Mobile Learning≫, Fayrouz Soualah Alila, Christophe Nicolle, Florence Mendes, IEEE Computer Society Technical Committee on Learning Technology (TCLT), vol.15, issue4, pp.14-18, 2013.

Chapitre de livre

≪Towards a Methodology for Semantic and Context-Aware Mobile Learning≫, Fayrouz Soualah Alila, Christophe Nicolle, Florence Mendes, The Encyclopedia of Information Science and Technology (IS&T2013), third edition, 2013.

Conférences internationales

≪Recommender System for Combination of Learning Elements in Mobile Environment≫, Fayrouz Soualah Alila, Christophe Nicolle, Florence Mendes, IADIS International Conference on Mobile Learning (IADIS 2012), pp.291-295, Berlin, Germany, 2012.

≪Context Aware Adaptive System for M-Learning Personalization≫, Fayrouz Soualah Alila, Christophe Nicolle, Florence Mendes, 13th International Conference on Informatics in Economy Education (IE2014), Bucharest, Romania, 2014.

Conférences nationales

≪Recommandation de Parcours de Formation dans un Contexte Mobile≫, Fayrouz Soualah Alila, Florence Mendes, Christophe Nicolle, 14 ème Conférence ROADEF de la Société Française de Recherche Opérationnelle et Aide à la Décision, Troyes, France, 2013.

≪Une Approche Web Sémantique et Combinatoire pour un Système de Recommandation Sensible au Contexte Appliqué à l'Apprentissage Mobile≫, Fayrouz Soualah Alila, Florence Mendes, Christophe Nicolle, 14 èmes Journées Francophones Extraction et Gestion des Connaissances, Rennes, France, 2014.

BIBLIOGRAPHIE

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines : A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, New York, NY, USA.

Abowd, G., A.K., Brown, P., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In Gellersen, H.-W., editor, *Handheld and Ubiquitous Computing*, volume 1707, pages 304–307. Springer Berlin Heidelberg.

Achiri, S. (2012). Une architecture agents pour l'adaptation au contexte des systèmes d'information ubiquitaires. Technical report.

Adomavicius, G., Sankaranarayanan, R., S. Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1) :103–145.

Ahn, J., Brusilovsky, P., Grady, J., He, D., and Syn, S. (2007). Open user profiles for adaptive news systems : Help or harm ? In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 11–20, New York, NY, USA. ACM.

Allen, J. F. and Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4 :531–579.

Amalric, P. (2007). Aperçu rapide de scorm 1.2 pour les développeurs de contenus, livre blanc. Technical report.

Andronico, A., Carbonaro, A., Casadei, G., Colazzo, L., Molinari, A., and Ronchetti, M. (2003). Integrating a multi-agent recommendation system into a mobile learning management system.

Autin, B. (2006). Les métaheuristiques en optimisation combinatoire. Technical report, Conservatoire National des Arts et Métiers Paris.

Azouaou, F. and Desmoulins, C. (2006). Using and modeling context with ontology in e-learning : the case of teacher's personal annotation. In *International Workshop on Applications of Semantic Web Technologies for E-Learning*, Dublin, Irlande.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., editors (2003). *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press.

Bader, R., Neufeld, E., Woerndl, W., and Prinz, V. (2011). Context-aware poi recommendations in an automotive scenario using multi-criteria decision making methods. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, CaRR '11, pages 23–30, New York, NY, USA. ACM.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Balabanovic, M. and Shoham, Y. (1997). Fab : Content-based, collaborative recommendation. *Commun. ACM*, 40(3) :66–72.

Balasubramanian, V. (1994). State of the art review on hypermedia issues and applications.

Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Computing*, 16(5) :507–526.

Baluja, S. (1994). Population-based incremental learning : A method for integrating genetic search based function optimization and competitive learning. Technical report.

Basile, P., Degemmis, M., Gentile, A., Iaquinta, L., and Lops, P. (2007). The jump project : Domain ontologies and linguistic knowledge @ work. In *SWAP*, volume 314. CEUR-WS.org.

Beale, R. and Lonsdale, P. (2004). Mobile context aware systems : The intelligence to support tasks and effectively utilise resources. volume 3160.

Behlouli, N. B. (2006). *Ajout de Mécanismes de Réactivité au Contexte dans les Intergiciels pour Composants dans le Cadre d'Utilisateurs Nomades*. PhD thesis, Université d'Évry Val d'Essonne.

Berners-Lee, T. (1998). Semantic web road map.

Berners-Lee, T. (2000). Semantic web - xml2000.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5) :34–43.

Berti-Equille, L. (2002). Annotation et recommandation collaboratives de documents selon leur qualité. *Ingénierie des Systèmes d'Information*, 7(1-2) :125–155.

