

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Djilali BOUNAAMA KHEMIS MILIANA



Faculté des Sciences et de la Technologie
Département des Mathématique et Informatique

Mémoire Présenté

Pour l'obtention de diplôme

Master en« Informatique»

Option :«Ingénierie du logiciel»

Titre :

**Evaluation des connaissances des apprenants-
application pour le M-Learning**

Réalisé par :

- Bouallouche Salah eddine

-NaciFaycal

Devant le jury composé de :

Mr S.HADJ SADOK.....Président

Mme F.BOUDALI.....Encadreur

Mr N.AZZOUZA.....Examineur1

Mr F.HANNICHE.....Examineur2

Année Universitaire : 2016/2017.

Remerciements

Nous remercions dieu le tout puissant qui nous a donné courage et volonté pour réaliser notre projet de fin d'études.

Qu'il nous soit permis d'adresser nos sincères remerciements à ceux qui de près ou de loin ont concouru à l'aboutissement de cet ouvrage et qui dans leurs labeurs quotidiens contribuent à repousser les bornes de l'ignorance. On pense particulièrement :

- ☛ *Notre encadreur Mme Boudali Fatiha pour leur aide, orientation et bienveillance.*
- ☛ *Aux membres du jury Ms Hadj sadouk salaheddine, Ms Haniche fayçal et Ms Azzouzanoureddine d'avoir accepté de juger notre travail ;*
- ☛ *Nous remercions également tous nos enseignants pour toutes les connaissances qu'ils nous ont inculquées ;*
- ☛ *Un grand merci a notre ami Ms Belmechri Nassim de son collaboration, son aide , son encouragement, sans ce personne ce projet n'aurait jamais vu le jour ;*
- ☛ *A toutes les personnes, souvent anonymes, qui ne comptent pas leurs temps pour mettre à notre disposition des documents de grande valeur ;*
- ☛ *Nous n'oublions surtout pas nos camarades pour leurs soutiens, questions et remarques pertinentes.*
- ☛ *Qu'assurance soit faite à tous ceux qui n'ont pas été explicitement cité ici, de toute l'estime que nous portons pour eux, de notre gratitude et profondes sympathie.*

Dédicace

Je dédie cet ouvrage à toutes les personnes qui m'ont soutenu durant les moments difficiles, en particulier :

- ♥ *A mon PÈRE, je voie un père dévoué à sa famille. Ta présence en toute circonstance m'a maintes fois rappelé le sens de la responsabilité ;*
- ♥ *A celle qui m'a mise au monde « ma Mère », aucun mot ne saurait exprimer ce sentiment de reconnaissance de m'avoir bercée et consolée, de m'avoir si bien guidée, protégée, encouragée et tout simplement aimée. Merci de tout cela.*
- ♥ *A toutes les personnes qui partagent aussi bien mes instants de bonheur que de malheur, ceci sous entend mon jardin d'amour rempli de : mes deux frères AHMED et Hamid je vous remercie pour m'avoir toujours supporté dans mes décisions. Merci pour tout votre confiance, pour m'avoir aidé a rangé mon éternel désordre et pour votre énorme support pendant la rédaction de mon projet, Et mes huit fleurs, mes sœurs : Fouzia, Samra, Khira, fatimazahra, Hanane, Fadhila, Nawel, Chaima et Ritej Pour leur soutien morale et leurs sacrifices .*
- ♥ *A mes beaux-frères Sofiane, Mohamed, Ahmed, Ali, Bellkacemet Mohamed.*
- ♥ *A Tous Les Membres De Ma Famille, Petits Et Grands.*
- ♥ *A mes meilleures amies et mes frères Hamza, Amine, Billel, Youcef, Abdou Salam et Saïd pour leur aide, leur temps, leur encouragement, leur assistance et soutien, sans oublier mon binôme Fayçal*
- ♥ *A toutes mes chères amies Amine, Imen, Rania et ma chère Nouha avec qui je partage des moments pendant cette année, je vous offre cette magnifique dédicace d'amitié.*

SALAH EDDINE

Dédicace

Je dédie cet ouvrage à toutes les personnes qui m'ont soutenu durant les moments difficiles, en particulier :

- ♥ *A mon **PERE**, je voie un père dévoué à sa famille. Ta présence en toute circonstance m'a maintes fois rappelé le sens de la responsabilité ;*

- ♥ *A celle qui m'a mise au monde « ma **Mère** », aucun mot ne saurait exprimer ce sentiment de reconnaissance de m'avoir bercée et consolée, de m'avoir si bien guidée, protégée, encouragée et tout simplement aimée. Merci de tout cela.*

- ♥ *A toutes les personnes qui partagent aussi bien mes instants de bonheur que de malheur, ceci sous entend mon jardin d'amour rempli de : mes deux frères **Chakibeet Farouk**. je vous remercie pour m'avoir toujours supporté dans mes décisions. Merci pour tout votre confiance, pour m'avoir aidé a rangé mon éternel désordre et pour votre énorme support pendant la rédaction de mon projet, Et mes huit fleurs, ma sœur : **Hasnaa** Pour leur soutien morale et leurs sacrifices*
.

- ♥ *A mes meilleures amies **Khalil, Mustapha, Alaa , Djalil , Hichem** , et a mes amies du petite groupe **Amine Rania Imene**, pour leur aide, leur temps, leur encouragement, leur assistance et soutien, sans oublier mon binôme **Salah***

FAYSSAL

Sommaire

Liste Des Figures.....	I
Résumé.....	II
Introduction générale	4
I. EIAH et Evaluation des Apprenants.....	- 3 -
I.1. Introduction	- 3 -
I.2. L'Informatique pour l'Apprentissage Humain.....	- 3 -
I.3. E-Learning.....	- 5 -
I.3.1. Définitions	- 5 -
I.3.2. Typologies des systèmes d'e-Learning.....	- 5 -
I.3.3. Plates formes d'e-Learning	- 6 -
• Du côté de l'apprenant :	- 6 -
• Du côté de l'entreprise :	- 7 -
I.4. Evaluation des Apprenants	- 7 -
I.4.1. Types d'évaluation	- 8 -
I.5. Entrer dans l'âge de l'apprentissage mobile.....	- 9 -
I.5.1. Le passage de l'apprentissage électronique au l'apprentissage mobile.....	- 9 -
I.5.2. Apprentissage mobile VS apprentissage électronique.....	- 10 -
I.5.3. Définition de l'apprentissage mobile	- 10 -
I.5.4. Histoire de l'apprentissage mobile.....	- 11 -
I.6. Caractéristiques et classifications	- 13 -
I.6.1. Caractéristiques de l'apprentissage mobile	- 13 -
I.7. L'avenir de l'apprentissage mobile	- 14 -
I.8. Intérêts et limites	- 15 -
II.10.1. L'intérêt du m-Learning.....	- 15 -
I.10.2. Limites et risques	- 15 -
I.9. Exemple d'approche d'évaluation.....	- 16 -
I.9.1. Le processus de conception	- 16 -
I.9.2. Le processus d'utilisation	- 17 -
I.10. Conclusion	- 18 -
II. Les Ontologies	- 23 -
II.1. Introduction	- 23 -
II.2. Web sémantique.....	- 23 -

II.3.	Définition d'une ontologie	- 25 -
II.4.	Constituants d'une ontologie.....	- 26 -
II.4.1.	Concepts.....	- 26 -
II.4.2.	Relations.....	- 28 -
II.4.3.	Axiomes.....	- 29 -
II.4.4.	Instances.....	- 29 -
II.5.	Types d'ontologie.....	- 29 -
II.6.	Développement d'une ontologie	- 33 -
II.7.	Langages de représentations	- 36 -
II.7.1.	XML.....	- 36 -
II.7.2.	KIF.....	- 37 -
II.7.3.	RDF	- 37 -
II.7.4.	DAML +OIL.....	- 38 -
II.7.5.	OWL.....	- 38 -
II.8.	Outils de développement des ontologies	- 38 -
II.8.1.	Ontolingua.....	- 38 -
II.8.2.	OntoSaurus.....	- 39 -
II.8.3.	WebOnto	- 39 -
II.8.4.	OilEd	- 39 -
II.8.5.	Protégé2000.....	- 39 -
II.8.6.	ODE etWebOde	- 40 -
II.8.7.	OntoEdit	- 40 -
II.9.	Alignement Des Ontologies.....	- 40 -
II.9.1.	Définition d'alignement	- 41 -
II.9.2.	Dimensions de l'alignement.....	- 42 -
II.9.3.	Etapes du processus d'alignement.....	- 43 -
II.9.4.	Similarité	- 44 -
❖	Quelques techniques de similarité et de distance	- 46 -
II.10.	Méthodes d'alignement.....	- 47 -
II.10.1.	Méthodes terminologiques.....	- 48 -
II.10.2.	Méthodes structurelles	- 48 -
II.10.3.	Méthodes extensionnelles.....	- 48 -
II.10.4.	Méthodes sémantiques.....	- 49 -
II.10.5.	Méthodes fondées sur la satisfaisabilité propositionnelle.....	- 49 -

II.10.6.	Méthodes basées sur les logiques de description	- 49 -
II.10.7.	Méthodes basées sur les langages de représentation d'ontologies	- 49 -
II.11.	Outils de l'alignement des ontologies	- 49 -
II.12.	Frameworks d'alignement	- 51 -
II.12.	Conclusion	- 52 -
III.	Conception et mise en œuvre	- 53 -
III.1.	Introduction	- 53 -
III.2.	UML (Unified Modeling Language)	- 53 -
II.2.1.	Définition	- 53 -
II.2.2.	Modèles d'UML	- 54 -
III.3.	Conception	- 55 -
III.3.1.	Diagrammes	- 56 -
III.4.	Environnement de développement	- 59 -
III.4.1	Environnement matériel	- 59 -
III.4.2	Environnement Logiciel	- 59 -
III.4.2.1	Langage JAVA	- 59 -
III.4.2.2	Langage PYTHON	- 60 -
III.4.2.3	Lien De Communication Entre Les Différentes Plateformes	- 61 -
4.2.3.1	Sockets :	- 61 -
4.2.3.2	Communication entre la plateforme Android et python :	- 61 -
III.4.2.4	NetBeans	- 62 -
III.4.2.5	Android Studio	- 63 -
III.4.2.6	Pycharm	- 64 -
III.4.2.6	Les Bibliothèques utilise	- 64 -
III.4.3.	Architecture globale de l'application	- 65 -
III.4.3.1.	Traitement des Réponses et création des ontologies :	- 66 -
III.4.3.2.	La comparaison des ontologies et le systèmes des notes	- 67 -
III.5.	Définition fonctionnelle	- 68 -
III.6.	Conclusion	- 77 -
	Conclusion générale	- 86 -
	Bibliographie	- 89 -

Liste Des Figures

Figure I-1: Principaux courants en informatique et éducation Avant 1997 [Bruillard 97]	- 4 -
Figure I-2: Différents formats de l'apprentissage durant les années.	- 9 -
Figure I-3: Définitions d'apprentissage mobile.....	- 10 -
Figure I-4: Caractéristiques de l'apprentissage mobile.	- 13 -
Figure I-5: Principales étapes de développement d'un système d'évaluation avec ODALA.[Bouarab-Dahmani & al. 10b].....	- 17 -
Figure I-6: Principales étapes du fonctionnement d'un système d'évaluation.[Bouarab-Dahmani & al. 10b].....	Error! Bookmark not defined.
Figure II-1 : Couches du standard du Web sémantique [Klein, M., 2001].....	- 25 -
Figure II-2 : Exemple d'une ontologie.	- 26 -
Figure II-3 : Types d'ontologie : ce type est class selon la conceptualisation	- 30 -
Figure II-4 : Classification des ontologies[Natalya F. Noy et Deborah L. McGuinness]	- 33 -
Figure II-5 : Exemple d'alignement entre deux ontologies [Euzenat, J., Scharffe, F. & Zimmermann 2007].....	- 42 -
Figure II-6 : Processus d'alignement[Euzenat, J., Scharffe, F. & Zimmermann 2007].....	- 43 -
Figure II-7 : Etapes du processus d'alignement [Lynda DJAKHDJAKHA]	- 44 -
Figure III-1: Diagramme de classe.	- 56 -
Figure III-2: Diagramme de séquence.	- 58 -
Figure III-3: Shema du processus de traitement d'une question libre.	- 58 -
Figure III-4: architecture global d'une creation d'une ontologie.....	- 58 -
Figure III-5: page d'accueil de l'application.	- 58 -
Figure III-6: Menu enseignant.....	- 58 -
Figure III-7: Menu etudiant.	- 58 -
Figure III-8: Menu des questions libres(etudiant).	- 58 -
Figure III-9: la zone de reponse libre de l'etudiant.	- 58 -
Figure III-10: la note de la reponse de l'etudiant.	- 58 -
Figure III-11: la liste des questions a choix pour l'etudiant.	- 58 -
Figure III-12: les choix a coché par l'etudiant.	- 58 -
Figure III-13:note de reponse de l'etudiant.	- 58 -

Résumé

L'apparition des mobile dans ces dernier année a ouvert une porte au développement de plusieurs technologie et il a aussi double et incrémente le flux des donnés sur le web et pour mieux contrôle ces donnes on utilise le web sémantique, dans le cadre pédagogique le mobile a poussée aussi l'apprentissage a distant vert a nouveau monde du M Learning, toutefois il soulève un certain nombre de défis de recherche.

Un des principaux défis est évaluation à partir une question libre on utilisant les ontologies ce défit concentre sure la transformation du texte au une ontologie et la faire aligné avec une autre dans le but de comparai ces deux pour sortir avec un résultat qui se sera une évaluation pour des réponses par texte libre

Mots clés :

ONTOLOGIE, WEB SEMANTIQUE, APPRENTISSAGE A DISTANT, M-LEARNING,

E-LEARNING, APPLICATION MOBILE, ANDROID.

Introduction générale

Introduction générale

Le but principale de l'apprentissage a distant est de faciliter la tache au enseignant et au étudiant dans le domaine pédagogique et il contient plusieurs outil d'utilisation, l'un des plus récent et plus facile outils est l'outil mobile qui a fait passer l'apprentissage a distant dans un autre niveau

Les ontologies sont utilisées pour la représentation des connaissances. Plusieurs ontologies couvrant totalement ou partiellement un même domaine d'application peuvent être développées indépendamment les unes des autres. Les entités de ces ontologies peuvent être définies selon différents niveaux de granularité ou simplement décrites grâce à des différents langages de représentation. Afin de permettre l'interopérabilité et le partage des données, il est primordial que l'hétérogénéité entre les connaissances décrites dans les ontologies soit résolue et ceci par la création des liens sémantiques entre les entités, c'est le but de l'alignement.

Ce mémoire est composé de trois chapitres :

— Chapitre 1 : ce chapitre présente Environnements Informatiques d'Apprentissage Humains et l'utilité de l'utilisation d'une application mobile pour l'apprentissage a distant

.

— Chapitre2 : ce chapitre définit le concept de l'ontologie et décrit ces défirent notion et l'intérêt d'utilise les ontologiesd une application m-Learning

.

— Chapitre3 : ce chapitre est dédié à la conception et à l'implémentation de notre Travail. Il est consacré à la présentation des outils utilisés pour la réalisation de notre Plateforme. Nous décrivons aussi l'architecture globale et le fonctionnement de notre Système.

Chapitre I :
EIAH et Evaluation des
Apprenants

I. EIAH et Evaluation des Apprenants

I.1. Introduction

Depuis l'apparition de l'informatique, beaucoup de systèmes sont développés dans le but d'utiliser l'ordinateur pour aider à accomplir la tâche pédagogique d'enseignement/apprentissage ,deux grandes périodes ont marqué l'histoire de la technologie de l'apprentissage qui a contribué à l'élaboration de beaucoup de logiciels plus ou moins réussis. Nous distinguons l'ère de l'enseignement assisté par ordinateur (EAO) classique où furent effectuées les premières tentatives de besoin d'assistance à l'apprenant par des didacticiels, et l'ère de l'EIAO qui a exploité à bon escient les techniques de l'intelligence artificielle pour réaliser des environnements d'enseignement/apprentissage plus intelligents et qui tiennent compte des capacités cognitives de l'apprenant. On convient que tous ces systèmes sont des Environnements Informatiques d'Apprentissage Humains (EIAH).

Après une présentation sur les différentes phases qui ont marqués l'histoire de l'apprentissage par ordinateur, nous détaillons au niveau de la deuxième section les EIAH de type e-learning puis présentons une synthèse sur l'évaluation des connaissances des apprenants.

I.2. L'Informatique pour l'Apprentissage Humain

Le domaine des technologies appliquées à la formation recouvre ce qu'on appelait traditionnellement des didacticiels, et qu'on appelle plutôt aujourd'hui des EIAH. Les environnements informatiques pour l'apprentissage humain EIAH sont des environnements informatiques conçu dans le but de favoriser l'apprentissage humain mobilisant des agents humains et artificiels utilisé dans des situations d'interaction prudentielles ou à distance.

L'enseignement programmé était le début des méthodes de l'apprentissage humain, d'où un enseignement par question réponse qui conduisait à la fois à des réflexions théoriques ainsi qu'à la mise en place des expérimentations, mais réellement c'était des machines à tester pas à enseigner.

Ensuite, l'apport des technologies informatique à l'enseignement précédent mène à l'EAO (Enseignement assisté par ordinateur), en anglais CBT (Computer based training) qui permet à l'apprenant d'apprendre à son rythme et participer à des simulations de situations et de phénomènes. L'intégration de l'intelligence artificielle a conduit à parler d'EIAO (Enseignement

intelligemment assisté par ordinateur), en anglais ICAI (Intelligent computer aided instruction) qui amène plus de souplesse, interactivité et d'adaptation à l'apprenant. On a ensuite traduit le I par interactif : EIAO devenait Environnement interactif d'apprentissage avec ordinateur ou EIAO. Cela créait toutefois quelque confusion avec l'apprentissage par les machines au sens de l'IA (intelligence artificielle); d'où le sigle actuel EIAH.

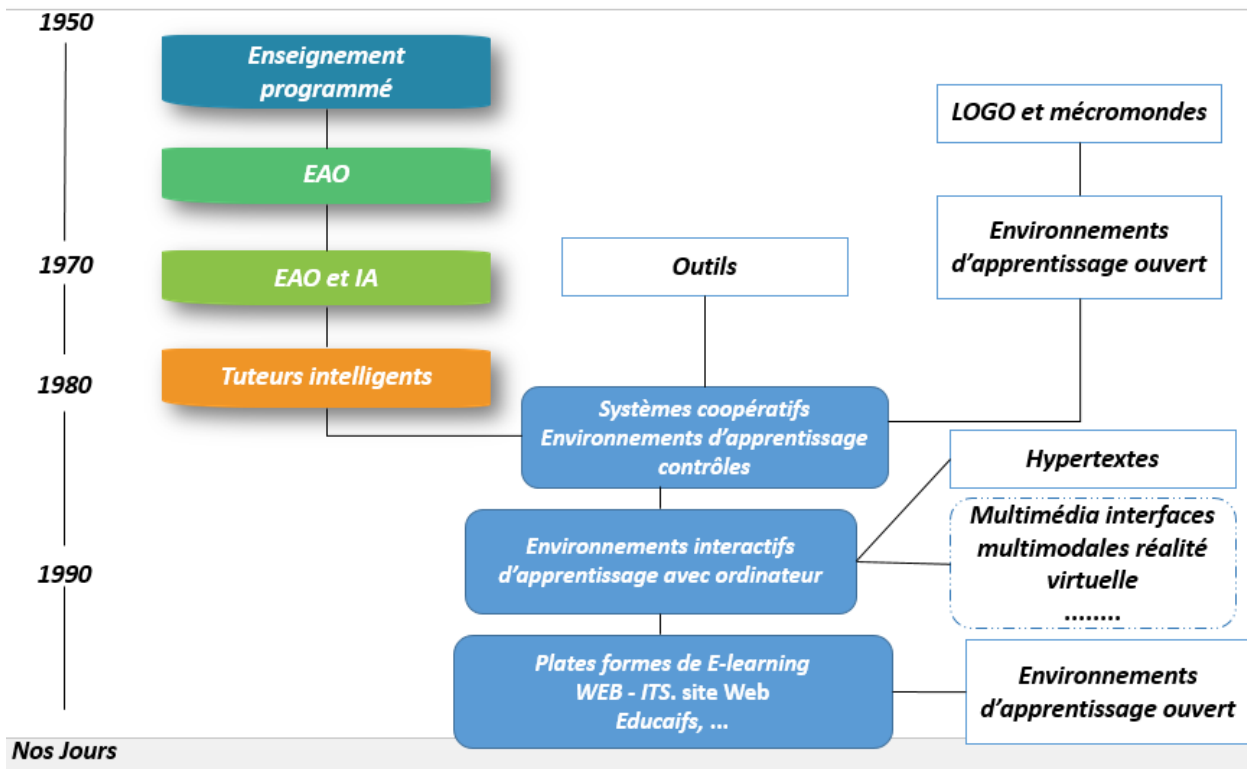


Figure I-1: Principaux courants en informatique et éducation Avant 1997 [Bruillard 97]

La première colonne représente les types de technologies qui ont privilégié l'enseignement, c'est-à-dire pour lesquels la machine principalement joue le rôle du maître (de l'enseignement programmé aux tuteurs intelligents). La troisième colonne présente des recherches dans lesquelles l'ordinateur est un moyen permettant aux apprenants d'effectuer des constructions, explorations et découvertes. La colonne centrale correspond aux tentatives de synthèse de ces deux approches, essayant de concilier un certain guidage par la machine dans des environnements largement contrôlés par les apprenants.

