

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Informatique Académique

Thème :

**Optimisation et Classification des chemins
d'apprentissage**

Encadré Par :

Mr. Bourbia Riad

Présenté par :

Lachaal Bassim

Araba Abderaouf

Juin 2014

Remerciements

Merci mon Dieu de nous avoir donné la force, la patience et la volonté d'arriver au terme de ce travail.

Nous tient à exprimer notre profonde gratitude et nos sincères remerciements à notre encadreur :

« Mr: BOURBIA Riad »

Pour son aide précieuse et ses conseils au cours de l'élaboration de ce travail.

On adresse nos vifs remerciements aux membres du jury pour avoir fait l'honneur d'évaluer notre travail.

Nos remerciements vont également à, nos amis pour leurs aides durant ce travail.

Et enfin, que nos chers parents et familles, et bien avant tout, trouvent ici l'expression de nos remerciements les plus sincères et les plus profonds en reconnaissance de leurs sacrifices, aides, soutien et encouragement afin de nous assurer cette formation dans les meilleures conditions.



Bassim & Abderaouf



Dédicaces

Je dédie ce travail à :

A ma très chère mère Farida

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon très cher père Madjid

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A notre chère et dynamique Encadreur « BOURBIA RIAD »

Un remerciement particulier et sincère pour tous vos efforts fournis. Vous avez toujours été présente.

Que ce travail soit un témoignage de ma gratitude et mon profond respect.

Mes souers , ma grand mère , mes oncles et tantes.

A tous mes cousins et cousines Amin, Hind, Jad, Khadija, Rahim, Younes son oublié le reste de la famille.

Mes amis Hamza, Zdame, Soufien, Adnen , Walide et Abderahman

Tous ceux qui de prêt ou de loin ont contribué à la réalisation de ce mémoire je dis du fond du cœur merci.

Et en final mon frère et amis Araba Abdraouf

Lachaal Bassim



Dédicaces

Je dédie ce travail à :

A ma très chère mère Fadila

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon très cher père Abdelghani

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A notre chère et dynamique Encadreur « bourbia riad »

Un remerciement particulier et sincère pour tous vos efforts fournis. Vous avez toujours été présente.

Que ce travail soit un témoignage de ma gratitude et mon profond respect.

Mes sœurs et Mon frère lotfi merci pour votre présence

A tous mes cousins et cousines khairedine, brahim, mohamed, anis, babo son oublié le reste de la famille.

Mes amis Walid, Abderahmén, Kouki, Hakou, Brahim ,Rahim, Amine et Rachide .

Tous ceux qui de prêt ou de loin ont contribué à la réalisation de ce mémoire je dis du fond du cœur merci.

Et en final mon frère et amis lachaal bassim

Araba Abderaouf

Résumé

Ce travail se place dans le cadre des EIAH. Il consiste à appliquer le paradigme d'adaptation à la formation à distance dispensée aux apprenants. L'adaptation est réalisée en appliquant un algorithme d'optimisation. En effet, de nouveaux paramètres de liberté sont introduits dans la mise en œuvre des pratiques d'E-learning. L'apprenant devient, le pilote de sa formation, il peut apprendre selon ses préférences, ses capacités en ayant des contenus personnalisés et des parcours plus adaptés.

Notre approche poursuivie consiste d'abord à évaluer l'apprenant psychologiquement et cognitivement afin de connaître son profil et lui fournir un chemin d'apprentissage adapté à ses compétences en utilisant l'ACO (Optimisation par Algorithme de Colonies de fourmis). La deuxième étape de ce travail consiste à classer les chemins d'apprentissage générés par l'ACO et ayant conduit l'apprenant à une réussite. Ces chemins considérés optimaux, seront recommandés aux futurs apprenants ayant des profils similaires.

Mots clés : E-Learning, optimisation, colonies de fourmis, adaptation, classification.

Table Des Matières

Résumé :	I
Table des matières :	II
Liste des figures :	VI
Liste des tableaux :	VIII
Introduction générale :	1
Chapitre I : Les Objets Pédagogiques Et Les styles d'apprentissages	
1. Introduction	3
2. L'Environnement EIAH	3
3. Définitions des concepts.....	4
3.1. Objets pédagogiques	4
3.2. Ressources pédagogiques	4
4. Propriétés des objets pédagogiques.....	5
5. Description des objets pédagogiques	7
5.1. Métadonnées.....	7
5.2. Modèles de description des objets pédagogiques.....	7
5.2.1. Learning Object Meta data (LOM)	7
5.3. Cycle de vie d'une ressource pédagogique	8
6. Différents supports de présentation.....	9
7. Style d'apprentissage.....	10
7.1. Qu'est-ce qu'un style d'apprentissage	10
7.2. Différentes approches et modèles	10
7.3. Style d'apprentissage et performances de l'apprenant :	12
7.4. Style d'apprentissage et documents adaptatifs.....	13
8. Conclusion.....	15
Chapitre II : Méta-heuristique d'optimisation	
1. Introduction	16