Blanchard, E. (2007). *Motivation et culture en e-Learning*. PhD thesis, Faculté des Arts et des Sciences, Département d'Informatique et de Recherche Opérationnelle.

Bomsdorf, B. (2005). Adaptation of learning spaces : Supporting ubiquitous learning in higher distance education. In Davies, N., Kirste, T., and Schumann, H., editors, *Mobile Computing and Ambient Intelligence : The Challenge of Multimedia*, number 05181 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence : From Natural to Artificial Systems*. Oxford University Press.

Bourda, Y. (2002). Des objets pédagogiques aux dossiers pédagogiques (via l'indexation). In *Document numérique*, pages 115–128.

Bra, P. D., Houben, G., and Wu, H. (1999). Aham : A dexter-based reference model for adaptive hypermedia. In *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia : Returning to Our Diverse Roots : Returning to Our Diverse Roots*, pages 147–156, New York, NY, USA. ACM.

Brown, P. (1996). The stick-e document : a framework for creating context-aware applications. pages 182–196.

Brown, P., Bovey, J., and Chen, X. (1997). Context-aware applications : from the laboratory to the marketplace. *IEEE Personal Communications*, 4(5) :182–196.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3) :87–129.

Brusilovsky, P., Kobsa, A., and Nejdl, W., editors (2007). *The Adaptive Web : Methods and Strategies of Web Personalization*. Springer-Verlag, Berlin, Heidelberg.

Brézillon, P. (2002). Expliciter le contexte dans les objets communicants. In Gérard, P., Kintzig, C., Privat, G., and Favennec, P., editors, *Les Objets Communicants*, pages 295–303. Hermès.

Burke, R. (2002). Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370.

Bush, V. (1945). As we may think. *Atlantic Monthly*, 176 :101–108.

Camara, F., Demumieux, R., Calvary, G., and Mandran, N. (2010). Cocoon, un système de recommandation sensible au contexte : Analyse de la valeur par une etude qualitative. In Calvary, G. and Wolff, M., editors, *Actes de la conférence Ergo 'IA 2010*, pages 211–218, Biarritz, France. ACM.

Cauchy, A.-L. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Compte Rendu des Séances de L'Académie des Sciences XXV*, Série A :536–538.

Chaari, T. (2007). *Adaptation d'applications pervasives dans des environnements multi-contextes*. PhD thesis, INSA de Lyon.

Chaari, T., Laforest, F., and Flory, A. (2005). Adaptation des applications au contexte en utilisant les services web : Le projet secas. In *La Deuxième Journées Francophones Mobilité et Ubiquit*, pages 111–118, Grenoble, France.

Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. Technical report, Hanover, NH, USA.

Chen, H., Finin, T., and Joshi, A. (2004a). Semantic web in the context broker architecture. In *the Second IEEE International Conference on Pervasive Computing and Communications PerCom'04*, pages 277–286. IEEE Computer Society.

Chen, H., Perich, F., Finin, T., and Joshi, A. (2004b). Soupa : Standard ontology for ubiquitous and pervasive applications. In *Mobile and Ubiquitous Systems : Networking and Services*.

Chen, L., Khan, S., Li, K., and Manning, E. (1999). Building an adaptive multimedia system using the utility model. In *IPPS/SPDP Workshops*, volume 1586 of *Lecture Notes in Computer Science*, pages 289–298. Springer.

Chomienne, E., Badaoui, M., Gonzalves, N., Zene, Y., and Gentil, C. (2008). Présentation des freins et des facteurs de changement pour l'e-learning à l'heure de l'université 2.0, prospective des tendances et des evolutions des méthodes d'apprentissage.

Chu, W. and Park, S. (2009). Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 691–700, New York, NY, USA. ACM.

Colorni, A., Dorigo, M., and Maniezzo, V. (1992). An investigation of some properties of an ant algorithm. In *Parallel Problem Solving from Nature Conference*, pages 509–520.

Cristea, A. and Andmethods, H. (2003). Laos : Layered www ahs authoring model and their corresponding algebraic operators. In *WWW 2003 Conference*. ACM.

Daruwala, Y. (2008). *A Constructive Memory Architecture for Context Awareness*. University of Sydney. Faculty of Architecture, Design and Planning.

de Groot, B. and van Welie, M. (2002). Leveraging the context of use in mobile service design. In *Mobile HCI*, volume 2411, pages 334–338.

Degemmis, M., Lops, P., and Semeraro, G. (2007). A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3) :217–255.

Delestre, N., Pecuchet, J., and Barry-Greboval, C. (1998). L'architecture d'un hypermédia adaptatif dynamique pour l'enseignement. In *Nouvelles technologies de l'information et de la communication dans les formations d'ingénieurs et dans l'industrie (NTICF)*, Rouen, France.