Il existe une grande diversité d'EIAH, allant de simples CD-Rom à de complexes plateformes du E-Learning. Les EIAH comprennent à la fois [Grandbastien 02] :

- Les micros modes, des univers isolés contenant des objets simplifiés permettant l'apprentissage par la manipulation et la planification.
- Des produits appelés "tuteurs intelligents", destinés à favoriser un certain apprentissage, par exemple en géométrie, ou en langues étrangères.
- Des services à valeur ajoutée, comportant d'autres fonctions comme le courrier électronique ou la gestion des élèves.
- Des plateformes de simulation, les outils de réalité virtuelle et augmentée.

I.3. E-Learning

I.3.1. Définitions

Le e-Learning est un des moyens disponibles pour les formations à distance, il repose sur l'utilisation de supports numériques (internet, intranet, cédérom, etc..) dans le but de rendre accessible des parcours d'autoformations individualisés mais également dans celui de permettre des échanges entre les acteurs des formations (travail collaboratif)¹⁰.

I.3.2. Typologies des systèmes d'e-Learning

Il existe différentes façons de classer les types d'e-Learning ou méthodes d'apprentissages que l'e-Learning va utiliser.

Généralement, les types sont classés selon deux dimensions ou critères :

- Les systèmes synchrones qui définissent une « classe virtuelle » où la communication entre les acteurs de la formation à distance est en temps réel en utilisant le son, la vidéo ou le chat (contact en direct par un outil ou un support capable d'utiliser les plateformes du e-Learning).
- Les systèmes asynchrones où l'élève n'est pas forcément en ligne avec l'enseignant, il peut travailler en déconnectant et communiquer à des moments planifiés ou à la

demande (rendez-vous électronique). Le mode de communication nécessaire et le plus utilisé dans ce genre de systèmes est la messagerie électronique ou les réseaux sociaux.

I.3.3. Plates formes d'e-Learning

Dans une plateforme e-Learning il faut prendre en considération les plans suivants :

-Technique : ça se rapporte aux matériels et logiciels comme la connexion internet (débit, type), système d'exploitation des utilisateurs de la plateforme etc. ...

-Pédagogique (*enseignant / formateur*) : Consiste à la création des parcours pédagogiques, des contenus pédagogiques et suivi de l'activité des apprenants de telle sorte qu'elle soit motivante et attrayante, comme elle prend en charge aussi la distance de l'enseignant.

-Administratif : C'est la gestion de scolarité des apprenants ou e-learners (inscription, relevé de note ...) et la gestion des enseignants ou e-formateurs (planning, recrutement ...).

-Considération de modélisation : C'est l'ingénierie du logiciel pure, c'est le choix des modèles de représentations des connaissances et la conception de l'application (interface/traitements) de manière à faciliter l'interopérabilité avec les systèmes en relation avec la plateforme, modularité, faciliter de mise à niveau pour s'adapter au changement de l'environnement ...

La formation e-Learning détient de nombreux atouts, le système possède aussi ses limites. Le tableau ci-dessous tend à énumérer les avantages et les inconvénients attribués au e-learning, du point de vue de 2 acteurs de la formation e-Learning et le formateur [11].

- Du côté de l'apprenant :

Avantages	Inconvénients
Apprenant acteur de sa formation e-learning	Appréhension de l'outil informatique, réticence face aux nouvelles technologies
Interactivité et attractivité du contenu e-learning	Gestion de son autonomie car pas de cadre de travail imposé
Flexibilité et adaptabilité selon ses disponibilités (heure, lieu)	Gestion de sa motivation et de son implication dans sa formation e-learning, effort de concentration
Formation à son propre rythme indépendamment des autres apprenants	Pas de contacts directs avec le formateur (sauf en cas de formation blended learning)
Auto-évaluation en cours et en fin de cursus	
Suivi personnel de son avancement dans la formation e-learning et bilan des résultats obtenus grâce au tracking	

- Du côté de l'entreprise :

Avantages	Inconvénients
Pré-requis pour évaluer le niveau des apprenants	Pas de contacts directs avec l'apprenant (sauf en cas de formation blendedlearning)
Formation sur des outils interactifs et attractifs pour lui et les apprenants	Transformation du rôle de « sachant » en rôle de « chef d'orchestre »
Flexibilité et adaptabilité selon les disponibilités (heure, lieu)	Passage d'une communication majoritairement orale à écrite
Suivi des apprenants depuis la plateforme (tracking)	

- ❖ Les principales fonctionnalités d'une plate-forme sont :

- Création de cours, de tests et de parcours-types.
- Gestion de documents pédagogiques (indexation, classification, mises à jour,...).
- Gestion d'un espace de travail coopératif entre apprenants et/ou entre enseignants.
- Suivi de l'apprentissage de l'apprenant et son évaluation.
- Gestion de la scolarité de l'apprenant.
- Mise à la disposition de l'apprenant d'outils de travail (éditeurs spécifiques, outils de téléchargement, outils de simulation de T.P,...).
- Mise à la disposition des différents acteurs d'outils de communications ainsi que les procédés de leurs gestions (forums, messagerie, chat, visioconférence,...).

I.4. Evaluation des Apprenants

L'évaluation des connaissances et des compétences de l'apprenant en résolution de problèmes est un aspect fondamental, elle permet d'améliorer la formation de l'apprenant et montrer ses forces et faiblesses comme elle permet aussi d'évaluer la qualité du positif e-learning (efficacité, efficience, équilibre, équité, engagement).

La plupart des gens croient que l'évaluation se produit après l'e-Learning a été complété.

Faux! L'évaluation se produit:

- Avant l'e-Learning (évaluation des besoins) pour planifier l'e-Learning
- Au cours de l'e-Learning (évaluation formative) pour apporter des améliorations.
- Après l'e-Learning (évaluation sommative) pour déterminer les résultats.

Les travaux qui ont porté sur l'évaluation en tant que partie intégrante du processus d'apprentissage permettent d'introduire une certaine catégorisation pouvant faciliter l'adaptation de l'évaluation aux objectifs recherchés. C'est ce que nous examinons dans ce qui suit, dans le but d'extraire un certain nombre d'aspects fondamentaux relatifs aux processus d'évaluation, aux techniques de diagnostic pouvant y être mises en œuvre, ainsi qu'aux différents types de diagnostics auxquels il est possible d'aboutir selon les cas [12].

I.4.1. Types d'évaluation

Selon son moment ou sa finalité, le type d'évaluation varie. En effet, il n'y a pas une mais des évaluations : pronostique diagnostique, formative, sommative... Toutes sont plus ou moins utilisées par l'enseignant, l'apprenant ou l'institution selon qui évalue, et chaque apprenant connaît au cours de son apprentissage différentes formes d'évaluation. Nous pouvons ainsi distinguer cinq types d'évaluation, les quatre premiers types étant cités dans [Bisault& Lavarde 95] :

- L'évaluation pronostique se résume comme un test qui permet de savoir la capacité d'un apprenant à acquérir des connaissances pendant un cycle d'étude, un exemple approprié pourrait en être un concours d'entrée dans un établissement. Il s'agit de recueillir le plus d'informations sur l'apprenant pour pouvoir prévoir de façon rationnelle ses vrais besoins
- L'évaluation diagnostique permet à l'enseignant d'identifier les savoirs et savoir-faire des apprenants. Elle a pour fonction d'établir un bilan des acquis antérieurs et des connaissances. Elle permet donc de s'adapter aux réels besoins et de programmer son enseignement. Elle n'est pas notée puisqu'elle précède les enjeux de la séquence à venir. Elle permet aussi à l'apprenant de s'inscrire dans son apprentissage
- L'évaluation formative, par contre, permet de guider l'élève dans la réalisation de la tâche par un retour d'information de la part du professeur à l'aide d'une liste de critères, par l'évaluation entre pairs. L'évaluation formative intègre le concept d'erreur formative : l'élève progresse en prenant conscience de ses erreurs et en les rectifiant. Elle permet de développer l'auto-évaluation et la Co-évaluation.
- L'évaluation sommative ou certificative a pour fonction l'attestation ou la reconnaissance des apprentissages. Elle survient au terme d'un processus d'enseignement et sert à sanctionner ou à certifier le degré de maîtrise des résultats d'apprentissage des

étudiants. Elle est sous la responsabilité de l'enseignant et doit être réalisée de façon juste et équitable en reflétant les acquis des étudiants.

Enfin, pour être le plus complet sur ce point, et bien qu'il s'agisse là d'une catégorisation différente, nous avons l'auto-évaluation, correspondant aux cas où l'apprenant (qui est généralement un adulte) [Ellis 07]. L'auto-évaluation consiste en la capacité à évaluer soi-même l'apprentissage personnel et son résultat. Les élèves qui savent approximativement juger leur travail de manière réaliste disposent d'une base solide pour leurs apprentissages futurs. Ils sont capables d'organiser eux-mêmes leur apprentissage et de garder leur motivation intacte. Une auto-évaluation comprend, en plus des contenus, le comportement social, d'apprentissage et face au travail.

I.5. Entrer dans l'âge de l'apprentissage mobile

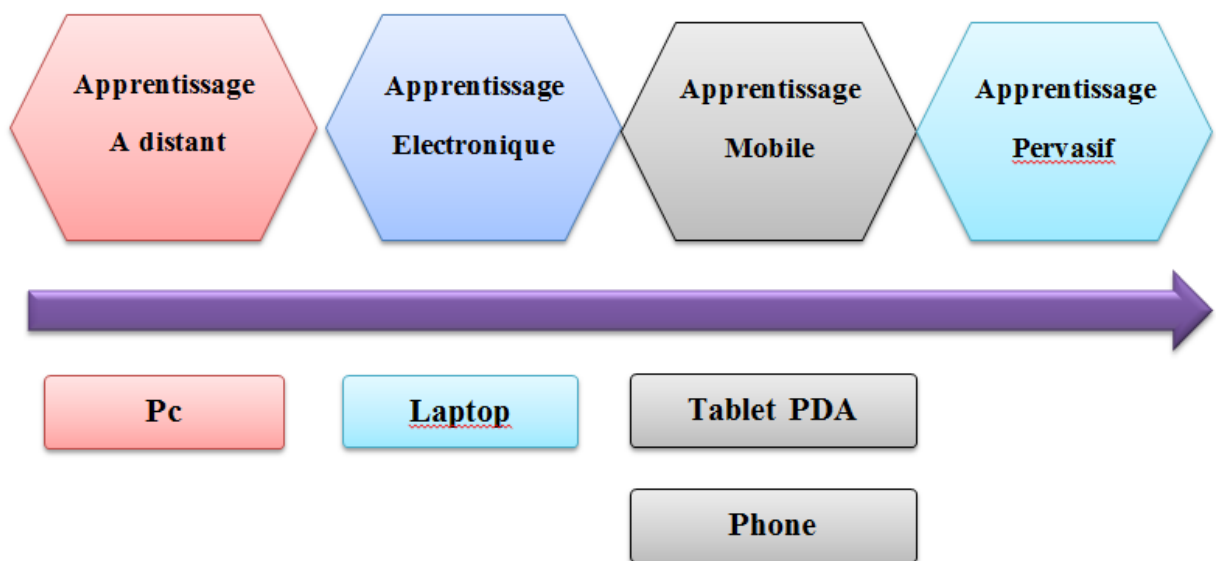


Figure I-2: Différents formats de l'apprentissage durant les années.

I.5.1. Le passage de l'apprentissage électronique au l'apprentissage mobile

La Figure I. 2 représente les différents formats de l'apprentissage durant les années. Avant de parler de l'apprentissage mobile (M-Learning) on parle de l'apprentissage à distance (D-Learning) suivi par l'apprentissage électronique (E-Learning) puis la prise en compte de la

mobilité avec l'apprentissage mobile, et l'omniprésence avec l'apprentissage ubiquitaire. Ces étapes correspondent à l'influence des technologies informatiques, comme l'informatique mobile et l'informatique ubiquitaire. [Chuantao YIN 2011]

I.5.2. Apprentissage mobile VS apprentissage électronique

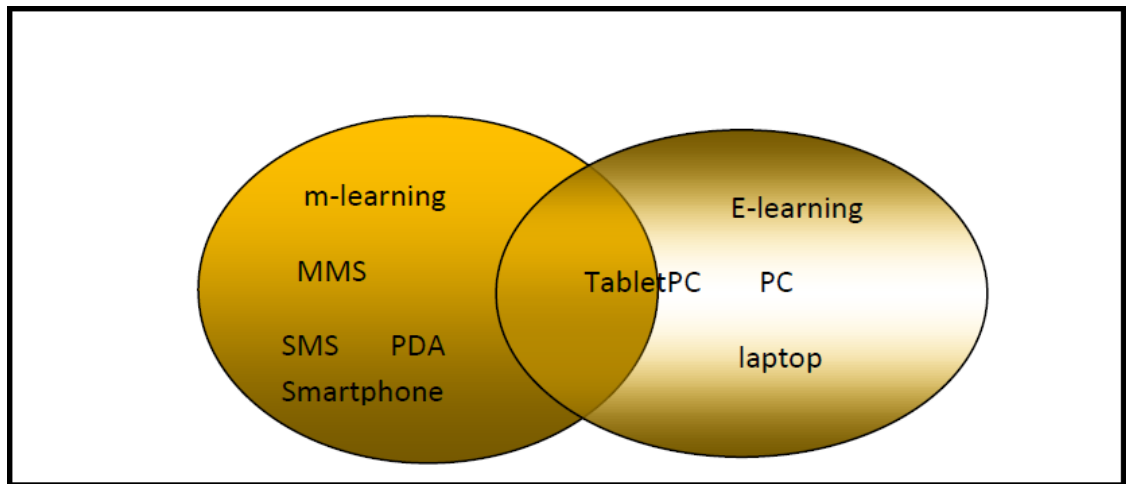


Figure I-3: Définitions d'apprentissage mobile.

La Figure I.3 représente les définitions et descriptions de l'apprentissage mobile basé sur la technologie n'est pas stable. Il existe des définitions qui considère que l'apprentissage mobile est une partie de e-Learning. L'incertitude que laptop et le Tablet PC expriment l'apprentissage mobile illustre la difficulté de ces définitions.

I.5.3. Définition de l'apprentissage mobile

M-Learning désigne la manière d'accéder à des cours à l'aide des supports mobile reçoit des SMS, il existe des définitions considère le m-Learning comme une version d'e-Learning.

❖ Quelques définitions représentatives :

Dans un premier temps, l'apprentissage mobile a surtout été décrit comme un apprentissage qui utilise des équipements mobiles tels que les ordinateurs portables, les PDA et les téléphones portables. [John Traxler2005]

Le mobile Learning également appelé m-Learning, le m-Learning est une notion très récente, qui permet de délivrer des formations à distance sur d'autres supports que les postes

informatiques. Ainsi, grâce aux nombreuses applications développées, l'apprenant peut poursuivre sa formation où qu'il soit grâce à un appareil mobile que ce soit un smartphone (téléphone mobile intelligent qui permet entre autre de naviguer sur internet), un lecteur multimédia comme l'iPod, une tablette mobile telle l'iPad ou encore depuis une console de jeux portative. [Caroline2011]

Depuis la naissance du terme l'apprentissage mobile, la définition de celui-ci n'a jamais arrêté d'évoluer. Jusqu'à aujourd'hui, pendant les conférences spécialisées sur l'apprentissage mobile, par exemple IADIS International Mobile Learning Conférence, la définition de l'apprentissage mobile est encore un point sensible de discussion pour les chercheurs et les éducateurs. [Chuantao YIN 2011]

Les plus récentes définitions de m-Learning sont en train de transférer le centre d'intérêt de la technologie de mobile à la mobilité des apprenants. Ce changement dans la focalisation de la mobilité est important, à l'égard de l'élargissement des multiples dimensions de la mobilité et de l'apprentissage en interaction. Ceci, mène à une perspective plus élargie dans la complexe notion de mobilité dans l'apprentissage, pour permettre d'embrasser, la mobilité dans des espaces physiques et sociaux et la technologie de mobile. [Sharples, & Arnedillo Sánchez 2007]

Le M-Learning est une sorte d'apprentissage qui se produit quand l'apprenant n'est pas à un endroit fixe, ou lorsque l'apprenant profite des technologies mobiles. [Meyer, & al 2006]

L'apprentissage mobile est un moyen moderne de compléter le processus d'apprentissage grâce à des appareils mobiles, tels que les ordinateurs de poche, tablettes numériques, baladeurs MP3, téléphones intelligents et portables. [Chuantao YIN 2011]

Le M-Learning serait un type d'apprentissage électronique, à distance ou en présentiel, qui utilise les technologies mobiles, et conçu pour répondre de façon convenable à la mobilité et aux préférences de l'étudiant moderne. [UNESCO 1995-2011]

I.5.4. Histoire de l'apprentissage mobile

L'histoire de l'apprentissage mobile est décrit sous format des phases indique par Mike Sharples. Il dit que la courte histoire du mobile learning se décompose en trois phases : une première phase focalisée sur les outils, une deuxième sur les apprentissages hors les murs, et une troisième sur la mobilité des apprenants [Pachler Norbert, Bachmair Ben 2010].

La première phase(1990) : est caractérisée par des questionnements sur les outils qui peuvent être utilisés en contexte d'enseignement ou de formation, en particulier les PDA, les ordinateurs portables et les téléphones mobiles. Ces outils sont considérés comme des réceptacles pour stocker et transporter avec soi des ressources multimédias. Par exemple : « iTunes-u5 » développée par Apple, « RLO (reusablelearningobjects) » qui peuvent être lus ou écoutés sur des supports mobiles, au-delà de la simple réalisation de fichiers mp3 ou mp4.

La deuxième phase se base sur les potentialités en termes d'apprentissage en contexte, essentiellement à l'extérieur de l'espace physique délimité par les murs de la classe ou de l'école. Il s'agit dès lors pour les apprenants, non pas d'être simplement destinataires de contenus formatés, mais d'agir ou plutôt de réagir dans un contexte extérieur, qu'il s'agisse d'un voyage d'études. L'outil embarque potentiellement un ensemble de fonctionnalités combinant capture, stockage, partage, et communique avec une plateforme distante (un site Web) permettant les échanges et la mutualisation.

L'environnement physique devient ici signifiant et des informations liées à la géo localisation sont potentiellement disponibles, par exemple le projet suédois « SkattjaktTreasurehunt » qui conduit les apprenants sur un campus universitaire, avec des jeux de questions réponses.

Dans la troisième phase, les projets se focalisent sur la mobilité des apprenants en tant que collectif. Contrairement aux phases indiquées précédemment .l'espace physique de mobilité est enrichi virtuellement grâce à la visualisation d'informations complémentaires, permettant d'appréhender l'espace, modifier ajouter des enregistrements audio ou vidéo à cet espace .par exemple le projet européen « Contsens ». [Laure Endrizzi 2011]

I.6. Caractéristiques et classifications

I.6.1. Caractéristiques de l'apprentissage mobile

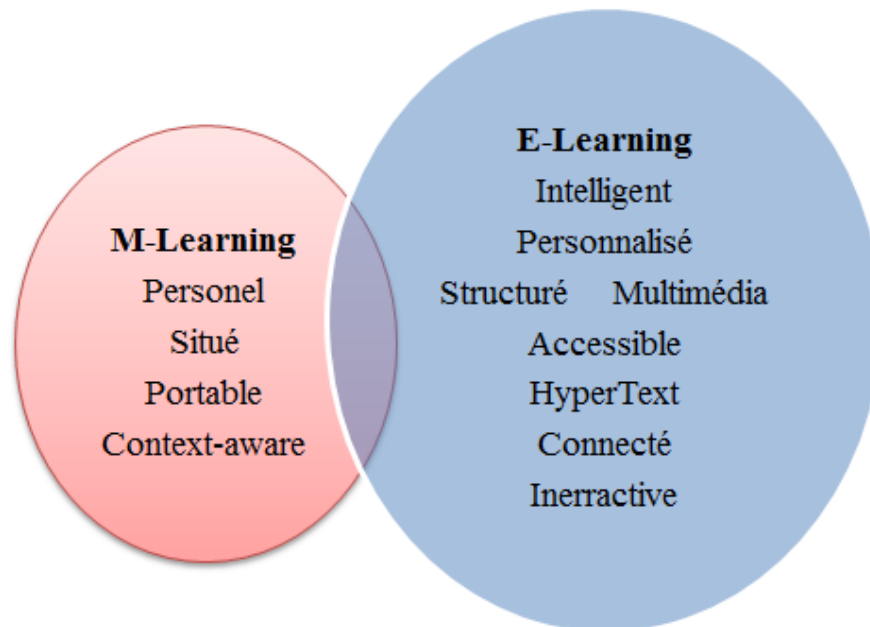


Figure I-4: Caractéristiques de l'apprentissage mobile.