2.	Les méta-heuristiques.....	16
2.1.	Définition	16
2.2.	Objectif des méta-heuristiques	17
2.3.	Concepts généraux	18
2.4.	Organisation générale.....	18
2.5.	Les avantages et les inconvénients.....	19
2.6.	Applications pour les méta-heuristiques	20
3.	Les algorithmes de colonies de fourmis.....	20
3.1.	Definition	20
3.2.	Origine.....	21
3.3.	Un modèle expliquant ce comportement est le suivant.....	22
3.4.	Principe d'un algorithme de colonie de fourmis.....	23
4.	Déroulement global de l'algorithme	24
4.1.	Itérations et déplacement des fourmis	24
4.2.	Cycle de Vie d'une fourmi.....	24
4.3.	Le dépôt de phéromones	24
4.4.	Le moteur de l'algorithme	25
4.5.	Les données du problème.....	25
5.	Conclusion.....	25

Chapitre III : Les Méthodes des classifications

1.	Introduction	26
2.	Les méthodes de la classification hiérarchiques.....	26
2.1.	Les méthodes hiérarchiques ascendantes	27
2.2.	Les méthodes hiérarchiques descendantes	28
3.	Les méthodes de classification non hiérarchique (partition) :.....	28
3.1.	Méthode des k-moyennes (K-means).....	29
3.2.	Algorithme des K-means.....	30
3.3.	Remédiation aux problèmes des K-means	30

4. Conclusion.....	31
Chapitre IV : Conception	
1. Introduction.....	32
2. Les objectifs de notre système	33
3. Architecture générale du système.....	33
3.1. Interface.....	36
3.2. Modèle Apprenant.....	36
3.2.1. Niveau Cognitif.....	37
3.2.2. Style d'apprentissage.....	37
3.3. Modèle Enseignant.....	37
3.3.1. Le cours.....	37
3.4. Générateur de contenu.....	39
3.4.1. Module de profilage	39
3.4.1.1. Les tests d'entrées	40
3.4.2. Module de classification.....	41
3.4.2.1. Comment classer un chemin	42
3.4.3. Générateur de chemin.....	42
4. Structure de la base des données	45
4.1. Le modèle conceptuel des données	45
4.2. La liste des entités	46
4.3. Le modèle logique des données (M.L.D).....	47
5. Conclusion.....	47
Chapitre V : Implémentation	
1. Introduction.....	48
2. Environnement de travail	48
2.1. Langage de programmation.....	48
2.1.1. Php.....	48
2.1.2. JavaScript	49

2.1.3. AJAX.....	49
2.2. Environnement de développement.....	55
2.2.1. EasyPHP.....	50
3. Description des tables de la base de données.....	50
4. Structure de l'application :	51
4.1. Description du fonctionnement et présentation du logiciel	52
4.1.1. Interface administrateur.....	53
4.1.2. Interface apprenant :.....	58
4.1.3. Interface enseignant.....	66
5. Conclusion.....	69
Conclusion Générale :	70
Bibliographie :	72

Liste des Figures

Figure	Titre	Page
1.1	Organisation du schéma de métadonnées LOM.	8
1.2	schéma de Cycle de vie d'une ressource pédagogique.	9
2.1	colonie des fourmis.	21
2.2	Le routage par colonie de fourmis :(Antcolonyrouting) ACR.	22
3.1	hiérarchie ascendant.	27
3.2	clustering k-means	30
4.1	Architecture générale du système.	33
4.2	Schémas détaillé de l'architecture du système.	35
4.3	Structure d'un chapitre.	38
4.4	Représentation hiérarchique d'un cours.	38
4.5	Fonctionnement du processus d'adaptation et de génération.	43
4.6	Le modèle conceptuel de donnée (M.C.D).	45
5.1	Les liens de la page d'accueil.	51
5.2	Interface principale du système (Page D'accueil).	52
5.3	Interface d'accès de l'administrateur.	53
5.4	Interface de l'espace administrateur.	53
5.5	Architecture de l'interface administrateur.	54
5.6	Liste des nouveaux apprenants.	54
5.7	Liste des apprenants acceptés.	55
5.8	Liste des nouveaux enseignants.	55
5.9	Liste des enseignants acceptés.	56
5.10	Distribution des membres	56
5.11	Liste des modules supervisés.	57
5.12	Liste des modules supervisés	57
5.13	Architecture De L'interface Apprenant.	58
5.14	Interface d'accès pour les apprenants.	58
5.15	Formulaire d'inscription.	59
5.16	Interface de l'espace apprenant.	59
5.17	Evaluation psychologique.	60
5.18	Questionnaire pédagogique.	61
5.19	le chemin le plus adapté.	62
5.20	Affichage du cours.	62
5.21	Chemin non adapté.	63