Derntl, M. and Hummel, K. (2005). Modeling context-aware e-learning scenarios. In *The 3rd Int'l Conf. on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops)*, pages 337–342. IEEE.

Dey, A. (2000). *Providing Architectural Support for Building Context-aware Applications*. PhD thesis, Atlanta, GA, USA.

Dey, A., Abowd, G., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2) :97–166.

Dijkstra, E. W. (1971). A short introduction to the art of programming. *International Review of Research in Open and Distance Learning*.

Djallel, B. (2014). Recommandation mobile, sensible au contexte de contenus evolutifs : Contextuel-e-greedy. *CoRR*.

Dobson, S. (2005). Leveraging the subtleties of location. In *The 2005 Joint Conference on Smart Objects and Ambient Intelligence : Innovative Context-aware Services : Usages and Technologies*, pages 189–193, New York, NY, USA. ACM.

Dogbe-Semanou, D., Durand, A., Leproust, M., and Vanderstichel, H. (2008). Etude comparative de plates-formes de formation à distance.

Dorigo, M. and Gambardella, L. (1997). Ant colony system : A cooperative learning approach to the traveling salesman problem. *Trans. Evol. Comp*, 1(1) :53–66.

Dréo, J. and Siarry, P. (2004). Continuous interacting ant colony algorithm based on dense heterarchy. *Future Generation Computer Systems*, 20(5) :841–856.

Droui, M., Hajjami, A. E., and Ahaji, K. (2014). Apprentissage mobile ou m-learning : opportunités et défis.

Endrizzi, L. (2011). Les apprentissages nomades en europe : éléments pour une cartographie de la recherche.

Engelbart, D. (1984). Authorship provisions in augment. In *COMPCON*, pages 465–472. IEEE Computer Society.

F. Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5) :533–549.

Fahy, P. and Clarke, S. (2004). Cass – a middleware for mobile context-aware applications. In *Workshop on Context Awareness, MobiSys*, pages 304–308, Boston, Massachusetts, USA.

Felder, R. and Carolina, N. (1988). Learning and teaching styles in engineering education. *Engineering Education*.

Fiehl, S., Diaz, M., and Solom, A. (2010). 1er baromètre du e-learning en europe. Technical report.

Fleming, N. and C.Mills (1992). Not another inventory, rather a catalyst for reflection. *To Improve the Academy*, 11 :137–155.

Fogel, L. (1999). *Intelligence Through Simulated Evolution : Forty Years of Evolutionary Programming*. John Wiley & Sons, New York, NY, USA.

Fournier-Viger, P. (2005). *Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents*. PhD thesis, Faculté des Sciences, Université de Sherbrooke, Québec, Canada.

Gagnon, M. (2009). Logique descriptive et owl. Technical report, Ecole Polytechnique de Montréal.

Garey, M. and Johnson, D. (1990). *Computers and Intractability ; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Gauch, S., Chaffee, J., and Pretschner, A. (2002). Ontology-based user profiles for search and browsing.

Georgiev, T., Georgieva, E., and Smrikarov, A. (2004). M-learning, a new stage of e-learning. *International Conference on Computer Systems and Technologies, CompSysTech*, pages 28–5.

Ghayam, Y. E. (2011). *La Sensibilité au Contexte dans un Environnement Mobile*. PhD thesis, Université Mohammed V Souissi-RABAT, École Nationale Supérieure d'Informatique et d'Analyse des Systèmes.

Ghebghoub, O., Abel, M., and Moulin, C. (2009). Lomonto : Une ontologie pour l'indexation d'objets pédagogiques. In *AFIA platform workshop : Constructions d'ontologies : Vers un guide de bonnes pratiques*, Hammamet, Tunisia.

Goldberg, D., Nichols, D., Oki, B., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12) :61–70.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001). Eigentaste : A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2) :133–151.

Grasha, A. and Riechmann, S. (1975). The grasha-riechmann student learning style scales. *Handbook for Faculty Development*.

Gräbener, T., Berro, A., and Duthen, Y. (2010). Time dependent multiobjective best path for multimodal urban routing. *Electronic Notes in Discrete Mathematics*, 36 :487–494.

Gruber, T. (1993). Toward principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, in press. Substantial revision of paper presented at the International Workshop on Formal Ontology*. Kluwer Academic Publishers.

Gu, T., Pung, H., and Zhang, D. (2004). A middleware for building context-aware mobile services. In *IEEE Vehicular Technology Conference (VTC)*.

Halasz, F. and Schwartz, M. (1994). The dexter hypertext reference model. *Commun ACM*, 37(2) :30–39.