La Figure I.4 représente les Caractéristiques de l'apprentissage mobile par rapport a l'apprentissage électronique.

Meyer définit quatre caractéristiques spécifiques pour l'apprentissage mobile : **Dispositifs, Mobilité, Contexte, Localisation.**

❖ **Dispositifs** : Quatre types de dispositifs sont distingués.

- 1) Dispositif fixe : non déplaçable, toujours connecté, pas de problème d'autonomie. Exemples : ordinateur de bureau, borne interactive dans un campus.
- 2) Dispositif portable : transportable, dispose d'une autonomie (plusieurs heures), connecté de façon non permanente par liaison filière ou radio (Wi-Fi). Exemple ordinateur portable.
- 3) Dispositif mobile : léger, la connexion peut être quasi-permanente. Exemple : PDA, téléphone mobile, E-book, Tablet PC.

4) Dispositif porté : il fait quasiment partie des vêtements, est disponible à tout moment et permet d'avoir les mains libres. Les moyens d'interaction sont complètement différents de ceux des types de dispositifs précédents.

Exemple : Ordinateur porté avec visualisation dans les lunettes.

❖ **Mobilité** : trois types de mobilité sont distingués.

1) Mobilité de l'acteur : un des acteurs de l'apprentissage se déplace.

L'acte l'apprentissage se produit soit au cours du déplacement, soit aux points de départ et/ou à l'arrivée du déplacement.

2) Mobilité du lieu : Le lieu dans lequel se trouve un des acteurs de la formation est mobile, par exemple l'acteur est dans un bus.

3) Mobilité du dispositif : le dispositif dont est équipé un acteur est mobile ou porté.

❖ **Contexte** : Les environnements riches et variés où l'apprentissage se produit.

Localisation : Une situation de l'apprentissage mobile implique dans les cas où il y a mobilité de l'acteur et/ou mobilité du lieu de multiples lieux où sont localisés les acteurs : salle de classe, lieu sur le campus, domicile personnel. Il existe quatre types de localisation : [Meyer, & al 2006]

1) Localisation indépendante du lieu

2) Localisation dépendante du lieu / Co-localisation,

3) Localisation dépendante du lieu / Lieu signifiant pour le contexte d'usage

4) Localisation dépendante du lieu / Lieu signifiant pour l'application.

I.7. L'avenir de l'apprentissage mobile

L'apprentissage mobile aura sa propre identité et ne sera plus seulement considéré comme du e-Learning sur un support mobile. Les concepteurs de dispositifs de l'apprentissage mobile prendront en compte les contraintes techniques liées à la taille de l'écran et à l'ergonomie, ainsi que la nécessité de produire des contenus simples et attractifs.

❖ Des sous-disciplines du m-learning vont émerger :

Des nouveaux besoins d'apprentissages spécifiques vont apparaître comme par exemple des touristes qui visitent un pays et qui pourront recevoir des informations sur les lieux à visiter dans la région.

- Le développement du m-Learning pour tablettes va se démocratiser :

L'avantage par rapport à un Smartphone est de pouvoir développer des contenus plus complexes,

- Le format Flash :

Il sera de plus en plus utilisé par les développeurs de projet m-Learning. Il permet de créer des contenus riches graphiquement, attrayant, interactif et animé, facilement et rapidement.

[MATI Montréal 2011]

I.8. Intérêts et limites

II.10.1. L'intérêt du m-Learning

Son principal intérêt est de constituer un complément aux outils de formation mis en place, nous venons de l'expliquer. Au-delà, nous pouvons lister un certain nombre d'autres points très intéressants :

- Proposer un nouveau type d'offres de formation.
- Pousser l'apprenant à être actif et plus un simple consommateur passif «subissant » l'offre de formation.
- Être au plus près des nouvelles pratiques technologiques des apprenants.
- S'immiscer dans le quotidien des apprenants (via des modules, des applications de m-Learning téléchargés et donc résidents).
- Coûts moindres à la fois pour l'institution et l'apprenant. [Fabien Lienard]

I.10.2. Limites et risques

- ❖ Le m-Learning (et l'e-Learning) ne doit pas :
 - Être perçu et pensé comme la solution.
 - Faire « disparaître » le formateur.
 - Omettre le problème de la propriété (des travaux, des renseignements personnels, des équipements, etc.)

- Etre l'occasion d'envahir l'espace personnel et intime de l'apprenant (à l'image des mises en garde formulées par les spécialistes du m-marketing).
- Se traduire par la transformation des contenus dédiés au e-Learning. Il est une absolue nécessité d'élaborer des applications appropriées aux terminaux et notamment aux Smartphones (la contrainte écran par exemple ne peut pas être solutionnée par un simple processus de Rendering . [Jacob,&Liénard]
- Faire oublier les limites techniques des terminaux (nous évoquions les écrans mais la problématique clavier est encore présent).
- Se satisfaire de contenus qui seraient inappropriés aux activités, aux pratiques.[Fabien Lienard]

I.9. Exemple d'approche d'évaluation

L'approche d'évaluation ODALA propose une méthodologie et des techniques pour le développement d'un système d'évaluation basée sur l'ontologie de domaine d'enseignement Onto-TDM(OntologicalTeaching Domain Modelling).

ODALA met à la disposition des concepteurs d'EIAH en mode apprentissage par l'exercice un moyen d'aborder le développement du système d'évaluation. Nous présentons dans ce qui suit le processus ODALA selon deux points de vue : celui du concepteur et celui de l'utilisateur.

I.9.1. Le processus de conception

Le processus de conception d'une évaluation automatisée (cf. Figure 5) commence par la génération de la base de connaissances du domaine par instanciation de l'ontologie Onto-TDM.

Ensuite, l'étape de développement du diagnostic des erreurs nécessite le développement du module d'analyse de la forme qui utilise la spécification ontologique (en particulier la terminologie dans le cas d'analyse de solutions textuelles).

La notation des apprenants et la mise à jour du module apprenant étant deux modules réutilisables d'un domaine à un autre, ils feront objets d'appels avec, à chaque fois des paramètres adéquats.

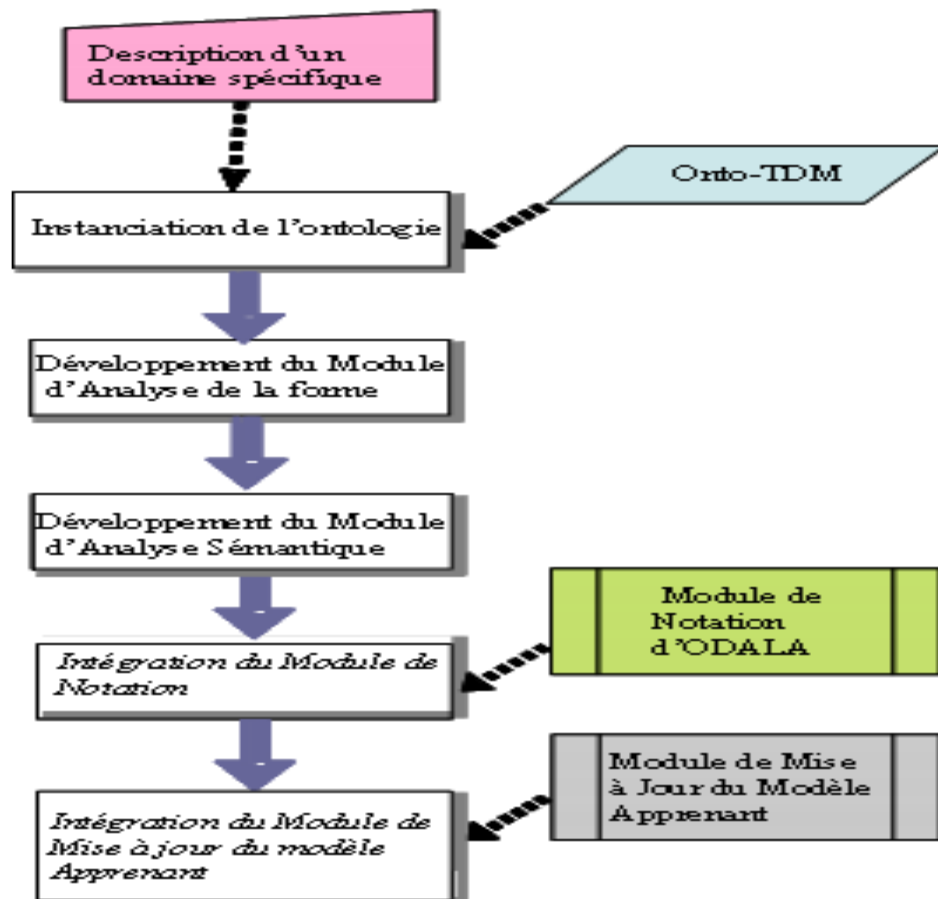


Figure I-5: Principales étapes de développement d'un système d'évaluation avec ODALA. [Bouarab-Dahmani & al. 10b]

I.9.2. Le processus d'utilisation

L'utilisation d'une évaluation conçue selon ODALA nécessite de permettre le recueil, au sein d'un module d'exercice, des données relatives à une session d'apprentissage effectuée par l'apprenant. Ces données devront être par la suite traitées par un module pédagogique, pour synthétiser une appréciation globale intégrant d'autres données sur l'apprenant et son parcours d'apprentissage. Il y a lieu de préciser que nous nous intéressons à un processus d'évaluation principalement destiné à des systèmes d'auto-apprentissage par l'exercice. La session commence donc par une génération automatique d'exercices, telle qu'indiquée dans [Bouarab-Dahmani 00]. Les énoncés d'exercices sont ainsi produits à la demande du pédagogue, en fonction du profil de l'apprenant, soit à partir de la sélection d'exercices dans une banque préenregistrée (génération simple), soit encore à partir d'un ensemble d'exercices paramétrables (génération semi-automatique), ou enfin par concaténation d'énoncés ou des parties d'énoncés déjà existants (génération automatique).

I.10. Conclusion

Nous pouvons déduire de la précédente étude que la conception de domaines d'enseignement ainsi que celles de certaines activités d'apprentissage, basiques pour un apprentissage par l'exercice telle que l'évaluation des connaissances des apprenants est une problématique quand il y a lieu d'utiliser lors de l'apprentissage des questions ouvertes pour lesquelles l'apprenant construit des réponses libres. L'utilisation des ontologies semble donc être une solution de recherche intéressante qui pourra sensiblement contribuer au développement de l'auto-apprentissage avec les EIAH.

L'amélioration des compétences des apprenants c'est le but de tous les acteurs du système éducatif. Les technologies mobiles donnent aux individus la possibilité de choisir quand et où ils veulent apprendre, de façon à répondre aux besoins de mobilité des travailleurs et apprenants. Ainsi, cette étude constitue une importante contribution à la rénovation de l'éducation dans une société de l'information et de la connaissance, soit dans le pays, soit ailleurs.

Chapitre II :

Les Ontologies

II. Les Ontologies

II.1. Introduction

La représentation de la sémantique est nécessaire pour l'intégration des systèmes. Les techniques de représentation des connaissances sont utilisées pour atteindre cet objectif. Ces dernières étudient comment transformer l'expression du sens en une représentation formelle manipulable par une machine. L'ontologie est l'un des moyens les plus utilisés.

De ce fait, durant la dernière décennie, une attention croissante a été portée sur les ontologies. Ces dernières sont largement utilisées de nos jours dans plusieurs domaines comme l'ingénierie de connaissance, les applications liées à la gestion des connaissances, l'e-Learning, la recherche d'information ainsi que le Web Sémantique.

Cet engouement est motivé par le fait que les ontologies peuvent fournir un moyen efficace pour gérer les connaissances partagées et communes à un domaine particulier, tout en permettant des échanges tant au niveau syntaxique que sémantique, entre personnes et/ou systèmes. Les ontologies servent à décrire une sémantique et sont au centre des développements du Web sémantique. Elles permettent la modélisation d'informations agréées par une communauté de personnes et accessibles par une machine pour développer des services automatisés. Elles jouent donc un rôle de référence pour décrire la sémantique des informations des différents sites web.

Dans ce chapitre, nous commençons par donner une brève définition du web sémantique, puis nous relèverons les différentes définitions qui ont été attribuées à la notion d'ontologie, nous verrons aussi les différents éléments dont elle est composée et nous présenterons ses différents types. Ensuite, nous détaillons le processus servant à leurs constructions ainsi que les outils utilisés pour leur développement y compris les langages de spécification.

Finalement, nous montrerons quelques approches d'alignement d'ontologie.

II.2. Web sémantique

D'après Tim Berners-Lee, James Hendler et Ora Lassila, « le Web sémantique est une prolongation du Web actuel avec une signification bien définie des données et des ressources. Il

permet aux ordinateurs et aux utilisateurs de coopérer et d'échanger l'information facilement»[AbdeltifElbyed.ROMIE].

Le Web nous permet aujourd'hui d'accéder à des documents et à des services par l'intermédiaire d'Internet. L'interface d'accès aux services est représentée par des pages Web écrites dans du langage naturel, qui doit être compris par l'être humain .Le Web sémantique est une prolongation du Web actuel dans lequel l'information permet de donner une signification bien définie, afin qu'elles ordinateurs et les personnes puissent travailler et coopérer entre eux efficacement. La vision du Web sémantique a été présentée la première fois par Tim Berners-Lee .L'utilité du Web sémantique peut être vue au travers des exemples suivants : supposons que vous cherchiez à comparer les prix des canapés qui se fabriquent dans votre région (c.-à-d. le même code postal que le votre), ou que vous cherchiez les catalogues en ligne des différents fabricants de pièces de rechange d'une voiture démarque Peugeot 406. Les réponses à ces questions peuvent être disponibles sur le Web, mais pas sous une forme exploitable par la machine. Vous avez toujours besoin d'une personne capable de discerner et de vous donner la signification de ces réponses et de leur importance par rapport à vos besoins.

Le Web sémantique aborde ce problème de deux manières. Il permet d'une part aux communautés d'exposer leurs données de sorte qu'un programme n'ait pas à examiner le format, les images et les annonces d'une page Web pour identifier l'information appropriée. Il permet d'autre part à des personnes d'écrire (de produire) les dossiers qui expliquent - à une machine les rapports entre différents ensembles de données .Par exemple ,on peut faire «un lien sémantique» entre la colonne «code postal» d'une base de données et une forme avec un champ « postal» ou encore « CP» puisqu'ils signifient réellement la même chose. Ceci permet à des machines de suivre des liens et de faciliter l'intégration des données de plusieurs sources d'informations.

Le schéma de la Figure (2.1) montre les couches du standard du Web sémantique .Elle contient les éléments suivants : [AbdeltifElbyed.ROMIE]

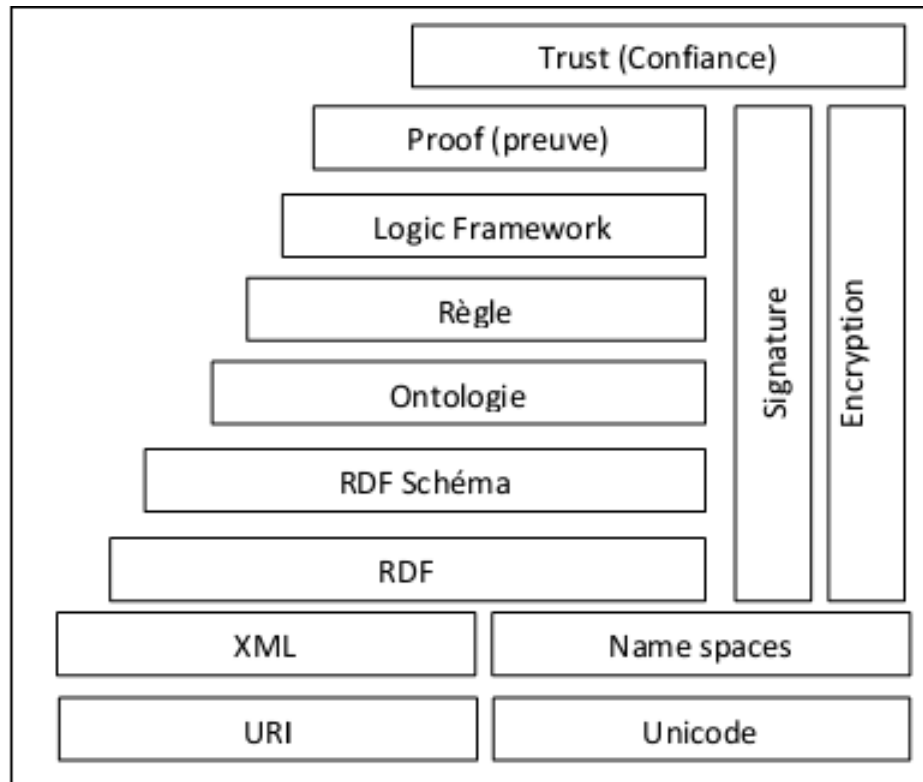


Figure II.2-1: Couches du standard du Web sémantique [Klein, M., 2001]

II.3. Définition d'une ontologie

« Ontologie » est un terme emprunté à la philosophie qui implique une branche de la philosophie qui traite la nature et l'organisation de la réalité » [Abdeltif Elbyed. ROMIE].

Dans le domaine de l'IA, de façon moins ambitieuse, on ne considère que des ontologies, relatives aux différents domaines de connaissances. En fait, plusieurs définitions d'ontologies sont données, mais celle qui caractérise l'essentiel d'une ontologie est fondée sur la définition suivante :

« Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée », avec la signification des termes suivants :

- Formelle : réfère au fait qu'une ontologie doit être compréhensible par la machine, c'est-à-dire que cette dernière doit être capable d'interpréter la sémantique de l'information fournie.
- Explicite : signifie que le type de concepts utilisés et les contraintes sur leur utilisation doivent être explicitement définis.

- Conceptualisation : se réfère à un modèle abstrait de certains phénomènes dans le monde qui identifie les concepts appropriés de ce phénomène.
- Partagée : indique que l'ontologie supporte la connaissance consensuelle, et elle n'est pas restreinte à certains individus mais est acceptée par un groupe.

La Figure (2.2) de la page suivante illustre un exemple d'ontologie

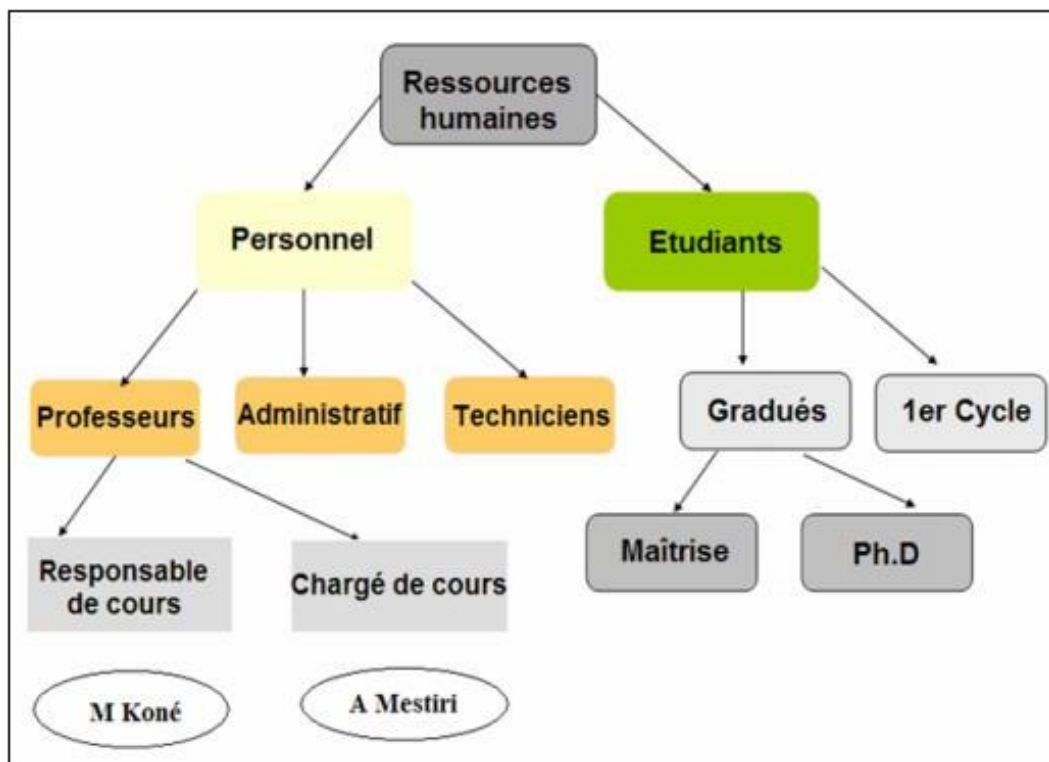


Figure II.2-2 : Exemple d'une ontologie.

II.4. Constituants d'une ontologie

II.4.1. Concepts

Un concept peut représenter un objet matériel, une notion ou encore une idée. Un concept peut être divisé en trois parties : un terme ou plusieurs, une notion et un ensemble d'objets. Les termes sont aussi appelés labels ou étiquettes du concept. La notion est appelée aussi intension du concept. Elle contient la sémantique du concept. L'ensemble des objets représentent l'extension du concept.