5.22	chemin d'apprentissage poursuivi	64
5.23	Matrice des objets pédagogiques.	64
5.24	Courbe des variations de niveau.	65
5.25	Architecture de l'interface enseignant.	65
5.26	Interface d'accès pour les enseignants.	66
5.27	Interface de l'espace enseignant.	67
5.28	Ajouter un module.	67
5.29	Formulaire d'ajout des objets pédagogiques.	68
5.30	Ajouter des QCM	69

Liste des Tableaux

Tableau	Titre	Page
4.1	La liste des entités	45

Introduction générale

L'évolution permanente des besoins de formation, vers plus d'efficacité, plus de flexibilité et à moindre coût, a favorisé l'émergence d'outils pédagogiques et informatiques dont l'objectif est en quelque sorte de distribuer la formation. Les technologies de l'information et de la communication, Internet en particulier, ont, ces dix dernières années, envahi notre quotidien tant personnel que professionnel. Après s'être immiscé dans de nombreux domaines tels que le E-Learning, Internet est en passe de devenir la clé de voûte d'une nouvelle forme d'enseignement. En effet, les sites de E-Learning se multiplient du fait de l'intérêt qu'ils apportent : gain de temps, économie de transport et d'hébergement, souplesse d'utilisation, interactivité, etc.

Depuis quelques temps, un nouveau courant s'intéresse à l'adaptation des contenus pédagogiques aux apprenants. Il serait donc plus intéressant d'offrir selon les préférences et les capacités des apprenants, des contenus personnalisés et des parcours plus adaptés, afin de maîtriser effectivement la gestion des connaissances à transmettre.

Ce mémoire vise deux objectifs principaux :

- Une adaptation des contenus pédagogiques en se basant sur le profil initial de l'apprenant (niveau pédagogique et style d'apprentissage).
- la spécification, la conception et la réalisation des contenus pédagogiques multimédias adaptatifs utilisent les divers outils pour composer des documents à partir d'un rassemblement d'objets pédagogiques (granules).

Dans ce mémoire, nous nous intéressons à l'adaptation des contenus pédagogiques, en utilisant conjointement deux paradigmes que nous jugeant complémentaires, à savoir

- L'**Optimisation**, on appliquant l'algorithme des colonies de fourmis comme outil de sélection d'objets pédagogiques pour le processus d'adaptation afin de fournir un cours adapté aux connaissances et au profil cognitif de l'apprenant.
- La **Classification** des chemins d'apprentissages considérés comme optimaux pour des recommandations ultérieures aux nouveaux apprenants ayant des profils similaires.

Organisation du mémoire

Ce mémoire est organisé de la manière suivante :

- ❖ Introduction générale qui présente le cadre d'étude, la problématique et les objectifs à atteindre.
- ❖ Les trois premiers chapitres de ce mémoire sont consacrés à l'état de l'art :
 - Dans le premier chapitre, nous faisons une introduction sur les EIAH (Environnement Informatique pour l'Apprentissage Humain), puis nous parlons des concepts d'objets pédagogiques et des styles d'apprentissage.
 - Dans le deuxième chapitre, nous discutons de l'optimisation par les méta-heuristiques et comment appliquées l'algorithme des colonies de fourmis.
 - Le troisième chapitre est consacré aux différentes méthodes de classification.
- ❖ Le quatrième chapitre est consacré à l'architecture de notre système. Cette dernière est structurée autour d'un certains nombres de modules : Profilage, Générateur de chemin (optimisation) et Classification.
- ❖ L'implémentation de l'ensemble de ces composants et modules est développée dans le cinquième chapitre, nous justifions le choix des techniques de réalisation et nous présentons quelques interfaces de notre système.
- ❖ Finalement, on termine par une conclusion générale et des perspectives.

Chapitre 1 : Les objets pédagogiques et les styles d'apprentissage

1. Introduction

Aujourd'hui, le terme d'objet pédagogique (ou Learning Object en anglais) est devenu central sur le terrain des environnements informatiques pour l'apprentissage humain et fait l'objet de nombreux travaux. La personnalisation de l'apprentissage humain remonte aux travaux sur les interactions entre aptitudes et traitements qui prônent l'adaptation de l'instruction aux caractéristiques de l'individu. A cette fin, plusieurs travaux de recherche se sont focalisés sur l'identification des dimensions des différences individuelles. Ces recherches ont mené à la naissance de la théorie du style d'apprentissage.

Dans la suite du chapitre nous allons consacrer une introduction à notre domaine de recherche et qui est les EIAH (Environnement Informatique pour l'Apprentissage Humain). Par la suite, nous allons discuter de deux concepts étroitement liés à notre projet, à savoir : les objets pédagogiques et les styles d'apprentissage.