Hansen, P., Mladenovic, N., and Moreno-Pérez, J. (2010). Variable neighbourhood search : methods and applications. *Annals of Operational Research*, pages 367–407.

Harries, K. and Smith, P. (1998). Code growth, explicitly defined introns and alternative selection schemes. *Evolutionary Computation*, 6 :339–360.

Haupt, R. and Haupt, S. (2004). *Practical Genetic Algorithms*. John Wiley & Sons, New York, NY, USA.

Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., and Kröner, A. (2007). The user model and context ontology gumo revisited for future web 2.0 extensions.

Henricksen, K. and Indulska, J. (2004). Modelling and using imperfect context information. In *The Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 33–27, Washington, DC, USA. IEEE Computer Society.

Herlocker, J., Konstan, J., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pages 241–250, New York, NY, USA. ACM.

Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., and Retschitzegger, W. (2003). Context-awareness on mobile devices - the hydrogen approach. In *The 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, volume 9, pages 292–302, Washington, DC, USA. IEEE Computer Society.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Hoogeveen, M. (1995). Towards a new multimedia paradigm : is multimedia assisted instruction really effective ? In *ED-MEDIA 95, World Conference on Educational Multimedia and Hypermedia*, pages 348–353, Graz, Austria.

IEEE-LTSC (2002). 1484.12.1-2002 ieee standard for learning object metadata. Technical report.

Jovanovic, J., Gasevic, D., Knight, C., and Richards, G. (2007). Ontologies for effective use of context in e-learning settings. *Educational Technology & Society*, 10(3) :47–59.

Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M., and Tran, L. (2005). Composite capability/preference profiles (cc/pp) : Structure and vocabularies. Technical report.

Koch, N. and Wirsing, M. (2002). The munich reference model for adaptive hypermedia applications. In *2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 213–222.

Kolb, D. (1984). *Experiential Learning : Experience as the Source of Learning and Development*. Prentis-Hall, New Jersey.

Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., and Malm, E. (2003). Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3) :42–51.

Kouninef, B., Djelti, M., and Rerbal, S. (2007). Conception et réalisation d'une plateforme e-learning avec migration au m-learning.

Lainé-Cruzel, S. (1999). Profildoc : Filtrer une information exploitable. *Bulletin des Bibliothèques de France*, (5) :60–65.

Leclercq, D. (1998). *Pour une pédagogie universitaire de qualité*. Mardaga.

Li, L., Chu, W., Langford, J., and Schapire, R. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670.

Lieberman, H. (1995). Letizia : An agent that assists web browsing. In *The 14th International Joint Conference on Artificial Intelligence*, volume 1, pages 924–929, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Liénard, F. (2010). Du e-learning au m-learning : quel(s) intérêt(s) pour les apprentissages ?

Liu, C. (2011). Providing recommendations for mobile learning : Opportunities and challenges from context.

Livingstone, D. (2001). Adults' informal learning : Definitions, finds, gaps, and future research : New approaches for lifelong learning (nall). Centre for the Study of Education and Work, OISE/UT.

Magnini, B. and Strapparava, C. (2000). Experiments in word domain disambiguation for parallel texts. In *Proceedings of the ACL-2000 Workshop on Word Senses and Multi-linguality*, volume 8, pages 27–33, Stroudsburg, PA, USA. Association for Computational Linguistics.

Makssoud, K. A. (2008). *Système d'Accès Personnalisé à l'Information : Application au Domaine Médical*. PhD thesis, Institut National des Sciences Appliquées de Lyon.

McCarthy, J. and Buvac, S. (1994). Formalizing context (expanded notes). Technical report, Stanford, CA, USA.

McSherry, F. and Mironov, I. (2009). Differentially private recommender systems : Building privacy into the net. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 627–636, New York, NY, USA. ACM.

Mercier, F., David, B., Chalon, R., and Berthet, J. (2004). Amphithéâtres interactifs dans l'enseignement supérieur. In *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie*, pages 400–406, Compiègne, France. Université de Technologie de Compiègne.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6) :1087–1092.

Mühlenbein, H. and Paass, G. (1996). From recombination of genes to the estimation of distributions i. binary parameters. In *PPSN 1996*, pages 178–187. Springer-Verlag.

Mladenovic, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24(11) :1097–1100.

Moran, T. and Dourish, P. (2001). Introduction to this special issue on context-aware computing. *Human-Computer Interaction*, 16(2) :87–95.

Mostéfaoui, G., Pasquier-Rocha, J., and Brézillon, P. (2004). Context-aware computing : A guide for the pervasive computing community. In *ICPS*, pages 39–48. IEEE Computer Society.

Murray, T. (1999). Authoring intelligent tutoring systems : An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10(1) :98–129.