Les principales propriétés sur les concepts peuvent être classées en deux groupes : les propriétés sur un seul concept et les propriétés sur deux concepts. Les propriétés portant sur un seul concept sont :

- La généralité : un concept est général s'il n'admet pas d'extension, c.-à-d., d'instances (par exemple "la vérité" est un concept général).
- L'identité : un concept a une propriété d'identité s'il dispose d'une propriété permettant de distinguer une instance d'une autre (par exemple le concept "étudiant" porte une propriété d'identité liée au numéro de l'étudiant, deux étudiants sont identiques s'ils possèdent le même numéro).
- La rigidité : un concept est rigide si toute instance de ce concept en reste instance dans tous les domaines (par exemple le concept "humain" est rigide, par contre le concept "étudiant" est non rigide).
- L'anti-rigidité : un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept (par exemple le concept "étudiant" est un concept anti-rigide car l'étudiant est avant tout un "humain").
- L'unité : un concept est unitaire, pour chacune de ses instances, les différentes parties de l'instance sont reliées par une relation qui ne lie pas d'autres instances de concepts (par exemple les deux parties d'un couteau, manche et lame sont liées par une relation "emmanché" qui ne lie que cette lame et cette manche).

✚ Les propriétés portant sur les deux concepts sont les suivantes :

- L'équivalence : deux concepts sont équivalents s'ils possèdent la même extension.
- La disjonction : deux concepts sont disjoints si leurs extensions sont disjointes (par exemple le concept "homme" et le concept "femme" sont deux concepts disjoints).
- La dépendance : deux concepts sont dépendants si pour toute instance du premier concept il existe une instance du second concept qui ne soit ni partie ni constituant du premier concept (par exemple le concept "parent" est un concept dépendant du concept "enfant" et vice-versa).

II.4.2. Relations

Une relation permet de lier des instances de concepts, ou des concepts génériques. Une relation est caractérisée par un terme (voir plusieurs) et une signature. Les termes sont les labels ou les étiquettes de la relation. La signature précise le nombre d'instances de concepts, que la relation lie, leurs types et l'ordre des concepts (c'est à dire la façon dont la relation doit être lue). Par exemple, la relation "écrit" lie une instance du concept "personne" et une instance du concept "texte", dans cet ordre.

Tout comme les concepts, les relations peuvent être caractérisées par des propriétés. Les relations sont organisées de manière hiérarchisée à l'aide de la propriété de subsomption. Les propriétés des relations sont classées en trois groupes : les propriétés intrinsèques, les propriétés liant deux relations et les propriétés liant une relation et des concepts.

✚ Les propriétés intrinsèques sont les suivantes :

- Les propriétés algébriques : comme la symétrie, la réflexivité et la transitivité.
- La cardinalité : indique le nombre possible de relation de ce type entre les mêmes concepts ou instances du concept.

✚ Les propriétés liant deux relations sont :

- L'incompatibilité : deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts (par exemple les relations "être rouge" et "être vert" sont incompatibles).
- L'inverse : deux relations binaires sont inverses l'une de l'autre si, l'une lie deux instances I1 et I2, l'autre lie I2 et I1 (par exemple les relations "a pour père" et "a pour enfant" sont inverses l'une de l'autre).
- L'exclusivité : deux relations sont exclusives si, l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa (par exemple "l'appartenance" et "la non appartenance" sont deux relations exclusives). L'exclusivité entraîne l'incompatibilité.

- **Lien relationnel**: il existe un lien relationnel entre une relation R et deux concepts $C1$ et $C2$ si pour tout couple d'instances des concepts $C1$ et $C2$, il existe une relation de type R qui lie les deux instances de $C1$ et $C2$. Un lien relationnel peut être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept (par exemple il existe un lien relationnel entre les concepts "texte" et "auteur" d'une part et la relation "a pour auteur" d'autre part).
- **La restriction de relation** : Il existe une restriction de relation entre une relation R et un concept C , si pour tout concept de type C , et toute relation de type R liant C , les autres concepts liés par la relation sont d'un type imposé, (ex., si la relation "mange" portant sur une "personne" et un "aliment" lie une instance de "végétarien", concept subsumé par "personne", l'instance de "aliment" est forcément instance de "végétarien").

II.4.3. Axiomes

Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation [OUSLIM HAMZA].

II.4.4. Instances

Instances ou individus constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances à propos du domaine du problème.

II.5. Types d'ontologie

Les ontologies peuvent être classifiées en fonction de plusieurs dimensions : selon leur formalisation, leur type de conceptualisation, leur propos, leur niveau de complexité, et selon le nombre de points de vue des concepteurs ou d'utilisateurs, etc [Lynda DJAKHDJAKHA].

- ✚ Selon l'état de formalisation: selon leur formalisation, les ontologies peuvent être classées en trois types:

1. Les ontologies semi-informelles : une ontologie est dite semi-informelle, si elle est exprimée dans une forme restreinte et structurée du langage naturel.

2. Les ontologies semi-formelles : une ontologie est dite semi-formelle, si elle est définie dans un langage artificiel et formellement défini.

3. Les ontologies rigoureusement formelles : une ontologie est dite rigoureusement formelle, si elle est définie dans un langage avec une sémantique formelle, des théories et des preuves des propriétés telles que la solidité et la perfection.

- ✚ Selon leur conceptualisation: selon leur nature de conceptualisation, les ontologies peuvent être classées voir la Figure (1.3) [Lynda DJAKHDJAKHA]:

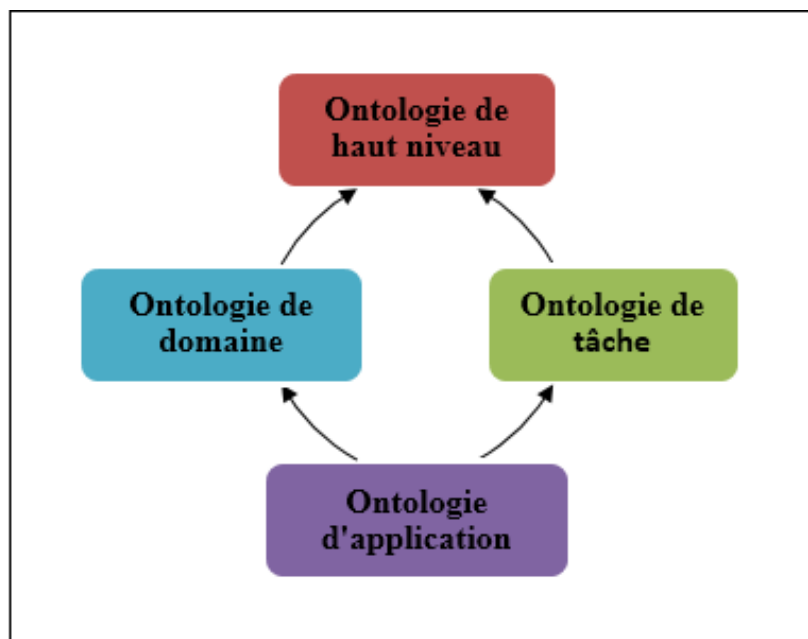


Figure II.2-3 : Types d'ontologie : ce type est classé selon la conceptualisation

- 1. Les ontologies de représentation de connaissances :** les ontologies de représentation des connaissances regroupent les concepts impliqués dans la formalisation des connaissances (classes, instances, propriétés, relations, restrictions, etc).
- 2. Les ontologies de haut niveau :** les ontologies de haut niveau sont des ontologies générales. Elles décrivent des concepts de manière générale, et de haute abstraction, c'est-à-dire indépendamment d'un domaine ou problème particulier, comme : les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, et les propriétés.

3. Les ontologies génériques: une ontologie générique appelée également méta-ontologie véhicule des connaissances génériques moins abstraites que celles véhiculées par l'ontologie de haut niveau, mais assez générales néanmoins pour être réutilisées à travers différents domaines.

4. Les ontologies de domaine : les ontologies du domaine expriment des conceptualisations spécifiques à des domaines particuliers et utilisées par des communautés bien définies. Ces conceptualisations mettent des contraintes sur la structure et les contenus des connaissances du domaine. La plupart des ontologies existantes dans le Web sémantique sont des ontologies de domaine.

5. Les ontologies de tâche: ces ontologies sont utilisées pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration de tutorat, soit tout ce qui concerne la résolution de problèmes.

✚ Selon leur propos : selon leur propos, les ontologies peuvent être classées en :

1. Les ontologies d'application : l'ontologie d'application est la plus spécifique, dont les concepts correspondent souvent aux rôles joués par les entités du domaine tout en exécutant une certaine activité. Elle contient toutes les définitions nécessaires pour décrire la connaissance requise pour une application particulière.

2. Les ontologies de référence : elles sont définies comme des ontologies conçues pour décrire un domaine correctement. Les ontologies de référence sont utilisées durant le processus de développement d'applications pour une compréhension mutuelle entre des agents appartenant à des communautés différentes, pour établir un consensus dans une communauté qui a besoin d'adopter de nouveaux termes, ou tout simplement pour expliquer le sens de termes aux nouveaux arrivants dans la communauté.

✚ Selon le niveau de complexité :

1. Les ontologies légères : sont des structures taxonomiques simples, composées de primitives ou de termes et des définitions qui leur sont associées. Les ontologies légères ne sont pas ou peu axiomatisées, et ne garantissent pas un engagement ontologique. Elles sont réduites à des structures taxonomiques entre des termes considérés comme appropriés.

2. Les ontologies lourdes : ces ontologies ont recours à une axiomatisation riche, à la différence des ontologies légères, les ontologies lourdes représentent explicitement l'engagement

ontologique. Le but de l'axiomatisation est d'exclure des ambiguïtés terminologiques et conceptuelles, due aux interprétations fortuites.

✚ Selon le nombre de points de vue de concepteurs :

1. **Les ontologies inspirationnelles**: les ontologies inspirationnelles sont conçues selon un seul point de vue du concepteur de domaine.
2. **Les ontologies collaboratives** : à la différence des ontologies inspirationnelles, les ontologies collaboratives sont conçues selon multiples points de vue de différents concepteurs et acteurs de domaines.

✚ Selon le nombre de points de vue des futurs utilisateurs: A travers ce que nous avons présenté dans les sections précédentes, il ressort que la notion d'ontologie constitue l'une des approches les plus efficaces pour représenter les connaissances d'un domaine, mais l'existence de plusieurs façons d'appréhender ces connaissances selon différents points de vue donne lieu à la naissance d'une autre notion d'ontologie qui est l'ontologie multipoints de vue. Les ontologies selon le nombre de points de vue de futurs utilisateurs de l'ontologie. Nous distinguons alors:

1. **Les ontologies mono-point de vue** : elles sont conçues selon un point de vue global des futurs utilisateurs.
2. **Les ontologies multipoints de vue** : Une ontologie étant une représentation subjective de la réalité, il semble difficile (voire impossible) d'imaginer que les concepteurs d'ontologies puissent en mettre d'accord sur une représentation commune de la réalité, et que celle-ci soit adaptée aux besoins de tous les utilisateurs

La Figure (2.4) montre un résumé de la classification présentée [Lynda DJAKHDJAKHA].

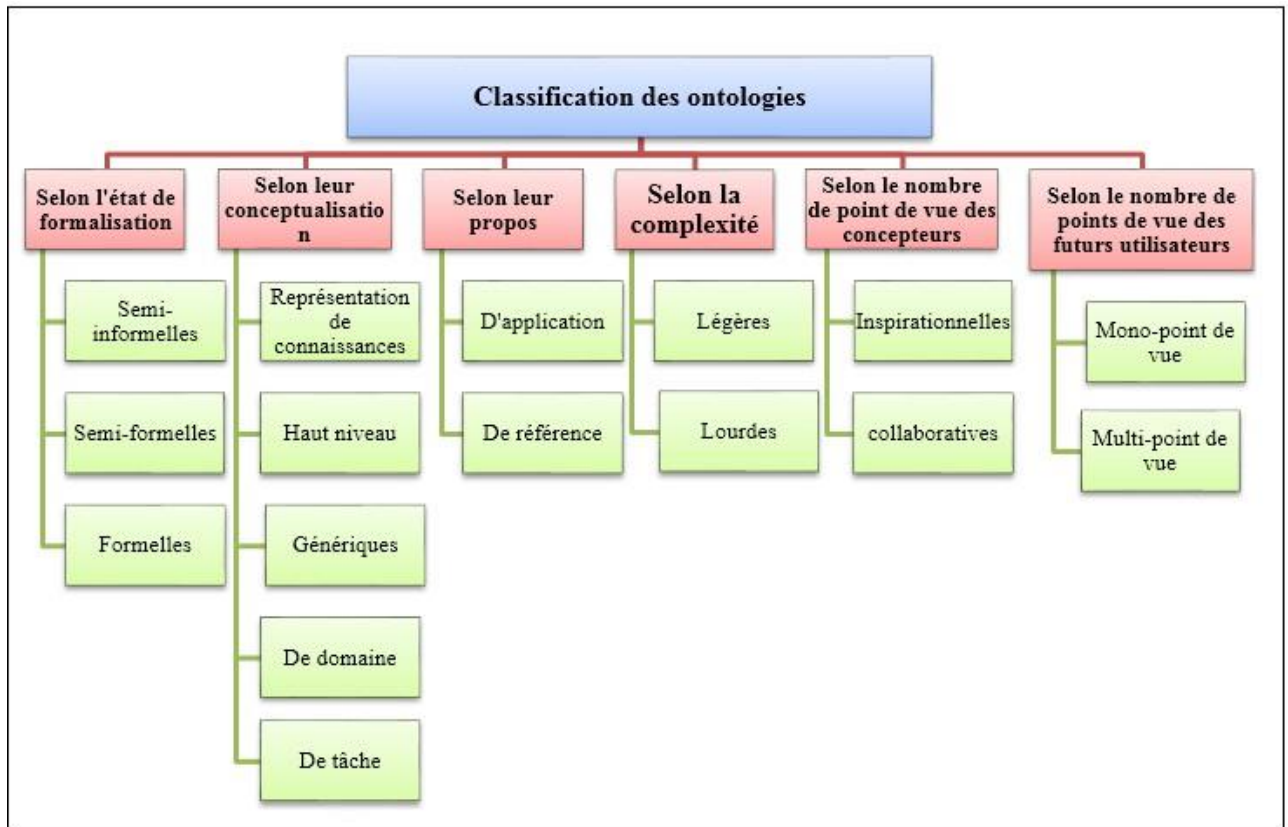


Figure II.2-4 : Classification des ontologies [Natalya F. Noy et Deborah L. McGuinness]

II.6. Développement d'une ontologie

En termes pratiques, développer une ontologie consiste donc à [OUSLIM HAMZA] :

- Définir les classes dans l'ontologie.
- Arranger les classes en une hiérarchie taxonomique (sous-classe / super-classe).
- Définir les attributs et décrire les valeurs autorisées pour ces attributs.
- Renseigner les valeurs pour les attributs des instances.

Le développement d'une ontologie est nécessairement un processus itératif.

Les concepts dans une ontologie doivent être très proches des objets (physiques ou logiques) et des relations dans un domaine d'intérêt. Fort probablement ils sont des noms (objets)

ou verbes (relations) dans des phrases qui décrivent un domaine. Il faut également se rappeler qu'une ontologie est un modèle de la « réalité » du monde et que les concepts dans l'ontologie doivent refléter cette réalité. Nous donnons ci-dessous les différentes étapes à suivre pour le développement d'une ontologie :

Etape 1 :

Nous suggérons de commencer le développement d'une ontologie en définissant son domaine et sa portée. C'est à dire en répondant à quelques questions de base :

- Quel est le domaine que va couvrir l'ontologie ?
- Dans quel but utiliserons nous l'ontologie ?
- A quels types de questions l'ontologie devra-t-elle fournir des réponses ?
- Qui va utiliser et maintenir l'ontologie ?

Les réponses à ces questions peuvent varier au cours du processus de la conception de l'ontologie, mais à chaque moment donné, elles aident à limiter la portée du modèle.

Etape 2 : Envisager une éventuelle réutilisation des ontologies existantes :

Il est toujours utile de prendre en considération ce que d'autres personnes ont fait et d'examiner si nous pouvons élargir des sources existantes et les affiner pour répondre aux besoins de notre domaine ou de notre tâche particulière. Plusieurs ontologies sont déjà disponibles sous forme électronique et peuvent être importées dans votre propre environnement, si cet environnement permet le développement des ontologies.

Il existe des bibliothèques d'ontologies réutilisables sur le Web et dans la littérature. Nous pouvons citer à titre d'exemple, les bibliothèques :

- Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>) ou bien
- DAML (<http://www.daml.org/ontologies/>).

Etape 3 : Enumérer les termes importants dans l'ontologie

Il est utile de noter sous forme de liste tous les termes à traiter ou à expliquer à un utilisateur.

Sur quels termes souhaiterions-nous discuter ? Quels sont les propriétés des termes ? Que veut-on dire sur ces termes ? Il est important d'établir une liste exhaustive de termes et de ne pas se soucier de l'éventuelle chevauchement entre les concepts qu'ils représentent, les relations entre les termes ou toute autre propriété des concepts, ni si ces concepts sont des classes ou des facettes [Natalya F. Noy et Deborah L. McGuinness].

✚ **Étape 4** : Définir les classes et la hiérarchie des classes

Il existe un certain nombre d'approches possibles pour développer une hiérarchie de classes :

- Un procédé de développement de haut en bas commence par une définition des concepts les plus généraux du domaine et se poursuit par la spécialisation des concepts.
- Un procédé de développement de bas en haut commence par la définition des classes les plus spécifiques, les feuilles d'une hiérarchie, et se poursuit avec le regroupement de ces classes en concepts plus généraux.
- Une procédé combiné de développement est une combinaison des deux approches, de haut en bas et de bas en haut. Au tout début, les concepts les plus saillants sont définis, ensuite ils sont généralisés ou spécialisés, suivant le cas.

✚ **Étape 5** : Définir les propriétés des classes - attributs

Les classes seules ne fourniront pas assez d'information pour répondre aux questions de compétence de l'étape 1. Après avoir défini quelques classes, nous devons décrire la structure interne des concepts [I. Bouazza].

- Les deux étapes précédentes - développement de la hiérarchie des classes et définition des propriétés des concepts (attributs) - sont étroitement entrelacées. Il est difficile de le faire en ordre linéaire strict. D'habitude, nous créons quelques définitions de concepts dans la hiérarchie, ensuite nous procédons à la description des propriétés de ces concepts et ainsi de suite.

✚ Étape 6 : Définir les facettes des attributs

Les attributs peuvent avoir plusieurs facettes décrivant la valeur du type, les valeurs autorisées, le nombre de valeurs (cardinalité), et d'autres caractéristiques de valeurs que les attributs peuvent avoir [Natalya F. Noy et Deborah L. McGuinness].

✚ Étape 7 : Créer les instances :

La dernière étape consiste à créer les instances des classes dans la hiérarchie. Définir une instance individuelle d'une classe exige (1) choisir une classe, (2) créer une instance individuelle de cette classe, et (3) la renseigner avec les valeurs des attributs [Natalya F. Noy et Deborah L. McGuinness].

II.7. Langages de représentations

Dans cette section, nous définissons quelques langages de représentation des ontologies les plus connus et les plus utilisés [I. Bouazza].

II.7.1. XML

L'Extended Markup Language (XML) est un langage de description, issu de SGML (Standard Generalized Markup Language) et défini par le consortium Web W3C. XML permet de décrire la structure arborescente de documents à l'aide d'un système de balises permettant de marquer les éléments qui composent la structure et les relations entre ces éléments. XML ne pose aucune contrainte sémantique sur la description des informations, il ne constitue donc pas un langage de modélisation d'ontologies à lui seul. Par exemple, avec XML nous pouvons décrire une Personne Mohammedage de 20 ans qui habite à la rue Lotfi, Oran. Pour un agent l'exemple ci-dessous correspondant est compris comme une simple arborescence de chaînes de caractères.

```
<Personne id=' Mohammed'>
  <Adresse> Lotfi rue, Oran
</Adresse>
  <Age>24</Age>
</Personne>
```

II.7.2. KIF

KIF est un langage basé sur les prédicats du premier ordre avec des extensions pour représenter des définitions et des méta-connaissances, la logique du premier ordre étant un langage de bas niveau pour l'expression d'ontologies. Une extension du langage KIF, ON-TOLINGUA, est utilisée dans les serveur d'édition d'ontologies, Ontolingua1 du même nom [Abdeltif Elbyed.ROMIE].

II.7.3. RDF

Le Resource Description Framework (RDF) est un modèle pour la représentation de métadonnées à propos de ressources. Cette représentation est faite sous la forme d'un triplet:

- **Sujet** : La ressource que l'on définit
- **Prédicat** : la propriété de la ressource, qui est une liaison étiquetée et orientée du sujet vers l'objet.
- **Objet**: La valeur de la propriété pouvant être une autre ressource ou bien un littéral.