2. L'Environnement EIAH

Un EIAH au sens large est un environnement qui intègre des agents humains (e.g. élève ou enseignant) et artificiels (i.e., informatiques) et leur offrir des conditions d'interactions, localement ou à travers les réseaux informatiques, ainsi que des conditions d'accès à des ressources formatives (humaines et/ou médiatisées) locales ou distribuées. [George, 01]. La période récente se caractérise par le contexte créé par l'avènement des TIC et le « E-learning » (terme qui ne se limite pas aux travaux de l'enseignement à distance fondés sur les TIC, mais dénote également les dimensions d'ouverture et de distribution introduites par ces technologies), qui conduisent à de nouveaux types de systèmes et renouvelle certaines des problématiques classiques du domaine. L'avènement des TIC conduit d'une part à différents travaux centrés sur la notion de ressource pédagogique : conception industrialisée de supports pédagogiques numériques, indexation et normalisation des objets pédagogiques, diffusion de ressources via des plateformes spécialisées. Ces problématiques conduisent notamment à utiliser en EIAH les travaux sur la notion d'ontologie ou les technologies de type XML.

Dans un EIAH, la machine peut donc avoir différents rôles non mutuellement exclusifs et peut être : outil de présentation de l'information (e.g. un hypermédia ou une plateforme Web spécialisée), outil de traitement de connaissances (e.g. un système à base de connaissances résolvant les exercices avec l'élève ou un module pilotant l'interaction) ou outil de communication entre l'homme et la machine ou entre les hommes à travers les machines.

3. Définitions des concepts

3.1. Objets pédagogiques

La définition du concept d'objet pédagogique diffère selon les auteurs. Selon le groupe de travail de l'IEEE (Institute of Electrical and Electronics Engineers) connus sous le nom LTSC (*Learning Technologies Standards Committee*), auteur du LOM (*Learning Object Metadata*) [IEEE, 05], le concept d'objet pédagogique se définit comme une entité sur support informatique ou non, qui peut être utilisé, réutilisé ou référencé dans une activité de formation assistée à partir d'un support. Cette définition restreint les objets pédagogiques aux entités utilisées uniquement dans le cadre des formations qui font usage des supports technologiques.

Pernin [Pernin, 04], allant dans le même ordre d'idée que le groupe IEEE définit un objet pédagogique comme toute entité numérique ou non, abstraite ou concrète, qui peut être utilisé, réutilisé ou référencé lors d'une formation. Cette définition apporte un élément de plus à la précédente. Cette définition prend en compte la nature de la ressource (concrète ou abstraite), et tout type de formation (traditionnel et à support technologique).

Hodgins quant à lui, définit un objet pédagogique comme suit: "*Learning Object represents a completely new conceptual model for the mass of content used in the context of learning.*" [Hodgins 00]. Pour lui, un objet pédagogique est un modèle conceptuel nouveau pour la grande masse de donnée utilisée dans un contexte d'apprentissage.

3.2. Ressources pédagogiques

Gomez [Gomez, 05], définit une ressource pédagogique comme étant toute entité (numérique ou non), utilisée dans un contexte d'enseignement, de formation ou d'apprentissage et qui est durable (doit pouvoir résister aux changements technologiques), adaptable (doit permettre une modulation sur mesure c'est-à-dire tenant en compte les

spécificités physiques des apprenants), gérable (doit pouvoir être administrée facilement), fiable (doit répondre aux exigences pour lesquelles on l'a créée), abordable (doit pouvoir être accessible), évaluable (doit pouvoir être estimée facilement), interopérable (permettre l'utilisation de contenus et composants développés par une organisation sur une plate-forme donnée par d'autres organisations sur d'autres plates-formes), retrouvable (doit pouvoir être accessible facilement), réutilisable (doit pouvoir être utilisé dans des contextes et des buts différents), indexable (doit être décrite par des métadonnées ou des annotations sémantiques).

Pour Gérard Puimatto [Gérard, 04] la notion de ressource pédagogique (ou éducative) est fortement liée à celle de document, et aux approches documentaires. Dans le domaine du multimédia, les « ressources multimédias pour l'éducation » constituent l'ensemble des informations, documents, logiciels, programmes, banques de données, etc. qui permettent de véhiculer, de transmettre ou d'appréhender des concepts et contenus d'enseignements.

Ainsi d'après toutes ces définitions, nous remarquons que les concepts d'objets pédagogiques et de ressources pédagogiques sont fortement liés et prêtent même à confusion. C'est ainsi que d'après la classification que fait Pernin [Pernin, 04] des objets pédagogiques à savoir qu'il existe trois principales classes d'objets pédagogiques :

- Les Unités d'Apprentissage (UA), qui permettent de structurer la formation et de l'organiser dans l'espace et dans le temps ;
- Les Activités Pédagogiques (AP), qui définissent les modalités précises d'acquisition, de validation, de communication d'une ou de plusieurs connaissances ;
- Les ressources pédagogiques, physiques ou numériques, nécessaires à la réalisation des activités ;

Nous remarquons que les ressources pédagogiques sont une classe ou spécificité d'objets pédagogiques.