Naismith, L., Lonsdale, P., Vavoula, G., and Sharples, M. (2005). Literature review in mobile technologies and learning. Technical report, A Report for NESTA Futurelab.

Nelson, T. (1965). Complex information processing : A file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th National Conference*, pages 84–100, New York, NY, USA. ACM.

Nicholson, T. (1971). *Optimization In Industry : Industrial Applications*.

NISO (2004). Understanding metadata. Technical report, NISO Press.

O.Catteau (2008). Le cycle de vie de l'objet pédagogique et de ses métadonnées.

O'Malley, C., Vavoula, G., Glew, J., Taylor, J., and Sharples, M. (2005). Guidelines for learning/teaching/tutoring in a mobile environment.

Pan, F. (2007). *Representing complex temporal phenomena for the semantic web and natural language*. PhD thesis, University of Southern California.

Pascoe, J. (1998). Adding generic contextual capabilities to wearable computers. In *ISWC*, pages 92–99.

Pazzani, M. and Billsus, D. (2007). Content-based recommendation systems. In *The Adaptive Web : Lecture Notes in Computer Science, chapitre Content-Based Recommendation Systems*, volume 4321, pages 325–341. Springer-Verlag.

Petrelli, D., Not, E., Strapparava, C., Stock, O., and Zancanaro, M. (2000). Modeling context is like taking pictures. In *Conference on Human Factors in Computers, Workshop "The What, Who, Where, When, Why and How of Context-Awareness"*.

Pham, C. (2010). *Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario*. PhD thesis, INFO - Dépt. Informatique (Institut Mines-Télécom-Télécom Bretagne-UEB), UBS - Université de Bretagne Sud, UEB - Université Européenne de Bretagne.

Picot-Clémente, R. (2011). *Une Architecture Générique de Systèmes de Recommandation de combinaison d'items. Application au Domaine du Tourisme*. PhD thesis, Université de Bourgogne.

Pinkwart, N., Hoppe, H., Milrad, M., and Perez, J. (2003). Educational scenarios for cooperative use of personal digital assistants. *Journal of Computer Assisted Learning*, 19(3) :383–391.

Platteaux, H. (2002). Apprentissage et navigation dans les multimédias educatif. Technical report.

Quillian, R. (1968). Semantic memory. In *Semantic Information Processing*, pages 216–270. MIT Press.

Quinn, C. (2000). mlearning : Mobile, wireless, in-your-pocket learning. *LiNE Zine. Fall*.

Razmerita, L. (2003). *Modèle Utilisateur et Modélisation Utilisateur dans les systèmes de Gestion des connaissances : une approche fondée sur les ontologies*. PhD thesis, Université Paul Sabatier, Toulouse, France.

Rechenberg, T. (1965). Cybernetic solution path of an experimental problem. Technical report, Royal Air Force Establishment.

Resnick, P. and H.R.Varian (1997). Recommender systems. *Commun. ACM*, 40(3) :56–58.

Rheaume, J. (1993). Les hypertextes et les hypermédias. *Revue EducaTechnologie*, 1(2).

Ricardo, A. B.-Y. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Ryan, N. (1999). Contextml : Exchanging contextual information between a mobile client and the fieldnote server.

Salton, G. (1989). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Salton, G. and McGill, M. (1983). *Introduction to modern information retrieval*. McGraw-Hill, New York.

Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. (2007). The adaptive web. chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg.

Schilit, B., Adams, N., and Want, R. (1994). Context-aware computing applications. In *The 1994 First Workshop on Mobile Computing Systems and Applications*, pages 85–90, Washington, DC, USA. IEEE Computer Society.

Schilit, B., Hilbert, D., and Trevor, J. (2002). Context-aware communication. *IEEE Wireless Commun*, 9(5) :46–54.

Schilit, B. and Theimer, M. (1994). Disseminating active map information to mobile hosts. *IEEE Network : The Magazine of Global Information Exchange*, 8(5) :22–32.

Schilit, B., Theimer, M., and Welch, B. (1993). Customizing mobile applications. In *The USENIX Symposium on Mobile and Location-Independent Computing*, pages 129–138.

Schmidt-Schaub, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1) :1–26.

Shahabi, C., Kashani, F., Yi-Shin, C., and McLeod, D. (2001). Yoda : An accurate and scalable web-based recommendation system. In *Proceedings of the 9th International Conference on Cooperative Information Systems*, pages 418–432, London, UK, UK. Springer-Verlag.

Sharples, M. (2006). How can we address the conflicts between personal informal learning and traditional classroom education ? In *Big Issues in Mobile Learning, Report of a Workshop by the Kaleidoscope Network of Excellence. Mobile Learning Initiative*, pages 21–24, University of Nottingham.