RDF est à la base exprimé sous la forme d'un graphe orienté, mais on peut le d'écrire par la syntaxe XML. On parle alors de RDF/XML qui par abus de langage peut être appelé RDF. En prenant l'exemple défini pour XML, nous pouvons définir avec RDF un graphe dont l'agent comprendra comme un concept Mohammed qui possède les propriétés adresse et age.

```

<rdf :RDF>
  <rdf :Description about='Mohammed'>
    <rdf : Property about='adresse'>
      Lotfi rue, Oran
    </rdf : Property>
    <rdf : Property about='age'>
      24
    </rdf : Property>
  </rdf : Description>
</rdf : RDF>

```

II.7.4. DAML +OIL

Dans l'optique d'une utilisation d'ontologies sur le Web, le langage RDF-S a été enrichi par l'apport du langage OIL (Ontology Interchange Language 1) qui permet d'exprimer une sémantique à travers le modèle des frames tout en utilisant la syntaxe de RDF-S. OIL offre de nouvelles primitives permettant de définir des classes à l'aide de mécanismes ensemblistes issus des logiques de description (intersection de classes, union de classes, complémentaire d'une classe). Il permet également d'affiner les propriétés de RDF-S en contraignant la cardinalité ou en restreignant la portée. Le langage OIL a été fusionné avec le langage DAML pour former le DAML+OIL. DAML (Darpa Agent Markup Language 1) est conçu pour permettre l'expression d'ontologies dans une extension du langage RDF. Il offre les primitives usuelles d'une représentation à base de frames et utilise la syntaxe RDF. L'intégration de OIL rend possible les inférences compatibles avec les logiques de description, essentiellement les calculs de liens de subsumption.

II.7.5. OWL

Ontology Web Language (OWL) est un langage fondé sur la syntaxe RDF/XML et hérité des travaux de DAML+OIL. OWL n'est pas simplement une extension de RDF, il introduit l'aspect sémantique lui manquant, comme les outils de comparaison de propriétés et de classes (identité, équivalence, contraire, cardinalité, symétrie, etc.). Ainsi par ses primitives plus riches, OWL offre une capacité d'interprétation plus grande à la machine que RDF et RDFS. OWL se décompose en trois sous-langages OWL Lite, OWL DL et OWL Full, offrant des capacités d'expression croissantes et donc destinés à des différentes utilisations. OWL Lite est le sous langage d'OWL le plus simple.

II.8. Outils de développement des ontologies

Nous décrivons ci-dessous quelques outils et plateformes pour le développement des ontologies [Abdeltif Elbyed. ROMIE] :

II.8.1. Ontolingua

Ontolingua est un serveur d'édition d'ontologies. Il utilise des classes, des relations, des fonctions, des instances et des axiomes pour décrire une ontologie. Une relation peut contenir des propriétés nécessaires (contraintes) ou nécessaires et suffisantes qui définissent la

relation. En plus le serveur Ontolingua offre la possibilité d'intégrer les ontologies Ontolingua, ce qui permet la construction modulaire des ontologies [Abdeltif Elbyed.ROMIE].

II.8.2. OntoSaurus

OntoSaurus de l'Information Science Institute de l'Université de Southern California est composé de deux modules : un serveur utilisant LOOM comme langage de représentation des connaissances, et un serveur de navigation créant dynamiquement des pages HTML qui affichent la hiérarchie de l'ontologie ; le serveur utilise des formulaires HTML pour permettre à l'utilisateur d'éditer l'ontologie [Abdeltif Elbyed.ROMIE].

II.8.3. WebOnto

WebOnto du Knowledge Media Institute de l'Open University, est une application Web pour naviguer et développer collaborativement les ontologies. Il supporte la navigation collaborative, la création et l'édition d'ontologies sur le Web. Les ontologies WebOnto sont implémentées dans le langage OCML. WebOnto distingue quatre types d'ontologies : ontologie de domaine, ontologie de tâche, ontologie de méthode, et ontologie d'application [Abdeltif Elbyed.ROMIE].

II.8.4. OilEd

OilEd (Oil Editor) est un éditeur d'ontologies utilisant le formalisme OIL. Il est essentiellement dédié à la construction de petites ontologies dont on peut ensuite tester la cohérence à l'aide de FACT, un moteur d'inférences bâti sur OIL [Abdeltif Elbyed.ROMIE].

II.8.5. Protégé2000

PROTEGE-2000 est un environnement graphique de développement d'ontologies développé par le SMI (Stanford Medical Informatics) de Stanford. Dans le modèle des connaissances de PROTEGE, les ontologies consistent en une hiérarchie de classes qui ont des attributs (slots), qui peuvent eux-mêmes avoir certaines propriétés (facets). L'édition des listes de ces objets se fait par l'intermédiaire de l'interface graphique, sans avoir besoin d'exprimer ce que l'on a spécifié dans un langage formel : il suffit juste de remplir les différents formulaires correspondant à ce que l'on veut spécifier. Ce modèle autorise d'ailleurs une liberté de conception assez importante puisque le contenu des formulaires à remplir peut être modifié

suivant les besoins via un système de méta classes, qui constituent des sortes de « patrons » de connaissance. L'interface, très bien conçue, et l'architecture logicielle permettant l'insertion de plugins pouvant apporter de nouvelles fonctionnalités ont participé au succès de PROTÉGÉ-2000. Aujourd'hui, il regroupe une large communauté d'utilisateurs et bénéficie de toutes dernières avancées en matière de recherche ontologique : compatibilité OWL de référence, services inférentiels, gestion de bases de connaissances, visualisation d'ontologies, alignement et fusion, etc [I. Bouazza].

II.8.6. ODE et WebOde

L'outil ODE (Ontology Design Environment) permet de construire des ontologies au niveau connaissance, comme le préconise la méthodologie METHONTOLOGY. L'utilisateur construit son ontologie dans un modèle de type frame, en spécifiant les concepts du domaine, les termes associés, les attributs et leurs valeurs, les relations de subsumption [Abdeltif Elbyed. ROMIE].

II.8.7. OntoEdit

OntoEdit (Ontology Editor) est également un environnement de construction d'ontologies indépendant de tout formalisme. Il permet l'édition des hiérarchies de concepts et de relations et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la genericité d'un concept. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement. OntoEdit intègre un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs. Un contrôle de la cohérence de l'ontologie est assuré à travers la gestion des ordres d'édition [Abdeltif Elbyed. ROMIE].

II.9. Alignement Des Ontologies

Dans le contexte du Web sémantique, l'interopérabilité sémantique basée sur les ontologies est devenue un challenge important. Aujourd'hui, plusieurs stratégies sont proposées pour améliorer cette interopérabilité et résoudre les problèmes d'hétérogénéité ontologiques, telle que les stratégies d'alignement, d'intégration, de fusion, d'articulation et celle de morphisme. Le choix d'utilisation d'une telle stratégie est motivé ou bien guidé par le contexte de l'application et par le but visé par cette application. Dans ce travail, nous nous intéressons au problème d'alignement, puisque nous voulons partager les connaissances sémantiques,

réutiliser les ontologies. Un alignement entre deux ontologies permet de produire des correspondances entre entités de différentes ontologies, et permet aux participants qui les utilisent de communiquer dans le sens où ils peuvent échanger des requêtes exprimées avec leurs connaissances respectives.

Nous focalisons dans cette partie sur les notions liées à l'alignement d'ontologie

II.9.1. Définition d'alignement

✚ Définition 1

L'alignement est défini comme une fonction $\text{map}: O \rightarrow O_t$ tel que : $\text{map}(e_i) = e_t$ si

$\text{sim}(e_i, e_t) > t$. Où O et O' sont les deux ontologies à aligner, t désigne un seuil minimal de similarité appartenant à l'intervalle $[0,1]$, il indique le niveau minimum pour que deux entités soient similaires, e_i et e_t appartiennent respectivement à O et O' , et représentent les entités au niveau des deux ontologies [Lynda DJAKHDJAKHA].

✚ Définition 2

«L'alignement de structures est le processus de mise en correspondance sémantique des entités qui les composent. » [Lynda DJAKHDJAKHA].

Ces structures peuvent être des ontologies, des schémas XML ou des bases de données. Les liens sémantiques comprennent les relations : d'équivalence (=), de généralisation/spécialisation (\pm, \pm) de chevauchement (\in) ou encore d'incompatibilité (\perp). L'évaluation de la véracité

de ces liens peut être booléenne ou par le biais d'autres mesures telles que : les probabilités, les mesures symboliques, les mesures de similarité. La Figure (2.1) donne un exemple d'alignement [6].

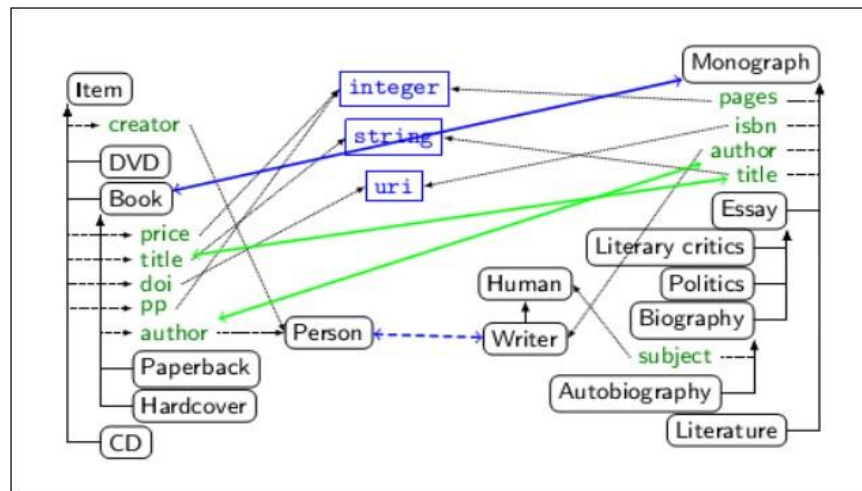


Figure II.2-5 : Exemple d'alignement entre deux ontologies[Euzenat, J., Scharffe, F. & Zimmermann 2007]

II.9.2. Dimensions de l'alignement

L'alignement regroupe trois dimensions : l'input, le processus d'alignement et l'output[7].

✚ Input

Est constitué essentiellement des structures destinées à être alignées et qui peuvent être, comme énoncé précédemment, des schémas XML, des schémas relationnels, des ontologies [7].

✚ Processus d'alignement

Comme le montre la Figure (2.2) le processus d'alignement peut être considéré comme une fonction f , qui à partir d'une paire d'ontologies O et O' , un alignement en entrée A (optionnel), un ensemble de paramètres P (ex : paramètres de pondération, seuils, ... etc) et un ensemble de ressources externes r (ex: thésaurus, lexique, etc.), détermine un alignement A' entre ces deux ontologies: $A' = f(O, O', A, p, r)$. Ceci peut être représenté schématiquement de la manière suivante [7]:

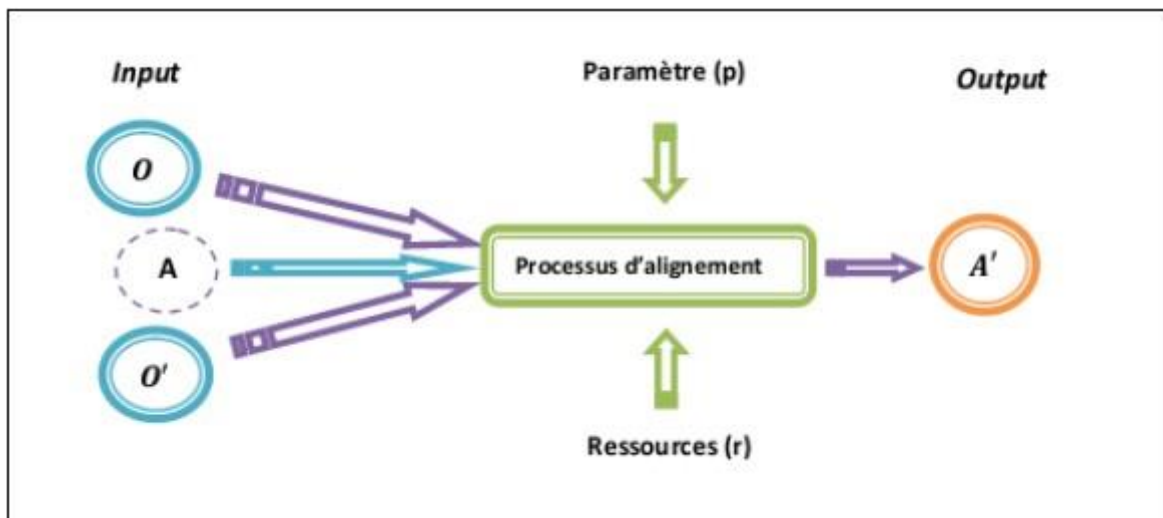


Figure II.2-6 : **Processus d'alignement** [Euzenat, J., Scharffe, F. & Zimmermann 2007]

✚ Output

Est un ensemble de correspondances sémantiques reliant les entités qui composent les deux ontologies. Une correspondance est décrite comme un ensemble de cinq éléments :

$\langle id, e, e', r, n \rangle$ tel que [7] : Où :

- id : identifiant unique d'un mapping.
- e : une entité, à aligner, appartenant à O (classe, propriété, contrainte, instance).
- e' : une entité, à aligner, appartenant à O' .
- r : la relation qui relie e à e' ($=, \supseteq, \subseteq, \in, \perp$).
- n : la mesure de confiance de la relation, généralement une valeur réelle comprise dans l'intervalle $[0,1]$. Plus n est proche de 1, plus la relation est considérée comme étant forte.

II.9.3. Etapes du processus d'alignement

Dans la plupart des systèmes d'alignement d'ontologies, le processus d'alignement est un processus itératif, qui se compose des étapes suivantes (Figure (2.7) [Lynda DJAKHDJAKHA]) :

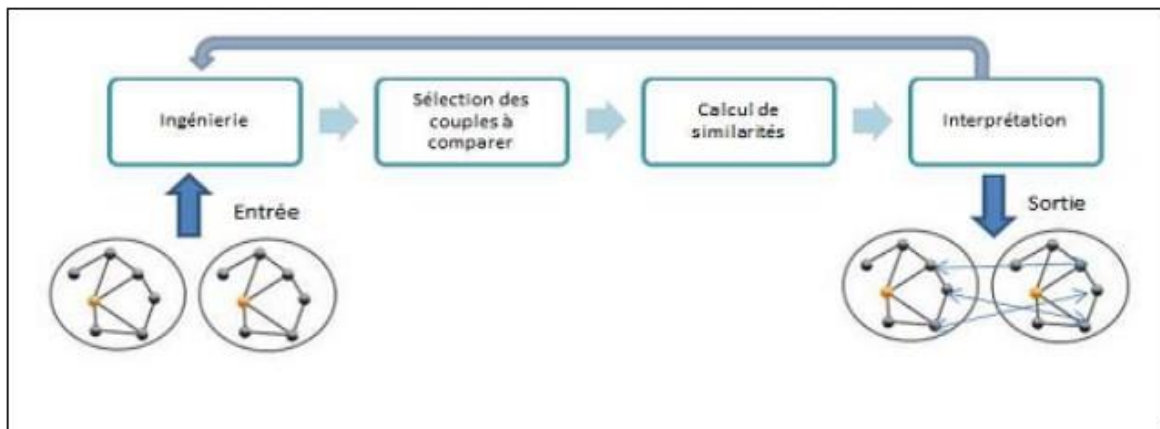


Figure II.2-7 : Etapes du processus d'alignement [Lynda DJAKHDJAKHA]

- L'étape d'ingénierie consiste à identifier les ontologies en entrée et à extraire les entités à aligner. Ces entités dépendent des méthodes d'alignements qui seront ensuite mises en oeuvre.
- L'étape de sélection des couples à comparer consiste à définir les couples d'entités qui seront considérés dans le processus d'alignement. La plupart des systèmes d'alignement existants comparent toutes les entités de la première ontologie avec toutes les entités de la deuxième ontologie.
- L'étape de calcul de similarité est l'étape la plus importante dans le processus d'alignement, comme l'indique son nom, elle calcule une similarité pour chaque couple d'éléments sélectionné dans l'étape précédente.
- Enfin, l'étape finale consiste à interpréter les résultats obtenus et à déduire des alignements à partir des mesures de similarités préalablement calculées.

II.9.4. Similarité

La notion de similarité sémantique est vue comme celle de la similarité topologique en mathématiques, où on l'associe à une fonction, appelée fonction de la similarité. La définition de cette fonction de la similarité peut changer selon les approches, selon les propriétés souhaitées. La valeur de cette fonction est souvent comprise entre 0 et 1, ce qui permet des

possibilités d'interprétation probabiliste de la similarité. Des propriétés ou des caractéristiques communes possibles de la fonction sont des caractéristiques positives, autosimilaires ou maximales, symétriques ou réflexives.

✚ Définition 1 (Similarité)

La similarité $S : O * O \rightarrow R$ est une fonction d'une paire d'entités à un nombre réel exprimant la similarité entre ces deux entités telle que [Lynda DJAKHDJAKHA] :

- $\forall a, b \in O, S(a, b) \geq 0$ (positivité)
- $\forall a, b, c \in O, S(a, a) \geq S(b, c)$ et $S(a, a) = S(a, b) \leftrightarrow a = b$ (autosimilarité ou maximalité)
- $\forall a, b \in O, S(a, b) = S(b, a)$ (symétrie)
- $\forall a, b, c \in O, S(a, b) = S(b, c) \rightarrow S(a, b) = S(a, c)$ (transitivité)
- $\forall a, b \in O, S(a, b) \leq \infty$ (finitude)

✚ Définition 2 (Dissimilarité)

La dissimilarité $DS : O * O \rightarrow R$ est une fonction d'une paire d'entités à un nombre réel exprimant la dissimilarité entre ces deux entités telle que :

- $\forall a, b \in O, DS(a, b) \geq 0$ (positivité)
- $\forall a, b, c \in O, DS(a, a) \leq DS(b, c)$ et $DS(a, a) = 0$ (minimalité)
- $\forall a, b \in O, DS(a, b) = DS(b, a)$ (symétrie)
- $\forall a, b \in O, DS(a, b) \leq \infty$ (finitude)

✚ Définition 3 (Distance)

La distance $D : O * O \rightarrow R$ est une fonction de la dissimilarité satisfaisant la définitivité et l'inégalité triangulaire :

- $\forall a, b \in O, D(a, b) = 0 \leftrightarrow a = b$ (définitivité)
- $\forall a, b, c \in O, D(a, b) + D(b, c) \geq D(a, c)$ (inégalité triangulaire)

✚ Définition 4 (Normalisation)

Une mesure est une mesure normalisée si les valeurs calculées par cette mesure ne peuvent varier que dans un intervalle de 0 à 1. Ces valeurs calculées sont appelées valeurs normalisées. Les fonctions du calcul sont appelées fonctions normalisées et notées f^- .

❖ Quelques techniques de similarité et de distance

On trouve plusieurs techniques pour calculer la similarité et la distance, nous citons quelques une :

✚ Définition 5 (Similarité de Jaccard)

Soit s et t deux chaînes de caractères. Soit S et T les ensembles des caractères de s et t respectivement. La similarité de Jaccard est une fonction de la similarité S (jaccard):

$S * S \rightarrow [0,1]$ telle que :

$$\overline{S}_{jaccard}(s, t) = \frac{|S \cap T|}{|S \cup T|}$$

✚ Définition 6 (Similarité hybride)

Soit $s = a_1 \dots a_k$ et $t = b_1 \dots b_L$ deux chaînes de caractères, où a_j et b_j sont des sous-chaînes de s et t respectivement. Soit S une mesure de la similarité entre deux chaînes de caractères.

La similarité hybride est une fonction de la similarité S hybride:

$S * S \rightarrow [0,1]$ telle que:

$$\overline{S}_{hybride}(s, t) = \frac{1}{K} \sum_{i=1}^K \max_{j=1}^L S(a_i, b_j)$$

✚ Définition 7 (Distance d'édition)

Soit un ensemble d'opérations d'édition OP , $op \in P, op: S \rightarrow S$ et une fonction de coût d'édition $w: OP \rightarrow \mathbb{R}$. Pour n'importe quelle paire des chaînes de caractères, il existe une séquence des opérations d'édition qui transforme la première en la seconde (et vice versa), la distance d'édition de deux chaînes de caractères s et t est une fonction de la dissimilarité DS_{de} :

$S * S \rightarrow [0,1]$ telle que

$DS_{de}(s,t)$ est le coût de la séquence des opérations la moins coûteuse qui transforme s en t .

$$\overline{DS}_{de}(s,t) = \min_{(op_1)_1:op_n(\dots op_1(s))=t} (\sum_{k=1}^n w_{op_k})$$

✚ Définition 8 (Distance de Jaro)

Soit s et t deux chaînes de caractères. Soit N_c le nombre des caractères communs apparaissant dans les deux chaînes dans une distance de moitié de la longueur de la chaîne la plus courte. Soit N_t le nombre des caractères transposés, qui sont des caractères communs apparaissant dans des positions différentes. La distance de Jaro est une fonction de la dissimilarité DS_{jaro} :

$S * S \rightarrow [0,1]$ telle que:

$$DS_{jaro}(s,t) = 1 - \frac{1}{3} \left(\frac{N_c}{|s|} + \frac{N_c}{|t|} + \frac{N_c - N_t/2}{N_c} \right)$$

II.10. Méthodes d'alignement

La mise en correspondances des entités d'ontologies est faite par des méthodes d'alignement. Ces méthodes permettent de comparer un élément d'une ontologie avec un autre

élément d'une autre ontologie. La classification des méthodes présentée est fondée sur les états de l'art présentés dans (Bach 2006) et (Eufémia 2007), qui donnent les catégories suivantes :

II.10.1. Méthodes terminologiques

Les méthodes terminologiques peuvent être appliquées pour comparer des chaînes de caractères, des termes ou des textes et employées pour calculer la valeur de similarité des entités textuelles, telles que des noms, des étiquettes, des commentaires, ... etc. Elles peuvent être [Lynda DJAKHDJAKHA]:

- Des méthodes lexicales se basent exclusivement sur les éléments à comparer et exploitent la structure de la chaîne de caractères en tant que séquence de caractères, l'ordre des caractères dans la chaîne, le nombre d'apparitions d'une lettre dans une chaîne pour concevoir des mesures de similarité, mais ces méthodes n'exploitent pas la signification des termes.
- Des méthodes linguistiques s'appuient sur des techniques de la langue naturelle et exploitent les propriétés de ses caractéristiques expressives et productives [Lynda DJAKHDJAKHA].