4. Propriétés des objets pédagogiques

Dans cette partie, il sera question pour nous de donner les propriétés qui régissent les objets pédagogiques et d'identifier les différents types d'objets pédagogiques.

Les objets pédagogiques possèdent un certain nombre de propriétés qui les caractérisent [Bourda, 2001]. Parmi elles, nous avons :

- **Granularité (taille)** : un objet pédagogique élémentaire est une composante d'un cours dont la durée de consultation ou d'exécution se situe entre 2 et 15 minutes.
- **Autonomie** : un objet pédagogique peut être utilisé indépendamment des autres. (grain de base : texte, image, son)
- **Réutilisabilité** et **adaptabilité** : un objet pédagogique élémentaire peut être utilisé à différentes fins, dans différentes applications, dans différents produits, dans différents contextes et par différents modes d'accès.
- **Agrégabilité**: les objets pédagogiques élémentaires peuvent être regroupés dans des ensembles y compris pour des cours traditionnels.
- **Indexation** : chaque objet pédagogique doit être muni d'une description permettant de le retrouver facilement.

A ces propriétés s'ajoutent les propriétés suivantes évoquées dans la définition de Gomez des ressources pédagogiques (valables pour l'ensemble des objets pédagogiques) :

- **Accessibilité** : chaque objet pédagogique est muni d'une description permettant de le retrouver, de l'identifier et d'y accéder facilement. Un ensemble de descripteurs connu sous le nom de **LOM** (Learning Object Metadata) permet d'indexer les objets pédagogiques.
- **Durabilité** : un objet pédagogique doit pouvoir affronter les changements technologiques en minimisant la réingénierie ou le redéveloppement.
- **Interopérabilité** : un objet pédagogique doit pouvoir être utilisé dans des environnements technologiques différents.

Les objets pédagogiques, en particulier les ressources pédagogiques permettent un apprentissage sans superflu (si un utilisateur a besoin d'une fraction de cours par exemple, il pourra se limiter aux objets pédagogiques correspondants à cette fraction de cours), au moment voulu (étant donné que les objets pédagogiques sont indexés, ils peuvent être retrouvés facilement), sur mesure (les objets pédagogiques permettent une personnalisation des cours à l'échelle d'une organisation ou de chaque personne).

5. Description des objets pédagogiques

Dans le but de faciliter l'accessibilité, la recherche, le partage ou la réutilisation, chaque objet pédagogique doit être décrit à l'aide de métadonnées et d'annotation sémantiques [Iles et al, 2008]. En effet, pour qu'un objet pédagogique puisse être retrouvé ou alors réutilisé, il doit être indexé ou alors annoté. La description d'une ressource pédagogique peut aussi bien être basée sur des métadonnées ou alors sur des représentations sémantiques. Dans les sections suivantes, nous allons aborder ces différentes notions avec plus de détail.

5.1. Métadonnées

Les métadonnées sont définies comme étant : « *des données relatives à d'autres données* » [Iles et al, 08]. C'est-à-dire que d'après Élisabeth Noël les métadonnées sont des données qui renseignent sur la nature de certaines autres données et qui permettent ainsi leur utilisation pertinente. Dans le domaine de l'apprentissage automatique, les métadonnées permettent de décrire les documents pédagogiques afin de les rendre plus accessibles et plus manipulables [Iles et al, 2008]. De telles métadonnées sont structurées par catégories ou champs sémantiques représentant une caractéristique particulière sur la ressource comme son titre par exemple [Bourda, 2001] selon le modèle de description utilisé. La suite de cette section présente les métadonnées de type LOM (Learning Object Meta data).

5.2. Modèles de description des objets pédagogiques

Un ensemble de modèles ou langages, reposant sur la notion d'objet pédagogique (tels que LOM, SCORM ou EML), a été mis au point au cours des cinq dernières années.

5.2.1. Learning Object Meta data (LOM)

Le 12 juin 2002, l'IEEE (Institute of Electrical and Electronics Engineers) a adopté le premier standard en matière de technologies éducatives, qui concerne la description des objets pédagogiques, et qui est plus connu sur le nom de LOM [IEEE, 02]. En effet le LOM est un ensemble de descripteurs permettant d'indexer les objets pédagogiques. La mise à disposition des objets pédagogiques n'est pas suffisante pour garantir leur réutilisation. Elle doit s'accompagner de moyens de recherche et d'accès à l'information utile. La description LOM qui peut être associée à chaque objet pédagogique constitue une première description indispensable à cet accès.