Sharples, M., Sánchez, I. A., Milrad, M., and Vavoula, G. (2009). Mobile learning : Small devices, big issues. In Balacheff, N., Ludvigsen, S., de Jong, T., Lazonder, A., Barnes, S., and L.Montandon, editors, *Technology-Enhanced Learning : Principles and Products*, pages 233–249. Springer Netherlands.

Sheng, Q. and Benatallah, B. (2005). Contextuml : a uml-based modeling language for model-driven development of context-aware web services. In *The 4th International Conference on Mobile Business*, pages 206–212.

Sheth, B. and Maes, P. (1993). Evolving agents for personalized information filtering. In *Proceedings of the 9th IEEE Conference on Artificial Intelligence for Applications*.

Siadaty, M., Torniai, C., Gasevic, D., Jovanovic, J., Eap, T., and Hatala, M. (2008). m-loco : An ontology-based framework for context-aware mobile learning. In *International Workshop on Ontologies and Semantic Web for Intelligent Educational Systems collocated with the 9th International Conference on Intelligent Tutoring Systems*.

Soylu, A., Causmaecker, P. D., and Desmet, P. (2009). Context and adaptivity in context-aware pervasive computing environments. *Ubiquitous, Autonomic and Trusted Computing, 2009, UIC-ATC 09, Symposia and Workshops on*, pages 94–101.

Stefani, A. and Strappavara, C. (1998). Personalizing Access to Web Sites : The SiteIF Project. *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98*.

Strang, T. and Linnhoff-Popien, C. (2004). A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, Nottingham, England.

Syvanen, A., Beale, R., Sharples, M., Ahonen, M., and Lonsdale, P. (2005). Supporting pervasive learning environments : Adaptability and context awareness in mobile learning. In *The IEEE International Workshop on Wireless and Mobile Technologies in Education*, pages 251–253, Washington, DC, USA. IEEE Computer Society.

Tadlaoui, M., Chikh, A., and Bouamrane, K. (2010). Un modèle de référence pour les applications web adaptatif Éducatif. In *LEAFA 2010 : The First International Conference on e-Learning For All*, Hammamet, Tunisia.

Tarski, A. (1941). On the calculus of relations. *Journal of Symbolic Logic*, 6(3) :73–89.

Traxler, J. (2005). Mobile learning : It's here, but what is it ? In Kukulska-Hulme, A. and Traxler, J., editors, *A Handbook for Educators and Trainers*. London, Routledge.

Traxler, J. (2007). Defining, discussing, and evaluating mobile learning : The moving finger writes and having writ. . . . *International Review of Research in Open and Distance Learning*, 8(2).

Vozalis, E. and Margaritis, K. (2003). Analysis of recommender systems' algorithms. In *The 6th Hellenic European Conference on Computer Mathematics & its Applications (HERCMA)*, Athens, Greece.

Wang, X., Zhang, D., Gu, T., and Pung, H. (2004). Ontology based context modeling and reasoning using owl. In *The Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Washington, DC, USA. IEEE Computer Society.

Weiser, M. (1991). The computer for the 21st century. *Scientific American Special Issue on Communications, Computers, and Networks*.

Widmer, M. (2001). Les métaheuristiques : des outils performants pour les problèmes industriels. In *3ème Conférence Francophone de MOdélisation et SIMulation : Conception, Analyse et Gestion des Systèmes Industriels, MOSIM'01*.

Wiley, D., Recker, M., and Gibbons, A. (2000). The reusability paradox.

Winograd, T. (2001). Architectures for context. In *Human-Computer Interaction*, pages 402–419.

Yin, C. (2010). *SAMCCO : un Système d'Apprentissage Mobile Contextuel et Collaboratif dans des Situations Professionnelles*. PhD thesis, Ecole Centrale de Lyon.

LISTE DES ABRÉVIATIONS

CAMLearn : Context-Aware Mobile Learning
ABox : Assertions Box
ACO : Ant Colony Optimization
AEP : Association of Educational Publishers
AFNOR : Association française de normalisation
AHAM : Adaptive Hypermedia Application Model
AL : Attribute Language
ALEM : Adaptive Learning Environment Model
AMULETS : Advanced Mobile and Ubiquitous Learning Environments for Teachers and Students
API : Application Program Interface
BOS : Bag-Of-Synsets
CASS : Context-Awareness Sub-Structure
CC : Creative Commons
CC/PP : Composite Capabilities/Preferences Profiles
CEdMA : Computer Education Managers Association
Checksem : Semantic Intelligence Research
CMF : Context Management Framework
CML : Context Modeling Language
CMS : Course Management System
CoBrA : Context Broker Architecture
ConteXtML : Contexte Markup Language
DGT : Grand Dictionnaire Technologique
DRIM-AP : Dispositifs Radios Interactifs Multiples et Amphis Participatifs
DTD : Document Type De ?nition
EAO : Enseignement Assisté par Ordinateur
EDA : Estimation of Distribution Algorithm
EIAO : Enseignement Intelligent Assisté par Ordinateur
FIFO : First In First Out
FIPA : Foundation for Intelligent Physical Agents
GPS : Global Positioning System
IADIS : International Mobile Learning Conference
IAmLearn : International Association for Mobile Learning