II.10.2. Méthodes structurelles

Les méthodes structurelles exploitent l'organisation des ressources sémantiques et les informations structurelles pour déduire la similarité de deux entités, lorsque les entités en question sont reliées entre elles par des liens sémantiques ou syntaxiques, formant une hiérarchie ou un graphe des entités. Les méthodes structurelles peuvent n'exploiter que les structures internes des entités (par exemple les informations relatives aux propriétés qui prennent leurs valeurs dans un type de données particulier) et dites méthodes structurelles internes, comme elles peuvent tirer parties des relations entre des entités (par exemple les relations d'héritage) et dites dans ce cas méthodes structurelles externes [Lynda DJAKHDJAKHA].

II.10.3. Méthodes extensionnelles

Les méthodes extensionnelles calculent la similarité entre deux entités (concepts/classes) en exploitant leurs extensions (i.e. instances). Ces méthodes emploient en général des techniques de fouille de données, des méthodes d'analyse ou des techniques d'apprentissage automatiques pour calculer des mesures de similarité [Lynda DJAKHDJAKHA].

II.10.4. Méthodes sémantiques

Les méthodes d'alignement sémantique exploitent l'interprétation sémantique formelle dans le traitement de l'information pour calculer la similarité entre les concepts de deux ontologies. Elles se basent généralement sur des modèles de logique, tels que les modèles de satisfiabilité propositionnelle (SAT), sur des méthodes de déduction et sur le raisonnement fondé sur les logiques de description (LD) [Lynda DJAKHDJAKHA].

II.10.5. Méthodes fondées sur la satisfiabilité propositionnelle

Les méthodes fondées sur la satisfiabilité propositionnelle emploient des techniques de satisfiabilité propositionnelles (SAT) pour vérifier la validité d'un ensemble de formules propositionnelles qui est construit en traduisant des relations déjà connues et des relations à vérifier entre des entités vers des formules propositionnelles. Une formule propositionnelle est valide si et seulement si sa négation est insatisfiable [Lynda DJAKHDJAKHA].

II.10.6. Méthodes basées sur les logiques de description

Les méthodes fondées sur les logiques de description utilisant un moteur d'inférences et l'algorithme des tableaux sémantiques pour calculer la similarité entre les ontologies décrites en logiques de description. Elles peuvent vérifier des relations sémantiques entre les entités en question pour déduire la similarité entre ces entités, telles que l'équivalence (la similarité est égale à 1), la subsumption (la similarité est de 0 à 1) ou l'exclusion (la similarité est égale à 0) [Lynda DJAKHDJAKHA].

II.10.7. Méthodes basées sur les langages de représentation d'ontologies

Les méthodes fondées sur les langages de représentation d'ontologies exploitent la syntaxe du langage utilisé pour calculer la similarité. Le processus d'alignement peut, par exemple, n'exploiter qu'un sous-ensemble des primitives OWL [Lynda DJAKHDJAKHA].

II.11. Outils de l'alignement des ontologies

Différents outils ont été développés dans le but d'aligner plusieurs ontologies. [8]

1. **PROMPT** est un système interactif constituant une aide pour la comparaison, l'alignement, la fusion et l'évolution d'ontologies. Son module d'alignement appelé Anchor-

Prompt permet de rapprocher des ontologies de la façon suivante : d'abord, des "matchers" terminologiques permettent de déterminer un ensemble initial de concepts similaires. À partir de cette liste, un algorithme analyse les chemins dans les sous-graphes délimités par ces concepts et détermine quelles classes apparaissent fréquemment dans les mêmes positions sur des chemins similaires. Cette analyse permet de guider l'utilisateur pour choisir les meilleurs mappings.

2. OLA (OWL Lite Alignment) est un système implémentant un algorithme d'alignement des ontologies décrites en OWL. OLA mesure la similarité entre deux entités à partir des calculs de similarité entre leurs caractéristiques (leurs types : classe, relation ou instance, leurs liens avec d'autres entités : sous-classes, domaine, ...). La valeur de similarité finale est la somme pondérée des valeurs de similarité de chaque caractéristique.

3. AROMA (Association Rule Ontology Matching Approach) est une approche d'alignement pour des ontologies représentées en OWL. Elle permet de découvrir des liens sémantiques de type « subsumption » ou « équivalence » entre deux entités (classes ou propriétés). Le processus d'alignement se déroule en trois étapes : la première procède à l'acquisition des termes contenus dans les descriptions et instances des entités à partir d'outils de Traitement Automatique du Langage (TAL). Ensuite, pour chaque entité, ainsi qu'à ses ancêtres, est associé un ensemble de termes représentatifs. La deuxième étape permet de créer des relations de subsumption entre les entités à partir de règles d'association. Des "matchers" terminologiques sont utilisés pour comparer les différentes descriptions. Enfin, la dernière étape vise à analyser les règles d'associations trouvées afin de : (1) déduire des relations d'équivalence, (2) éliminer les incohérences (cycles) (3), supprimer les relations redondantes, (4) sélectionner le meilleur alignement pour chaque entité.

4. ASMOV (Automated Semantic Mapping of Ontologies with Validation) est un système d'alignement d'ontologies conçu pour faire coopérer des ontologies issues de sources de données hétérogènes. ASMOV permet de produire des mappings entre des concepts et/ou des propriétés et/ou des instances de deux ontologies. L'algorithme implémenté est automatique, il calcule de façon itérative, la similarité entre deux entités appartenant à deux ontologies différentes suivant quatre caractéristiques (les éléments lexicaux, les relations structurelles, la structure interne, les instances de classes et valeurs des propriétés). La similarité finale est calculée à partir de la somme pondérée des quatre mesures et permet

d'obtenir un alignement. Le système vérifie ensuite cet alignement afin de s'assurer qu'il ne contient pas d'incohérence sémantique.

II.12. Frameworks d'alignement

Plus récemment, les Frameworks sont apparus dans les systèmes d'alignements d'ontologies. Leur avantage est qu'ils permettent de multiples combinaisons de stratégies de calcul de la similarité. Par exemple [8]:

- **COMA++ (COmbiningMAtching)** est un système générique de mise en correspondance de schémas (XML, Schémas relationnels). L'outil permet l'importation, le stockage, l'édition de schémas ainsi que leurs alignements et ce, afin de les transformer ou de les fusionner. Il fournit une bibliothèque extensible d'algorithmes de mappings, un module pour combiner les résultats obtenus et une plateforme pour l'évaluation des différentes mesures. L'utilisateur peut interagir dans le processus de mise en correspondance en sélectionnant le mode de combinaison de matchers. COMA++ est une évolution de COMA qui améliore ses algorithmes et son interface graphique.
- **MAFRA (Mapping Framework for distributed ontologies)** est un Framework interactif, dynamique et progressif pour l'alignement d'ontologies distribuées dans le cadre du Web sémantique. L'approche de MAFRA se déroule suivant deux dimensions : la dimension horizontale définit les étapes du processus d'alignement : importation et normalisation des ontologies à aligner, calcul de similarités entre les éléments des différentes ontologies à partir d'une combinaison de plusieurs mesures de similarités, formalisation des mappings en établissant des « ponts sémantiques » entre les entités des différentes ontologies, exécution des mappings pour transformer les instances d'une ontologie source vers les instances d'une ontologie cible à partir des ponts sémantiques, créés précédemment, et enfin vérification des résultats obtenus. Quant à la dimension verticale, elle est constituée de quatre modules additionnels (gestion de l'évolution, gestion des ponts sémantiques, construction d'un consensus coopératif et Interface graphique).

- **FOAM (Framework for Ontology Alignment and Mapping)** est un Framework utilisé dans plusieurs systèmes comme QOM2, NOM3, APFEL4, à des fins d'intégration de données, de fusion d'ontologies, d'évolution d'ontologies. L'outil implémente plusieurs mesures et stratégies existantes de recherche de similarités et permet de faire des mappings entre des ontologies décrites en OWL. Le processus général d'alignement est le suivant: on sélectionne les paires d'entités à comparer ainsi que les caractéristiques sur lesquels se fera la comparaison. Le système calcule, pour chaque paire et pour chaque caractéristique, une similarité. Ces résultats sont combinés pour obtenir la similarité finale entre chaque paire d'entités. À partir de ces résultats, FOAM permet de proposer à l'utilisateur un ensemble de suggestions d'alignement, qu'il peut accepter ou rejeter.
- **RiMOM (Risk Minimisation based Ontology Mapping)** est un Framework interactif implémentant diverses stratégies pour l'alignement d'ontologies et suivant plusieurs étapes: la première consiste à sélectionner les matchers à utiliser selon la similarité supposée entre les ontologies (terminologique ou structurelle). Dans la deuxième phase, plusieurs mesures sont appliquées, indépendamment les unes des autres. Ensuite, les résultats sont agrégés en utilisant une fonction d'interpolation linéaire. La troisième étape consiste à propager les similarités (de concept à concept, de propriété à propriété et de concept à propriété). Enfin, la dernière étape génère des mappings à partir des résultats obtenus précédemment. Le processus est itératif, avec une validation des résultats à chaque itération.

II.12. Conclusion

Dans ce chapitre, l'accent est mis sur les ontologies ainsi que leurs utilisations. Les ontologies ont pour rôle de décrire formellement des concepts et des relations entre ces concepts. La tâche de modélisation d'une ontologie est réalisée en suivant un ensemble d'étapes ordonnées et reliées entre elles. Les types d'ontologies sont reliés entre eux afin de permettre la représentation de l'information de façon cohérente et efficace.

Dans le 3em chapitre de ce mémoire ont vas démontrer notre conception et mis en oeuvre de notre application.

Chapitre III :
Conception et mise en œuvre

III. Conception et mise en œuvre

III.1. Introduction

L'objectif de notre projet est de proposer une solution à la problématique d'évaluation avec question libre, qui consiste principalement à faire des tests. Cette application facilite à l'apprenant la réalisation de ces activités pédagogiques.

Le présent chapitre est consacré à la Conception et mise en œuvre du Dispositif de Formation Mobile. Cette conception s'effectue en élaborant les diagrammes classe, les diagrammes de séquence relatifs à notre dispositif.

III.2. UML (Unified Modeling Language)

Pour la conception de notre projet nous avons choisi d'utiliser le langage de modélisation UML (Unified Modeling Language) car UML est le langage de modélisation le plus connu et le plus utilisé au monde. UML représente plusieurs méthodes sous format graphique. L'usage d'une représentation graphique est un complément excellent à celui des représentations textuelles. Son utilisation permet de diminuer les ambiguïtés et facilite la programmation de l'application.

III.2.1. Définition

UML (en anglais *Unified Modeling Language* ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes¹. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. De plus, cela autorise l'échange électronique de documents qui deviennent des spécifications exécutables en UML [22].

UML représente l'état de l'art des langages de modélisation objet, il fournit les fondements pour spécifier, construire, visualiser et décrire les artefacts d'un système logiciel.

UML est un langage de modélisation objet qui facilite l'expression et la communication de modèles en fournissant des symboles.

II.2.2. Modèles d'UML

UML un formalisme, ou plutôt un ensemble de formalismes (modèles) permettant d'appréhender un problème ou un domaine et de le modéliser, Le succès du développement d'une application dépend évidemment de la bonne utilisation d'un langage comme UML mais il dépend surtout de la façon dont on utilise ce langage. Les 13 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

II.2.2.1. Diagrammes en UML

a) Diagrammes statiques :

❖ Diagramme des cas d'utilisation (use-cases ou Use Case Diagram)

Il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système. Ce type de diagrammes interviennent tout au long du cycle de développement, depuis le cahier des charges jusqu'à la fin de réalisation.

❖ Diagramme de classes (Class diagram)

Il représente les classes intervenant dans le système. Un diagramme de classe exprime de manière générale la structure statique d'un système en terme de classe et relation entre ces classes. Outre les classes, ils présentent un ensemble d'interface et de paquetage ainsi que leurs relations.

b) Diagrammes dynamiques :

❖ Diagramme de séquence (SequenceDiagram)

Représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

Ce type de diagramme sert à modéliser les aspects dynamiques des diagrammes des systèmes temps réels et des scénarios complexes. Dans ce type de diagramme, l'accent est mis sur la technologie des envois de messages.

❖ Diagramme d'activité (ActivityDiagram)

Permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

Un diagramme d'activité représente l'état de l'exécution d'un mécanisme, sous la forme d'un déroulement d'étapes regroupées séquentiellement dans des branches parallèles de flot de contrôle. Le début et la fin d'un mécanisme sont définis respectivement par un état initial et un état final [23].

III.3. Conception

Notre conception se base sur le diagramme de classe et le diagramme de séquence.

III.3.1. Diagrammes

II.3.1.1. Diagramme de classe

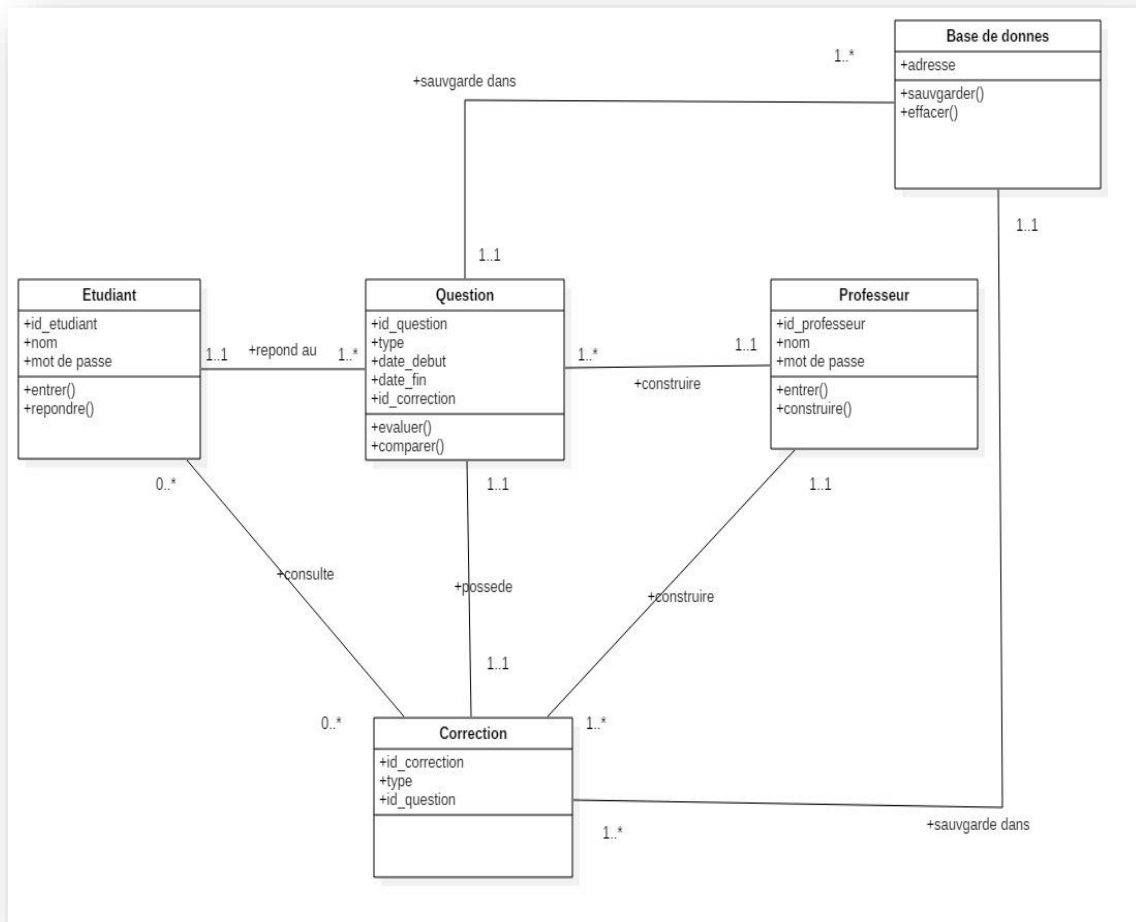


Figure III-1: Diagramme de classe.

Figure III.1 représente le diagramme de classe, possède 6 classe définit comme suit :

1) **Classe étudiant:** contient le nom et mot de passe qui lui permettant d'accéder à l'application pour répondre à des questions.

2) **Classe Professeur:** contient le nom et mot de passe qui lui permettant d'accéder à l'application pour construire les questions.

3) **Classe Question:** contient le types des questions avec une date de début et une date de fin, ces questions est construit par le professeur et répondre par l'étudiant.

4) **Classe Correction:** contient un type de la correction et un id-question et se relier avec, elle est construit par le professeur.

5) **Classe Base de donné :** contient tous les données utilisé dans cette application.

III.3.1.2. Diagramme de séquence

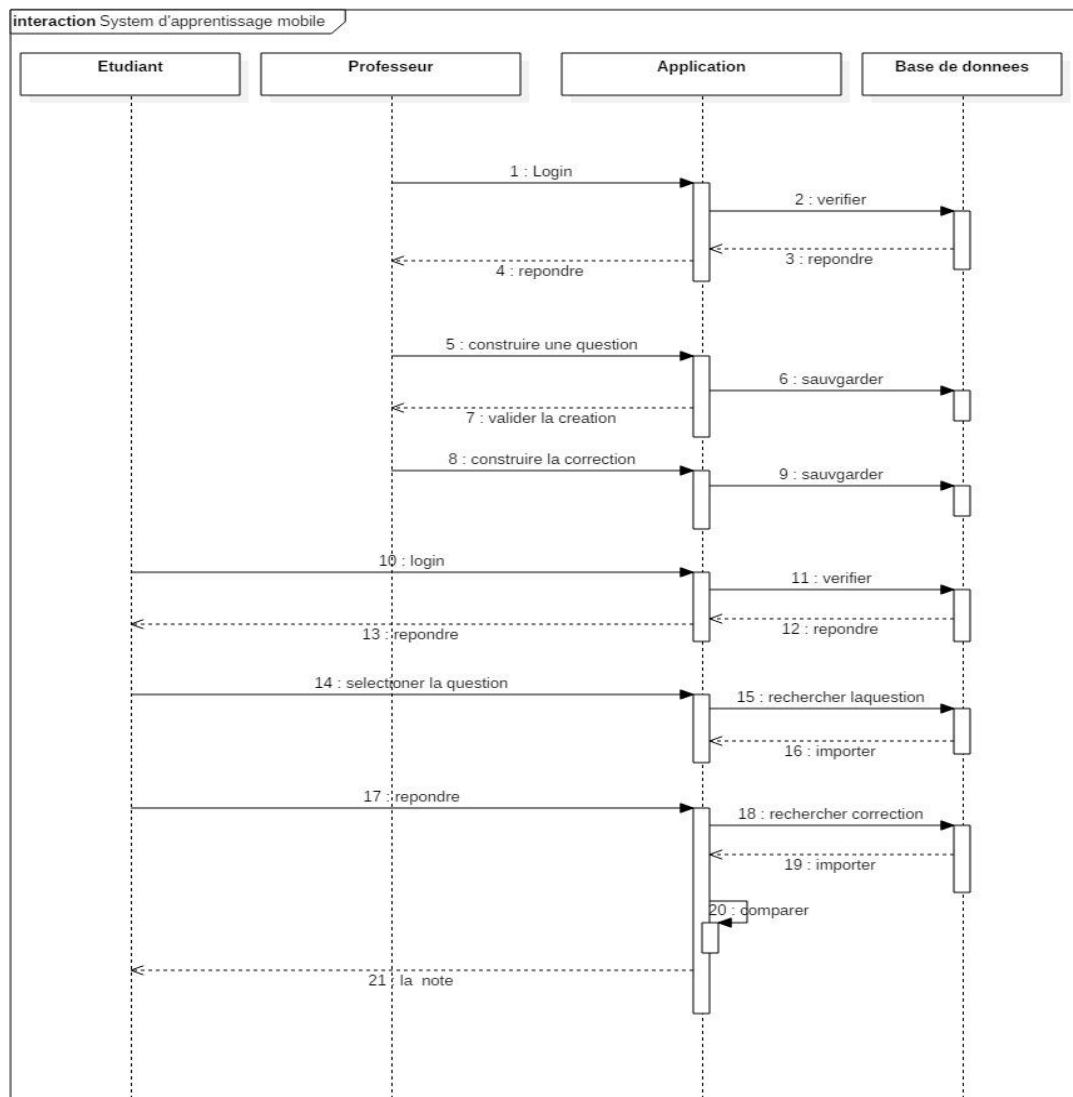


Figure III-2: Diagramme de séquence.