Le standard LOM (version 6) spécifie la syntaxe et la sémantique des métadonnées pédagogiques et définit les attributs nécessaires pour une description adéquate et complète des ressources pédagogiques. Il existe 78 attributs regroupés en neuf catégories [Bourda, 2001] :

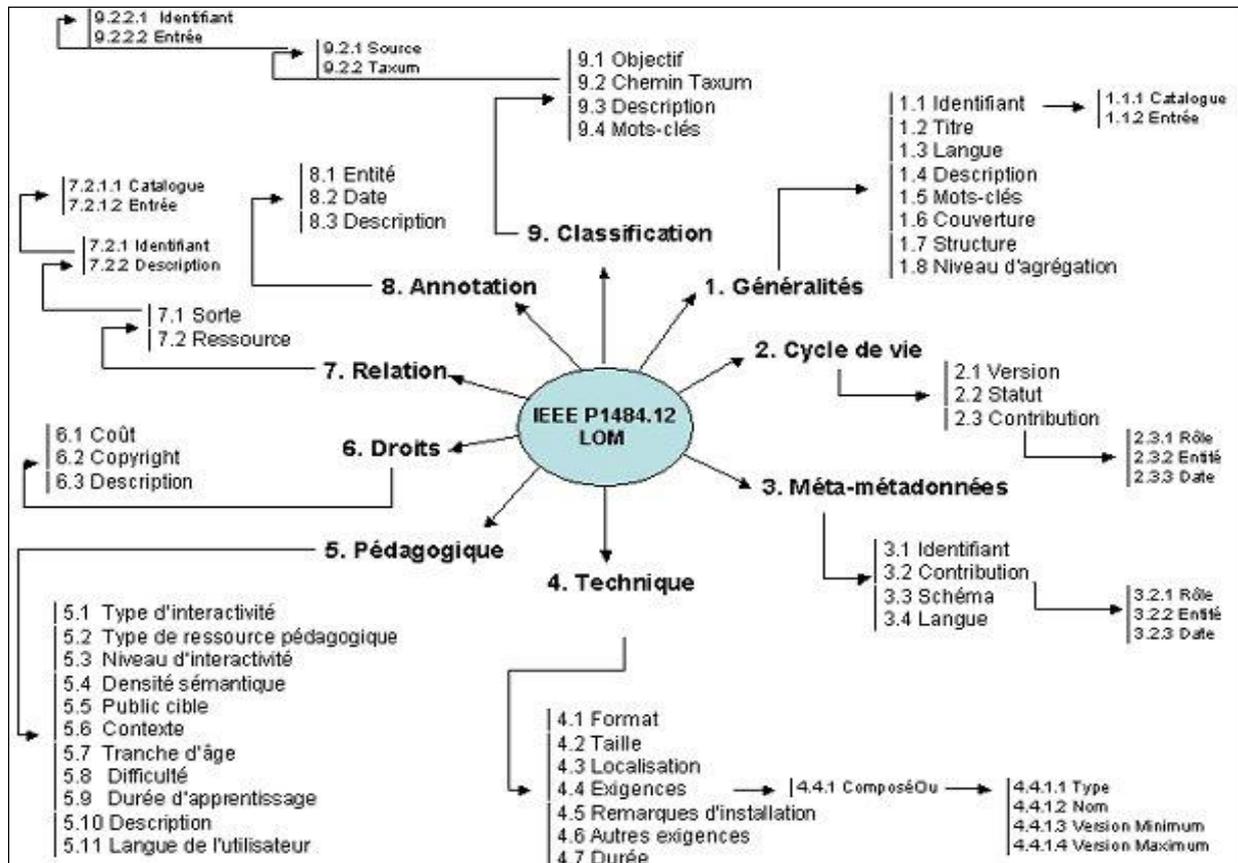


Figure 1.1 : Organisation du schéma de métadonnées LOM [De Bra., 04]

Les caractéristiques de LOM ne sont pas toutes nécessaires pour une application donnée. Un profil d'application permet d'en limiter le nombre. Un profil d'application est un assemblage de métadonnées adaptées au contexte d'utilisation spécifique. Ce profil peut soit restreindre les descripteurs de LOM, soit les enrichir (Celebrate, Normetic, OpenCartable, ManUeL).