IEEE : Institute of Electrical and Electronics Engineers
IMS-LD : IMS Learning Design
ISO : International Organization for Standardization
ITR : ITerm Recommender
LabSET : Laboratoire de Soutien à l'Enseignement Téléematique
LD : Logique de Description
LE2I : Laboratoire Electronique, Informatique et Image
LMS : Learning Management Systems
LMS : Learning Management System
LO : Learning Object
LOM : Learning Object Metadata
LRMI : Learning Resource Metadata Initiative
LRS : Learning Record Store
LSS : Learning Support System
LTSC : Learning Technology Standards Committee
MARA : Mobile Augmented Reality Applications
MEV : Modèle d'Espace Vectoriel
MLE : Managed Learning Environment
MLR : Management Learning Resources
MoULe : Mobile and Ubiquitous Learning
MP3 : Mpeg Audio Layer 3
NLS : oN Line System
OCL : Object Constraint Language
ONON : CONtext ONtology
OWL : Web Ontology Language
PC : Personal Computer
PDA : Personal Digital Assistant
RDF : Resource Description Framework
RDFS : RDF Schema
RDIF : Radio Frequency IDentification
RIF : Rule Interchange Format
RuleML : Rule Markup Language
SCO : Sharable Content Object
SCORM : Sharable Content Object Reference Model
SE : Stratégies d'Evolution
SGA : Système de Gestion de l'Apprentissage
SGBD : Système de gestion de base de données
SGBDO : Systèmes de Gestion de Base de Données Objet
SGML : Standard Generic Markup Language
SGRM : Système de Génération de Règles Métiers

SOCAM : Service-Oriented Context-Aware Middleware
SPARQL : Protocol and RDF Query Language
SQL : Structured Query Language
STI : Systèmes Tutoriels Intelligents
SWRL : Semantic Web Rule Language
T2T : Text to Teach
TBox : Terminology Box
TFIDF : Term Frequency-Inverse Document Frequency
TIC : Technologies d'Information et de Communication
UIT : Union Internationale des Télécommunications
UML : Unified Modeling Language
URI : Uniform Resource Identifier
UTC : Coordinated Universal Time
VAK : Visuel, Auditif, Kinesthésique
VLE : Virtual Learning Environmen
VNS : Variable Neighborhood Search
W3C : World Wide Web Consortium
WWW : World Wide Web
XML : eXtended Markup Language
XML : Extensible Markup Language
XSL : Extensible Stylesheet Language

TABLE DES FIGURES

1.1 Pourcentage des salariés ayant suivi une formation e-learning . . 19
1.2 Évolution de l'usage des technologies de télécommunication et mobiles 20
1.3 Évolution de l'usage des dispositifs mobiles 20
1.4 Cartographie du m-learning, version 1 21
1.5 Cartographie du m-learning, version 2 23

2.1 Architecture minimale d'un système sensible au contexte 44
2.2 Architecture de Context Toolkit 46
2.3 Architecture de CMF 47
2.4 Architecture de CoBrA 48
2.5 Architecture de SOCAM 49
2.6 Architecture de CASS 50
2.7 Architecture en couches (Daruwala, 2008) 51
2.8 Architecture générale de m-LOCO 53

3.1 Recommandation basée sur le contenu 60
3.2 Filtrage collaboratif 65

4.1 Cycle de vie d'un LO 81
4.2 Carte heuristique des métadonnées de LOM 86
4.3 Architecture de la spécification IMS-LD 88
4.4 Composition du content package 88
4.5 Le modèle d'agrégation de contenu SCORM 90
4.6 Le modèle de séquencement et de navigation SCORM 92
4.7 Modèle des objets pédagogiques 94

5.1 Architecture générale du système m-learning 100
5.2 Architecture du serveur sémantique de connaissances 101
5.3 Architecture en couches des standards du Web sémantique 103
5.4 Aperçu simplifié du modèle des LO 110
5.5 Aperçu simplifié des concepts de la dimension spatiale 112
5.6 Aperçu simplifié des concepts de la dimension temporelle 114
5.7 Aperçu simplifié des différents acteurs du système 115
5.8 Aperçu simplifié des concepts de la dimension utilisateur 116
5.9 Aperçu simplifié des concepts de la dimension dispositif 118
5.10 Exemple de contexte 120
5.11 Interface de gestion des règles 123
5.12 Architecture du modèle d'adaptation 125