Figure III.2 représente le diagramme de séquence .Il représente les taches en fonction du temps. De premier temps le professeur accédé àl’application (login),le système accédé a la base de donné et vérifier les données de professeur , après cette étape ce professeur peut construire des questions et des réponses qui serrant sauvgarder dans la base de donnée.

Et pour l’étudiant sa sera la même chose, au lieu de construire les questions et réponses il doit répondre a ces questions et consulte la résultat.

III.4. Environnement de développement

III.4.1 Environnement matériel

Notre plateforme a été développée sur une machine Intel(R) I57500Processor(3.4 GHz, 6MB L3 cache) RAM 8 Go, sous le système d'exploitation MicrosoftWindows10 professionnel (64bits).

III.4.2 Environnement Logiciel

L'implémentation de notre application a été réalisée avec le langage de programmationJAVA sous la plateforme NetBeans IDE 8.0.2 et androidstudio . Nous avons aussi utilisé des APIs JAVAetle langage python .

III.4.2.1 Langage JAVA



JAVA est un langage de programmation à usage général qui a été inventé au début des années 90 et dont les droits sont détenus par la société Sun Microsystems. La création du langage provient des résultats obtenus avec le C++ dans l'élaboration de programmes de domotique qui furent jugés insatisfaisants. Très vite, pendant sa conception, il s'est avéré que Java convenait aussi parfaitement à la programmation multimédia, et au monde de l'internet qui commençait à éclore et à se faire connaître du grand public.

Depuis sa création, le langage a subi de nombreuses évolutions et corrections. Les mises à jour régulières ont permis au fur et à mesure d'inclure de nombreuses bibliothèques, permettant de développer plus facilement des applications complexes.

La manière d'exploiter un code source Java est singulière : le code source est d'abord compilé en utilisant un compilateur Java ; le résultat de cette compilation est ensuite interprété par la machine cible, au moyen d'un interpréteur Java.

Ce concept d'exploitation découle de la volonté de produire un langage le plus indépendant possible des spécificités du hardware. Pour exécuter un programme Java, il est donc nécessaire

et suffisant d'installer sur la machine cible un interpréteur Java, plus communément appelé machine virtuelle Java ou JVM (Java Virtual Machine), qui a en charge l'interfaçage entre les réquisitions de l'application Java et les spécificités hardware et/ou software de la plate-forme d'exécution.

III.4.2.2 Langage PYTHON



Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Dès l'instant où vous l'installez sur votre ordinateur, vous disposez de nombreuses fonctionnalités intégrées au langage que nous allons découvrir tout au long de ce livre.

Il est, en outre, très facile d'étendre les fonctionnalités existantes il existe ce qu'on appelle des **bibliothèques** qui aident le développeur à travailler sur des projets particuliers. Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python.

III.4.2.3 Lien De Communication Entre Les Différentes Plateformes

4.2.3.1 Sockets :

Dans le contexte des logiciels, on peut le traduire par « connecteur réseau » ou « interface de connexion »¹.

Apparu dans les systèmes UNIX, un socket est un élément logiciel qui est aujourd'hui répandu dans la plupart des systèmes d'exploitation. Il s'agit d'une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau.

Il lui sera ainsi par exemple aisé d'établir une session TCP, puis de recevoir et d'expédier des données grâce à elle. Cela simplifie sa tâche car cette couche logicielle, de laquelle il requiert des services en appelant des fonctions, masque le travail nécessaire de gestion du réseau, pris en charge par le système. Le terme socket désigne en pratique chaque variable employée dans un programme afin de gérer l'une des sessions.

4.2.3.2 Communication entre la plateforme Android et python :

On a réalisé le traitement de texte en utilisant python à l'aide des deux bibliothèques RAKE et NLTK, on a encapsulé les scripts de python par une couche serveur et on a créé une application Android encapsulée par une couche client. On utilise une connexion TCP/IP à l'aide des sockets sur java pour faire la communication entre l'application Android et les scripts python.

III.4.2.4 NetBeans



NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDENetBeans s'appuie sur cette plate forme.

III.4.2.5 Android Studio



Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA.

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.

III.4.2.6 Pycharm



PyCharm est un environnement de développement intégré (abrégé EDI en français ou en anglais : IDE (IntegratedDevelopmentEnvironment)) utilisé pour programmer en Python.

Il offre l'analyse de code, un débogueur graphique, la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

Il est développé par l'entreprise tchèque JetBrains. Il est multi-plateforme et fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, réalisé sous licence propriétaire, et en édition communautaire réalisé sous licence Apache.

III.4.2.6 Les Bibliothèques utilise

III.4.2.6.1 NLTK

Natural LanguageToolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API).

III.4.2.6.2 RAKE

Une mise en œuvre de module de Python de l'Extraction de Mot-clé Automatique Rapide (RATISSE) l'algorithme comme décrit dans : Rose, S., Engel, D., Cramer, N. et Cowley, W. (2010). Extraction de Mot-clé Automatique de Documents Individuels. Dans M. W. Le Berry et J. Kogan (Ed.), Extraction de Texte : Théorie et Demandes(Applications) : John Wiley et Fils. Initialement par @aneesha.

III.4.3. Architecture globale de l'application

On peut dire que notre plateforme est un composant logiciel qui sert à utiliser les avantages de mobile pour l'apprentissage à distance précisément le type de réponse libre fournit par l'apprenant et le rendre sous forme d'une ontologie et le comparer avec l'ontologie du réponse type qu'elle a été fournie par l'enseignant pour évaluer l'apprenant et le donner une note.

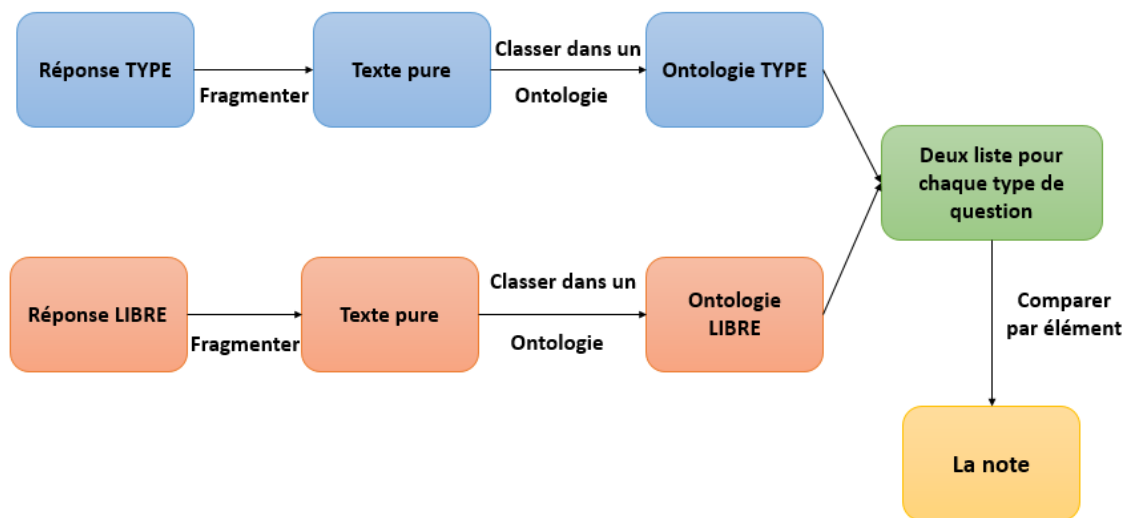
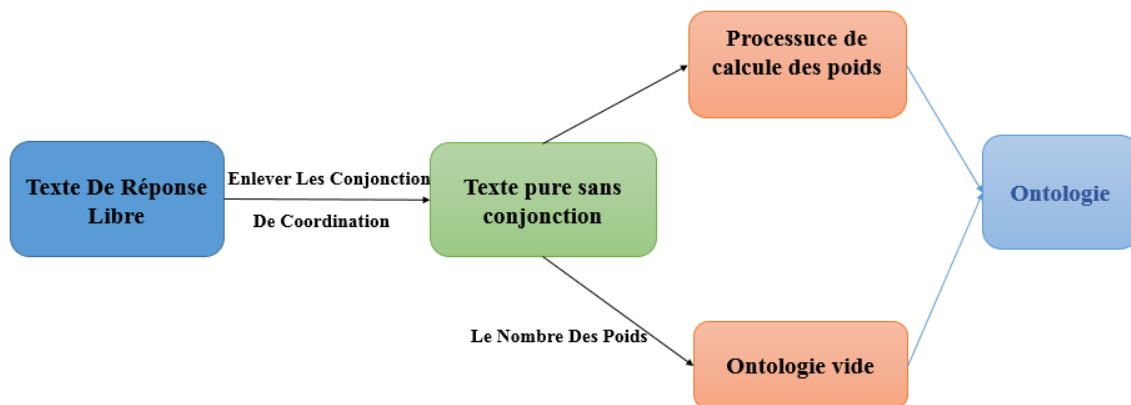


Figure III-3: schéma du processus de traitement d'une question libre

Le schéma de la Figure (III-3) représente l'architecture générale de notre système dans

laquelle sont représentés les différents composants de notre application. Pour chaque composant nous présentons sa structure et son fonctionnement.

III.4.3.1. Traitement des Réponses et création des ontologies :



Dans cette étape de l'application la réponse (étudiant / enseignant) sera traité et transformé sous forme d'ontologie après avoir fait l'extraction des mots clefs et l'élimination des conjonctions de coordination tout ça en utilisant les méthodes de traitement de texte.

```

Result: Ontologie
initialisation X[ ] P[ ];
P = Extraction des phrases clefs ;
while i lesser than P.length() do
|   while j lesser than i.length() do
| |   if (poïd greater than 40.0) then
| | |   X.append(j);
| | |   end
| | |   j++;
| | end
|   i++;
end
while k lesser than X.length() do
|   create class (k);
|   add attribute(k);
|   k++;
end

```

Algorithm 1: Creation de l'ontologie Des Reponses

III.4.3.2. La comparaison des ontologies et le systèmes des notes

L'étape de la comparaison démarre avec deux ontologie (réponse type / réponse libre) qui sont construit en classe ordonne en rapport des poids des mots, la comparaison utilise dans notre application est une comparaison naïf que sert a compare chaque classe de l'ontologie de la réponse libre a tous les classe de l'ontologie de réponse type .

Le nombre totale des classe de la réponse type se sera la note complète si la reponse libre de l'étudiant a le mm nombre que la reponse type il aura la note complet ,si non la note sera dans l'un des trois cas suivant si la réponse d'étudiant a 50% de la reponse type la note sera ½ de la note complète ,si le pourcentage est supérieur a 50% mais pas 100% la note sera donne complète mais si la note est inferieur a 50% la note se sera 0

```

initialisation X1, X2, Y1, Y2, Cpt, Note ;
Y1 = GetClasses.Getattributes(X1) Y2 = GetClasses.Getattributes(X2)
while i lesser than Y1.length() do
  while j lesser than Y2.length() do
    if (i equals j) then
      | Cpt++;
    end
    j++;
  end
  i++;
end
if (Cpt greater (Max(Y2)/2) then
  | Note = 10;
end
if (Cpt lesser (Max(Y2)/2) then
  | Note = 0;
end
if (Cpt equals (Max(Y2)/2) then
  | Note = 5;
end
end

```

Algorithm 2 : Comparaison Des Ontologie Et Derivation Des Notes

III.5. Définition fonctionnelle

Tout d'abord on a appelés notre application **Smart AnswerChecker**.

Le processus suivant décrit le fonctionnement de notre système :

Phase1 :Après avoir lancé l'application il faut tout d'abord choisir votre role d'utilisation (enseignant/étudiant)

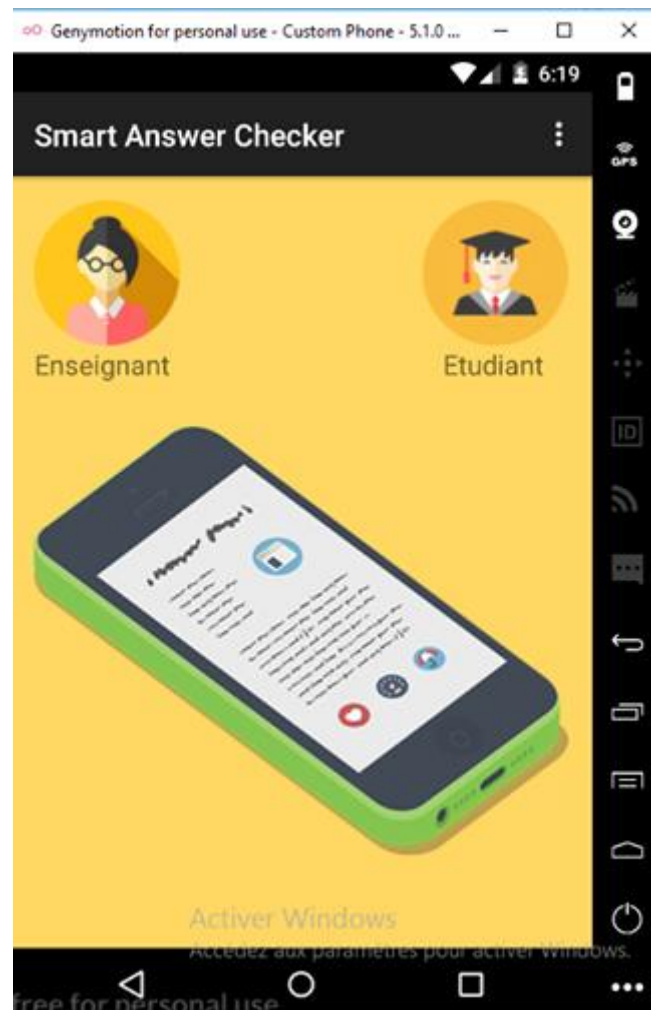


Figure III-5: Page d'accueil de l'application.

Phase2 : Si on a choisit un utilisateur enseignant, il passera donc à la création des questions. La figure suivante montre l'existence de deux types de questions (question libre/question à cocher).

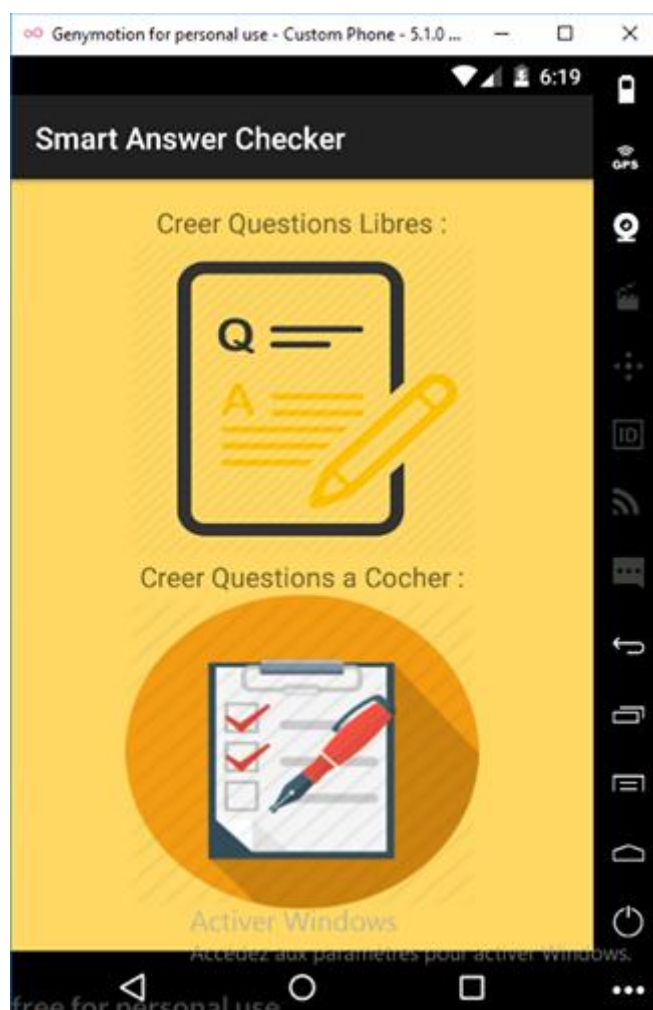


Figure III-6: Menu enseignant.

Phase3 : Si on a choisi l'utilisateur étudiant, la figure suivante montre les choix de ce type d'utilisateur (répondre a questions libres/répondre a questions cocher).



Figure III-7: Menu étudiant.

Phase 4 : si l'étudiant choisi de répondre au question libre il va lui apparaitre la figure suivant

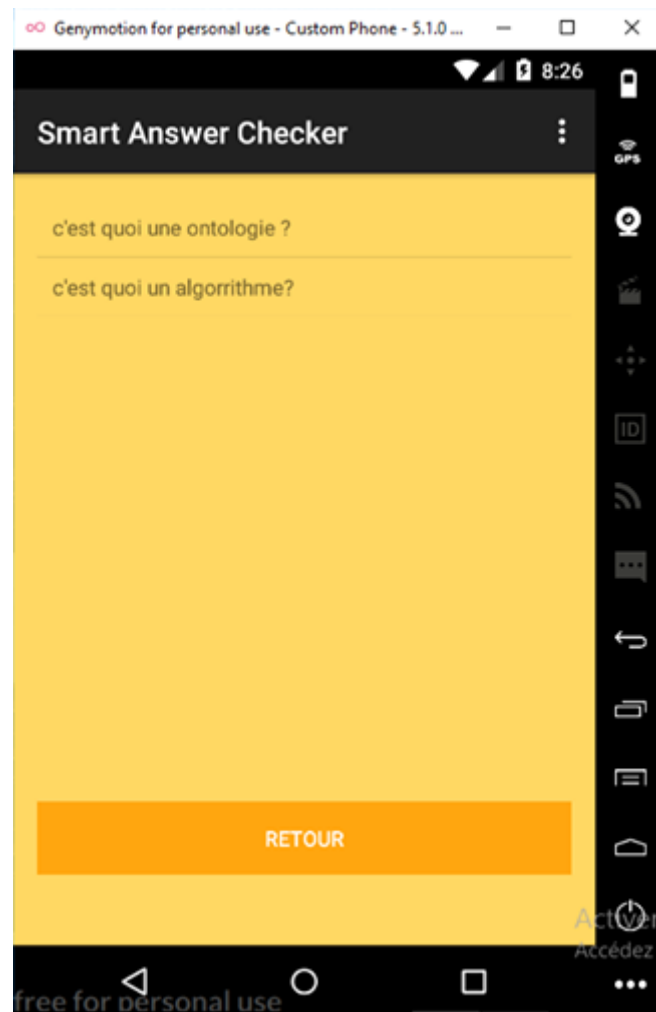


Figure III-8: Menu des questions libre (etudiant).

Phase 5 : après qu'il choisi quelle question il va réponde il va passe a une nouvel page qui lui aide à écrire sa réponse comme il est démontre dans la figure suivant

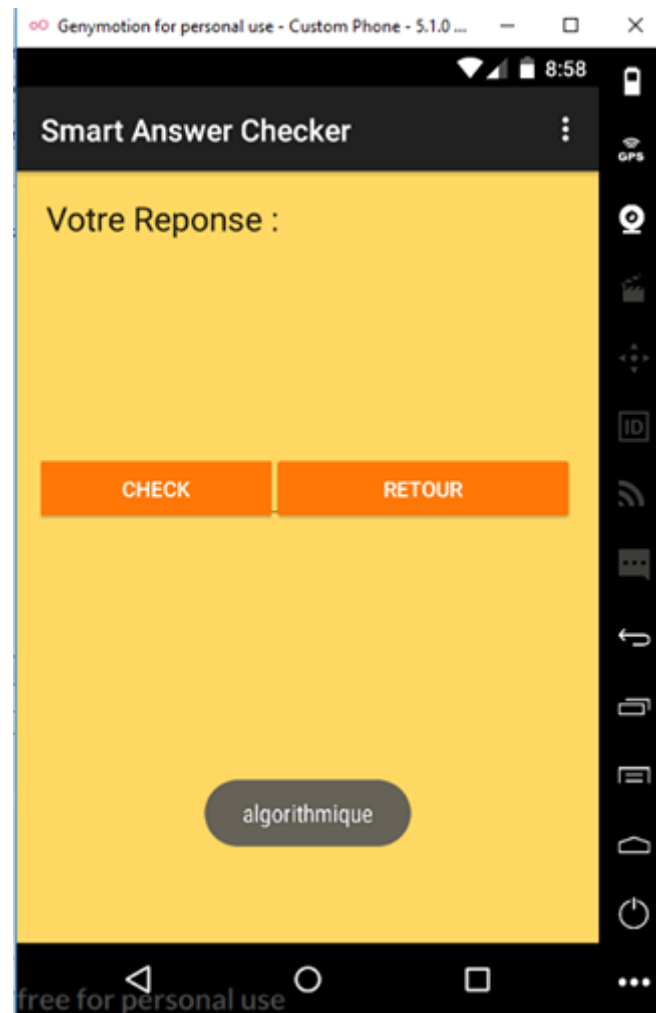


Figure III-9: La zone de réponse libre de l'étudiant.