5.3. Cycle de vie d'une ressource pédagogique

Ce schéma résume le cycle de vie :

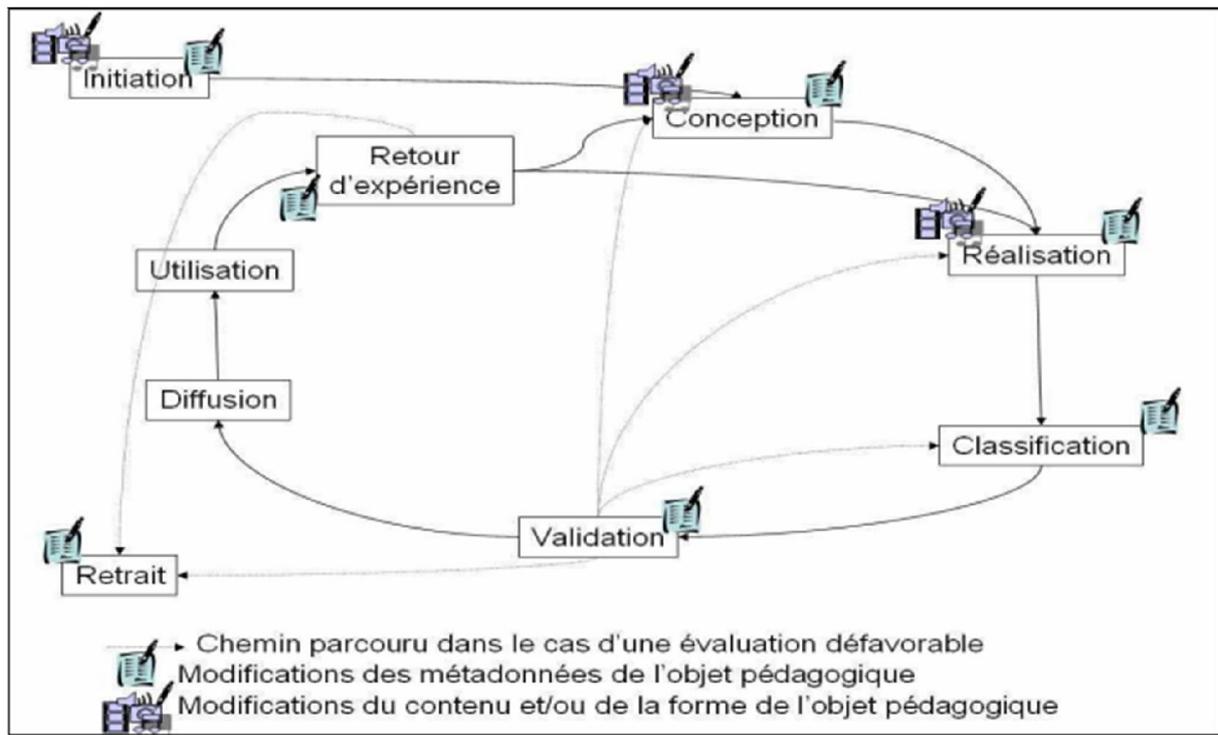


Figure1.2 : schéma de Cycle de vie d'une ressource pédagogique [Alenou ,09]

6. Différents supports de présentation

Les objets pédagogiques peuvent se présenter sous différentes formes :

- Textes : fichiers au format PDF, RTF, TXT, DOC, etc.
- Images : par exemple des photos, des cartes, des schémas, images pouvant être animées (GIF animé) ou cliquables dans un environnement HTML.
- Hypertextes : il s'agit des pages web, de la simple page de consignes pour l'étudiant au site pédagogique fournissant un contenu complet jouant le rôle de support de cours
- Présentations assistées par ordinateur
- Emissions sonores, vidéos
- multimédias : cours ou parties de cours présentés sous une forme multimédia combinant les diverses formes (texte, son, image, vidéo) et faisant appel aux (TIC).
- animations : les documents de type hypertexte peuvent être enrichis par des animations intégrées dans les pages web (animation Flash, applet Java, code Java script, SMIL, HTML+TIME). Ces animations peuvent ainsi donner une dimension

temporelle aux pages HTML traditionnellement statiques ou encore se présenter sous la forme d'applications exécutables de manière indépendante.

7. Style d'apprentissage

Les recherches dans le domaine de l'enseignement nous démontrent qu'on a tendance à enseigner en se basant sur notre propre style d'apprentissage. Or, si notre apprenant n'a pas le même style d'apprentissage que nous, il y aura des difficultés à l'horizon. Il serait donc pertinent pour tout tuteur de se familiariser avec les différents styles d'apprentissage pour la simple et bonne raison que cela les aidera à devenir des tuteurs plus efficaces

7.1. Qu'est-ce qu'un style d'apprentissage

Comme nous l'avons vu plus haut, le style, c'est la « manière personnelle d'agir et de se comporter... » (ROBEAT) Par extension, le style d'apprentissage d'un individu, c'est son mode personnel de saisie et de traitement de l'information. En pratique, et en d'autres termes, le style d'apprentissage c'est donc la manière préférentielle d'aborder et de résoudre un problème. Cette définition, délibérément succincte, appelle quelques commentaires.