6.1 Aperçu simplifié du graphe des LO 132
6.2 Parcours mono-support 133
6.3 Modélisation simplifiée multi-support 135
6.4 Fonction de coût d'un problème combinatoire 138
6.5 Principes généraux d'une métaheuristique à base de solution unique 139
6.6 Principes généraux d'une métaheuristique à population de solutions 139
6.7 Tournée du voyageur de commerce 140
6.8 Exploration de l'espace des solutions par la méthode de recherche locale 141
6.9 Variation de l'énergie par l'algorithme de recuit simulé 144
6.10 Exploration par une approche évolutive 148
6.11 Comportement d'une colonie de fourmis 153
6.12 Exploration de l'espace des solutions par une méthode VNS . . . 161
6.13 Introduction de la perturbation dans la recherche à voisinage variable 161

A.1 Méthodes et technologies des hypermédia adaptatifs 183
A.2 Modèle Dexter (Halasz and Schwartz, 1994) 184
A.3 Modèle AHAM (Bra et al., 1999) 185

A.4 Modèle Munich (Koch and Wirsing, 2002) 186
A.5 Modèle LAOS (Cristea and Andmethods, 2003) 187
A.6 Modèle ALEM (Tadlaoui et al., 2010) 188

B.1 Évolution des logiques de description 198

C.1 Triplet RDF 210
C.2 Exemple de graphe RDF 211

LISTE DES TABLES

1.1 Tableau comparatif de LMS 16

2.1 Évaluation des approches de modélisation du contexte par rapport aux critères de (Strang and Linnhoff-Popien, 2004), (Yin, 2010) . . 43

2.2 Exigences pour les systèmes sensibles au contexte (Daruwala, 2008) 54

3.1 Forces et faiblesses des approches de recommandation 68

4.1 Liste des éléments du Dublin Core 85

5.1 Concepts de la dimension dispositif 119

6.1 Valeurs moyennes de $Pert$ pour les algorithmes G, HC, RS, T et VNS165

6.2 Temps moyen d'exécution pour les algorithmes G, HC, RS, T et VNS166

B.1 La grammaire du langage de description de concepts selon $\mathcal{AL}$. 194

B.2 La sémantique du langage de description de concepts selon $\mathcal{AL}$. 195

B.3 Constructeurs des logiques de description 196

B.4 Constructeurs : syntaxe et sémantique 196

B.5 Familles de logique de description 197

B.6 Constructeurs OWL 199

B.7 Axiomes OWL 2 199

B.8 Moteurs d'inférence 204

C.1 Formats RDF supportés par chaque les triplestores 214

C.2 Langages de requêtes supportés 215

C.3 Moteurs d'inférence 216

C.4 Langages de requêtes supportés 217

C.5 Licence des triplestore 218

Résumé :

Au vu de l'émergence rapide des nouvelles technologies mobiles et la croissance des offres et besoins d'une société en mouvement en formation, les travaux se multiplient pour identifier de nouvelles plateformes d'apprentissage pertinentes afin d'améliorer et faciliter le processus d'apprentissage à distance. La prochaine étape de l'apprentissage à distance est naturellement le port de l'apprentissage électronique vers les nouveaux systèmes mobiles. On parle alors de m-learning (apprentissage mobile). Jusqu'à présent l'environnement d'apprentissage était soit défini par un cadre pédagogique soit imposé par le contenu d'apprentissage. Maintenant, nous cherchons, à l'inverse, à adapter le cadre pédagogique et le contenu d'apprentissage au contexte de l'apprenant. Nos travaux de recherche portent sur le développement d'une nouvelle architecture pour le m-learning. Nous proposons une approche pour un système m-learning contextuel et adaptatif intégrant des stratégies de recommandation de scénarios de formations sans risque de rupture.

Mots-clés : M-learning, Contexte, Recommandation, Web sémantique, Algorithmes combinatoires.

Abstract:

Given the rapid emergence of new mobile technologies and the growth of needs of a moving society in training, works are increasing to identify new relevant educational platforms to improve distant learning. The next step in distance learning is porting e-learning to mobile systems. This is called m-learning. So far, learning environment was either defined by an educational setting, or imposed by the educational content. In our approach, in m-learning, we change the paradigm where the system recommends content and adapts learning follow to learner's context.

Keywords: M-learning, Context, Recommendation, Semantic Web, Combinatorial algorithms.

SPIM

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex
■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UNIVERSITÉ DE BOURGOGNE