Phase 6 : quand il réponde a la question il click sure le buttons check et il va voir sa note, si il veut retourne et répondre a une autre réponse il click sure retour

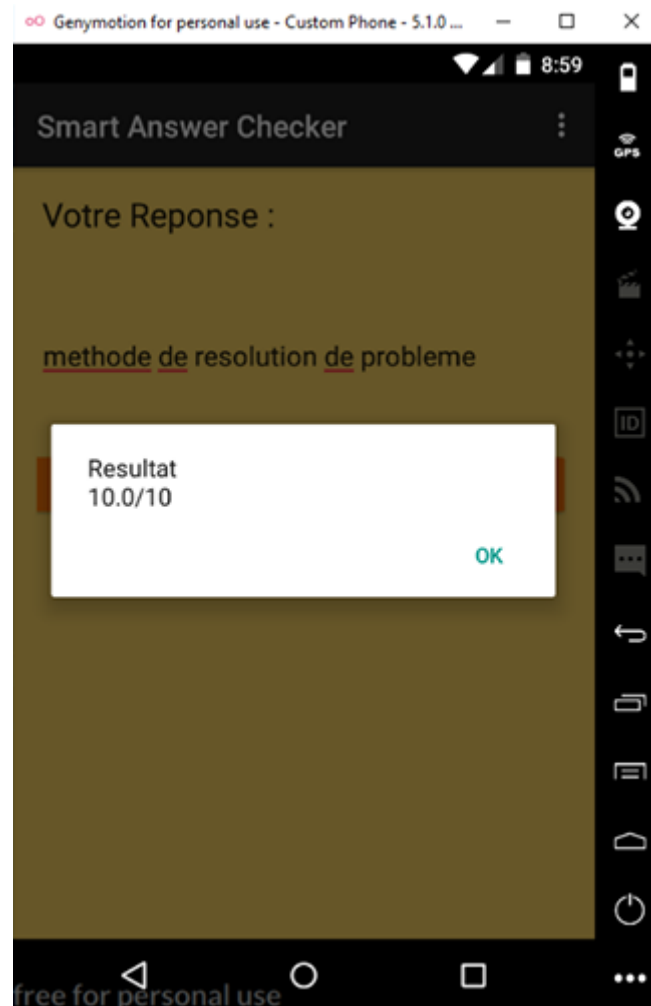


Figure III-10: La note de la réponse de l'étudiant.

Phase 7 :dans cette figure l'étudiant a déjà choisi le type de réponse pour une réponse a cochet

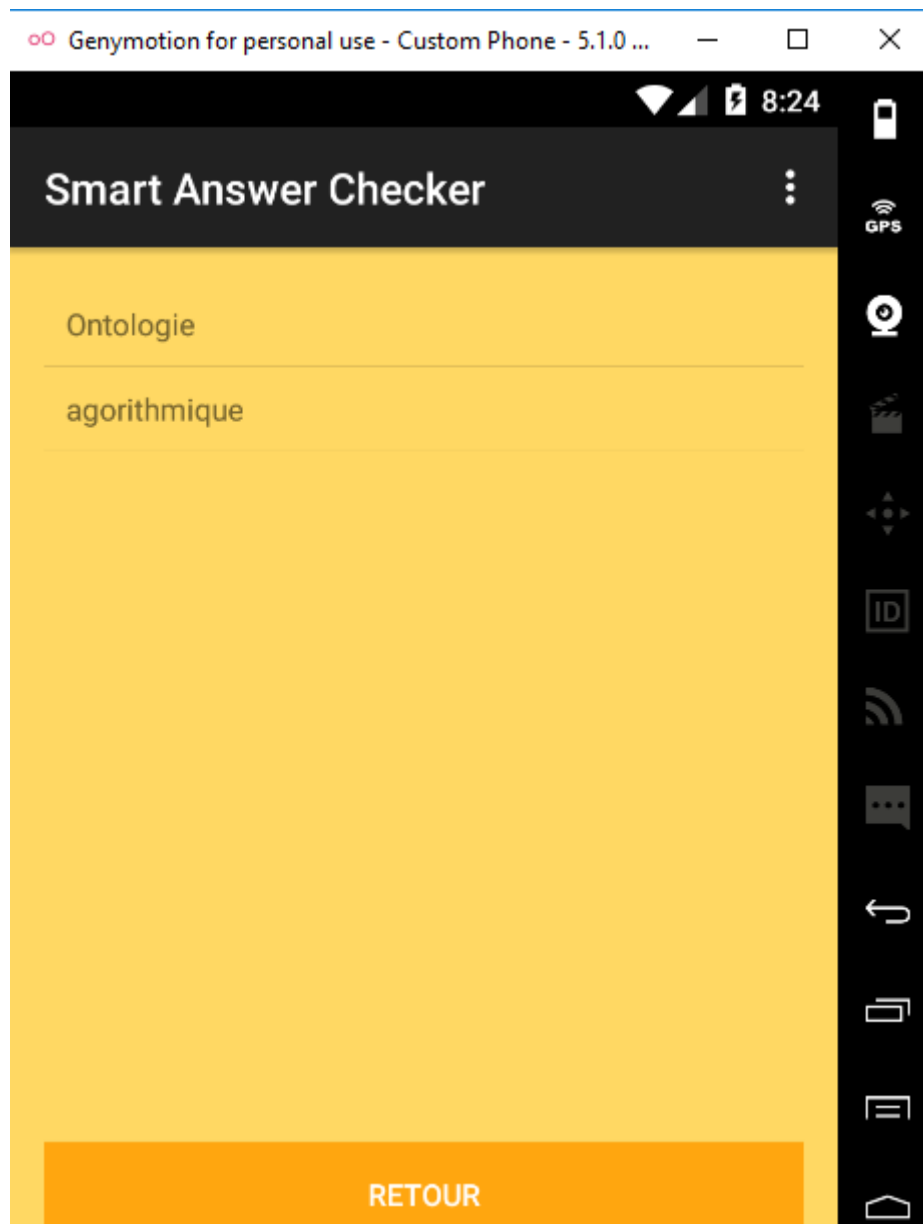


Figure III-11: Listes des questions à choix pour étudiant.

Phase 8 : après avoir choisi la réponse voulus répondre il apparaitre les multi choix pour les sélections et répondre a la question

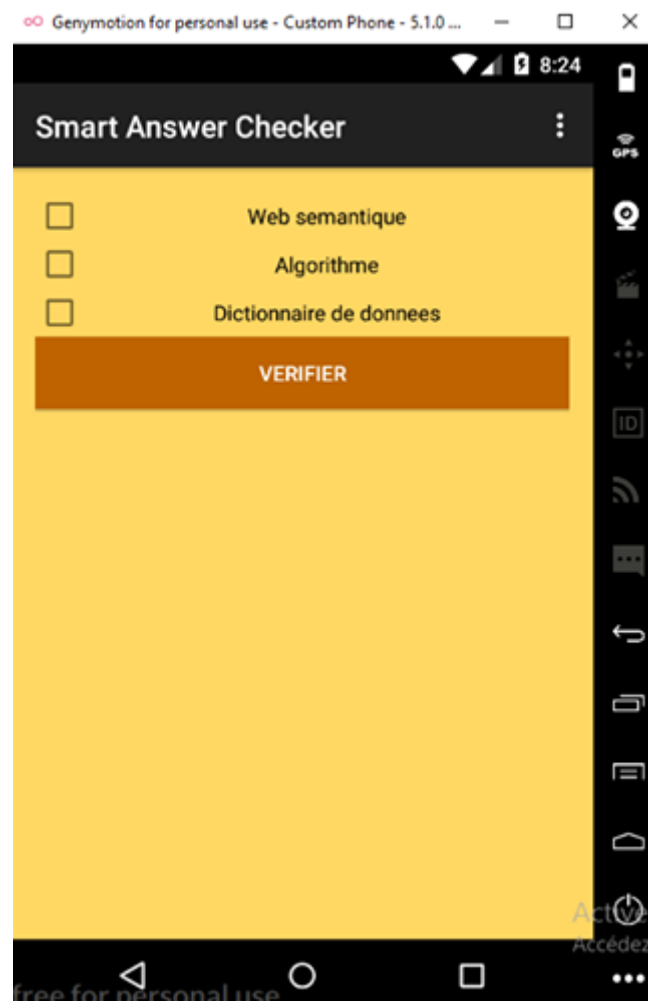


Figure III-12: les choix a coché par l'étudiant.

Phase 9 : l'étudiant doit répondea la question par click sure la bonne réponse ou la bonne réponse pour qu'il puits consulte sa note comme il le démontre la figure suivant



Figure III-13: Note de réponse de l'étudiant.

III.6. Conclusion

Dans ce chapitre nous avons élaboré des diagrammes statique (diagramme de classe et diagramme de cas d'utilisation) et dynamiques (diagramme de séquence et diagramme d'activité) en UML. Ceci en utilisant l'outil de modélisation «Pacstar UML Diagramme », ce qui nous a permis de concevoir notre application.

Conclusion Générale

Conclusion générale

Les travaux qui ont été menés et décrits dans ce mémoire se situent dans le cadre de l'évaluation des connaissances des apprenants.

Nous avons consacré les deux premiers chapitres de ce mémoire à une examinaisons de l'état de l'art qui se tourne autour de l'environnement Informatique d'Apprentissage Humain (EIAH) et l'explication de la notion ontologie et ses notions et comment on peut la développer, et on a défini le processus d'alignement, ses dimensions et les méthodes de base utilisées dans l'alignement.

Enfin nous avons présenté notre application avec un scénario de déploiement. Dans notre travail nous avons contribué au développement d'une application mobile pour le m-Learning. Plus précisément la plateforme proposé est un composant logiciel qui utilise les avantages de mobile pour l'apprentissage a distant, précisément le type de réponse libre fournit par l'apprenant et le rendre sous forme d'une ontologie et le compare avec l'ontologie du réponse type qu'elle a été fourni par l'enseignant pour évalue l'apparent et le donne une note.

En guise de perspectives nous envisageons de développer les aspects suivants :

- Intégrés des ontologies spécial pour chaque module.
- Ajouter une section des cours.
- Ajouter un autre type d'examen (examen finale) qui se passera en même temps.

Références bibliographiques

Bibliographie

- [1] <http://claroline.up.univ-mrs.fr/claroline/>
- [2] <http://www.ganesh.fr/>
- [3] <http://moodle.org/>
- [4] <http://www.e-doceo.net/>
- [5] <http://www.cerfi.ch>
- [6] <http://www.blackboard.com/>
- [7] <http://www.coursegenie.com>
- [8] <http://www.easygenerator.com>
- [9] <http://www.easyprof.com>
- [10] <http://e-learning.prestataires.com/conseils/quest-ce-que-le-learning>
- [11] <http://www.e-doceo.net/blog/formation-e-learning-avantages-et-inconvenients/>
- [12] <https://www.efrontlearning.com/blog/2011/01/introduction-to-evaluation-in-elearning.html>
- [Chuantao YIN 2011] Chuantao YIN, SAMCCO : un Système d'Apprentissage Mobile Contextuel et Collaboratif dans des Situations Professionnelles, Le 25 Janvier 2010, diplôme Doctora Ecole Centrale de Lyon.
- [John Traxler 2005] John Traxler, Defining mobile learning, 2005, University of Wolverhampton John.traxler@wlv.ac.uk
- [Caroline 2011] Caroline, Le m-learning, l'avenir de la formation ?, lundi 4 juillet 2011, <http://mobile-learning/Le-m-learning,-avenir-de-la-formation.htm>.
- [Sharples,&Arnedillo Sánchez 2007] Sharples, M.; Arnedillo Sánchez I.; Milrad M.; Vavoula G, Mobile Learning: Small Devices, Big Issues In Technology Enhanced Learning, 2007.
- [UNESCO 1995-2011] Les TIC dans l'éducation, l'organisation des Nations Unies pour l'éducation la science et la culture, 2011, © UNESCO 1995-2011.
- [Pachler Norbert, Bachmair Ben 2010] Pachler Norbert, Bachmair Ben et Cook John, Mobile Learning: Structures, Agency, Practices. New York, 2010.
- [Laure Endrizzi 2011] Laure Endrizzi, Les apprentissages nomades en Europe : éléments pour une cartographie de la recherche, 2011.
- [Meyer, & al 2006] Meyer, et al. Caractérisation De Situations De M-Learning, 2006, Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise, Toulouse.

- [KouninefBelkacem,&Djelti Mohamed 2008]KouninefBelkacem,Djelti Mohamed ,Rerbal Sidi Mohamed ,conception et réalisation de la plateforme e-learning avec la migration m-learning , 2008, institut des télécommunications d’Oran .
- [Naismith, L., &Sharples 2004]Naismith, L., Lonsdale, P., Vavoula, G.-N., &Sharples, M , Literature Review in Mobile Technologies and Learning. Bristol, NESTA FutureLab, 2004.
- [MATI Montréal 2011]Comment les jeunes utilisent-ils les technologies mobiles? Que peuvent apporter de nouveau ces technologies à l’apprentissage?, MATI Montréal, 18 février 2011.
- [Fabien Lienard]Fabien Lienard ,IUT Le Havre , Département Information Communication BP 4006 – 76 610 le Havre,abien.lienard@univ-lehavre.fr.
- [Jacob,&Liénard 2009] Jacob, Liénard, Marketing mobile, 2009, Dunod
- [Anderson &Reiser 85] : Anderson J.R., Reiser B.J. The Lisp Tutor, Byte, 10(4), 1985, p. 159-175.
- [Baker & al 04] :Baker, R. S., Corbett, A. T., &Koedinger, K. R. 2004. Detecting student misuse of intelligent tutoring systems. ITS 2004, LNCS 3220, 531-540.
- [Bisault&Lavarde 95] :Bisault J., Lavarde, A. Le mémoire en IUFM : Théorie et pratique, Beauvexcedex, CDDP Oise, 1995, 104 p, ISBN2903729700.
- [Bouarab-Dahmani 00] : Bouarab-Dahmani F. Conception et évaluation de la composante experte du domaine d’un système intelligent pour l’enseignement de l’algorithmique. Thèse de Doctorat, soutenu le 29/02/2000 à l’université Mouloud Mammeri de Tizi-Ouzou Option : Système d’information et de Gestion Avancés.
- [Brown & Burton, 78] : Burton R., Brown J-S. Diagnostic Models for procedural bugs in basic mathematical skills, Cognitive Science, Volume 2, pages 155-192, 1978.
- [Després& Leroux 97] : Després C., Leroux P. Raisonner sur la trace : Analyse de sessions avec l’application ROBOTEACH, Actes des 5^{èmes} journées EIAO de Cachan, Hermès, pp. 277-288, 1997.
- [Fernandez & al 04] : Fresno-Fernandez V., Montalvo-Herranz S., Pérez-Iglesias J., Urquiza-Fuentes J., Velazquez-Iturbide J.A. An approach to the use and automatic generation of web-based learning materials, W. Liu & al. edition, ICWL 2004, LNCS 3143, pp. 201-208, 2004

[Goldstein & Miller 76] : Goldstein I., Miller M. Structured planning and debugging : a linguistic theory of design, Artificial intelligence Lab Memo 387, Massachusetts Institute of technology, Cambridge, Massachusetts, 1976.

[Grandbastien02] : Grandbastien M., interview dans l'ASTI (Association française des Sciences et Technologies de l'Information), N° 79 DU 17 Juin 2002.

[Hadji 97] : Hadji C. L'évaluation démystifiée, ESF, Paris, 1997

[Horacek et Wolska 06] :Horacek H., Wolska M. Handling errors in mathematical formulas, in ITS 2006, LNCS 4053, PP 339-348, 2006.

[Jean 00] : Jean S. Pépite, un système d'assistance au diagnostic des compétences », thèse de doctorat de l'Université du Maine, 2000

[Johnson 86] : Johnson W. L. Intention-Based Diagnosis of Novice Programming Errors , Morgan Kaufmann Publishers, Los Altos, 1986.

[Labat 02] : Labat J. M. EIAH : Quel retour d'information pour le tuteur, Conférence TICE, 2002, P. 81-87, Novembre 2002.

[Liu & al. 04] : Liu Q., Yang Z., Yan K., Wu D. Reusable Learning Object and its strategies for e-learning, W. Liu & al. edition, ICWL 2004, LNCS 3143, pp. 386-392, 2004.

[L'haire&Faltin 03] : Haire S., Faltin A. V. Diagnostic d'Erreurs dans le Projet FreeText ", Alsic journal, volume 6, N°2, Décembre, 2003, pp. 21-37.

[Michel & Rouissi 03] : Michel C., Rouissi S. E-learning : normes et spécifications, Etude des spécifications LOM et IMS-QTI caractérisant des documents inter-échangeables et réutilisables pour l'acquisition et l'évaluation des connaissances », dans la revue Document numérique, 2003

[Nicaud& al.06]: Nicaud J.F., Chaachoua H., Bittar M. Automatic calculation of students' conceptions in elementary algebra from aplusix log files, in ITS 2006, LNCS 4053, PP 433-442, 2006.

[O'Neil 01] : O'Neil M. What is E-learning, Home Magazine, May 2001.

[Oravep 00] : ORAVEP. Etude comparative technique et pédagogique des plateformes pour la formation à distance », www.oravep.asso.fr, 2000.

[Py 01] : Py, D. Environnements interactifs d'apprentissage et démonstration en géométrie. Habilitation à diriger des recherches présentée à l'Université de Rennes 1, Institut de Formation Supérieure en Informatique et en Communication.

[Ruth 73] : Ruth G. Analysis of algorithm implementations, Technical report MIT project MAC TR 130, Massachusetts Institute of technology, 1973.

[Saleh & Bouyahi 04] : Saleh I., Bouyahi S. Enseignement ouvert et à distance : épistémologie et usages, chapitre 5, Edition Lavoisier, 2004.

[Sleeman & Smith 81] : Sleeman D., Smith M. Modelling student's problem solving, Artificial Intelligence, (16), 171-187, 1981.

[Suraweera & al. 04] : Suraweera P., Mitrovic A., Martin B., The role of domain ontology in knowledge acquisition for ITSs, Proceedings of the 7th international conference, ITS 2004, Brazil 2004.

[Tchounikine 09] : Tchounikine P. Précis de recherche en ingénierie des EIAH, Version Juin 2009, en ligne sur le Web.

[Wenger 87] : Wenger E. Artificial Intelligence and Tutoring Systems, Computational and Cognitive Approaches to the Communication of Knowledge, Morgan Kaufmann Publishers Inc., Los Altos, 1987.

[Westbrook 93] : Westbrook R. B. John Dewey (1859-1952). Revue trimestrielle d'éducation comparée (Paris, UNESCO : Bureau international d'éducation), vol. XXIII, n° 1-2, 1993, p. 277-93.

[Ellis 07] : Ellis H. J.C. An Assessment of a Self-Directed Learning Approach in a Graduate Web Application Design and Development Course, IEEE Transactions on education, Vol. 50, N° 1, February 2007.

[Hatzily & Prentzas 04] : Hatzilygeroudis I., Prentzas J. Knowledge representation requirements for intelligent tutoring systems, Proceedings of the 7th international conference, ITS 2004, Brazil 2004.

[Horacek et Wolska 06] : Horacek H., Wolska M. Handling errors in mathematical formulas, in ITS 2006, LNCS 4053, PP 339-348, 2006.

[Abdeltif Elbyed.ROMIE] une approche d'alignement d'ontologies à base d'instances, Institut National des Télécommunications (2009. Français) <https://tel.archives-ouvertes.fr/tel-00541874/document>, (consulté le 15/02/2017)

[Lynda DJAKHDJAKHA], Un formalisme pour l'alignement des ontologies multi-points de vue basé sur une extension de la logique de description, (<http://www.univ-constantine2.dz/files/Theses/Informatique/Doctorat/Djakhdjakha-Lynda.pdf>), (consulté le 15/02/2017)

[OUSLIM HAMZA], Analyse de l'Apport de la Classification Automatique pour l'Identification de l'Alignement entre Ontologies, Mémoire Master, Université Oran 1 Ahmed Benbella, (2015).

[Natalya F. Noy et Deborah L. McGuinness], Guide pour la création de votre première ontologie, Université de Stanford, (<http://www.bnf.fr/documents/no-DevOnto.pdf>) (consulté le 25/2/2017)

[I. Bouazza], Un environnement basé sur l'apprentissage pour l'alignement des ontologies, Mémoire Master, Université Oran 1 Ahmed Benbella, (2014).

[6] <http://exmo.inria.fr> May 16, 2014 Alignement d'ontologies (consulté le 17/02/2017).

[7] <http://biblio.univ-annaba.dz/wp-content/uploads/2014/07/These-Djeddi-Warith-Eddine.pdf> [26/2/2017] (consulté le 02/03/2017).

[8] http://liris.cnrs.fr/sbc/sbcwiki/images/c/cf/RISE2010_1.pdf, (consulté le 3/3/2017).