- a. **Enseigner n'est pas synonyme d'apprendre**, les styles d'enseignement désignent des modalités de la communication didactique ; les styles d'apprentissage sont des modalités de résolution de problèmes.
- b. **Le style d'apprentissage et le style cognitif sont des concepts distincts**, même s'ils sont souvent confondus. Pour les puristes, le style cognitif est inné et stable tandis que le style d'apprentissage résulte de l'inné et de l'acquis et peut donc évoluer par l'expérience.
- c. **Les styles d'apprentissage ne sont pas l'expression d'une typologie rigide** qui prétendre classer les individus en catégories strictes (à l'instar des typologies classiques). En fait, ils ne reflètent qu'un aspect particulier de la complexité des personnes.
- d. **On ne peut hiérarchiser les styles d'apprentissage**. Tout comme dans le domaine sportif, il n'existe pas de relation univoque entre le style et la qualité de la performance.

Il n'existe donc pas une bonne façon d'apprendre ou de résoudre un problème. Nous sommes tous différents, mais complémentaires.

7.2. Différentes approches et modèles

Les travaux sur le style d'apprentissage ont donné de multiples méthodes et instruments permettant de catégoriser les apprenants en fonction de leurs différences :

- Learning Styles Theory [Kolb, 85], Index of Learning Styles [Felder et al, 1988],
- Learning Styles [Honey et Mumford, 92], Student Learning Style Scales [Grasha, 96],
- Multiple Intelligences [Gardner, 99], Auditory Visual Tactile Learning Styles [Sarasin, 98].

Riding a esquissé une classification des principaux modèles de styles d'apprentissage [Riding et al, 01] qu'il a classé en trois groupes :

- Modèles axés sur le *processus d'apprentissage*,
- Modèles axés sur les *préférences pédagogiques*,
- Modèles axés sur les *habiletés cognitives* ainsi que les *stratégies d'apprentissage*.

Riding [Riding et al, 01] soutient que l'individu développe des stratégies d'apprentissage qui reflètent les processus sous-jacents à son style. Le modèle qu'il propose se fonde sur les habiletés cognitives et représente l'approche préférée et habituelle d'un individu pour organiser et se représenter l'information. Ce modèle est composé uniquement de deux dimensions :

- *wholist - analytic*: modélise la tendance d'un individu à organiser l'information comme un tout ou bien comme un ensemble de parties,
- *verbaliser - imager* : modélise la tendance d'un individu à se représenter l'information pendant la phase de réflexion sous forme verbale ou bien sous forme d'images mentales.

Felder R. considère que l'ensemble des différences individuelles qui comprennent les préférences personnelles indiquées pour l'apprentissage et certaines activités pédagogiques, mais aussi les différences intellectuelles et psychologiques individuelles, font référence au style d'apprentissage d'un individu [Felder et al., 88]. Pour lui, le style d'apprentissage d'un élève doit être considéré en termes de présentation, organisation, traitement et assimilation d'information. Il propose un modèle composé de quatre dimensions en émettant l'hypothèse

que le style d'apprentissage d'un élève peut être défini en répondant aux quatre questions suivantes [Felder, 93] :

- **déductif - inductif** : avec quelle organisation de l'information l'élève est-il le plus à l'aise ?
- **visuel - verbal** : à travers quelle modalité sensorielle l'information est-elle réellement perçue ?
- **séquentiel - global** : comment l'élève progresse-t-il dans l'activité de compréhension ?
- **actif - réfléchi** : comment l'élève préfère-t-il traiter l'information ?

Felder propose un questionnaire composé de 44 questions pour évaluer la position d'un élève sur les quatre dimensions.

7.3. Style d'apprentissage et performances de l'apprenant :

Depuis l'apparition des modèles sur le style d'apprentissage, plusieurs études ont été menées afin de montrer si l'adéquation de la séquence pédagogique au style d'apprentissage améliorerait la performance de l'apprenant.

Les résultats semblent être mitigés, mais la majorité des auteurs s'accordent à dire que la prise en compte du style d'apprentissage dans la conception d'une séquence pédagogique ne peut qu'être un facteur positif pour le soutien de l'élève dans l'activité qui lui est proposée. Par exemple, on a constaté que l'intérêt porté par un élève pour suivre une activité pédagogique, augmente si cette dernière était en accord avec son style d'apprentissage notamment en incorporant dans les contenus des composants multimédia. De même, la dernière version d'ELM-ART, système adaptatif d'apprentissage par le Web du langage de programmation LISP [Brusilovsky, 04] a montré que l'adaptation de la navigation avait une incidence directe sur la motivation des élèves.

Néanmoins, Triantafillou [Triantafillou et al., 03] précise que la majorité des élèves, s'ils apprécient d'un côté que le système cherche à adapter la séquence pédagogique à leur style d'apprentissage, aiment malgré tout d'un autre côté pouvoir garder la maîtrise sur le fonctionnement du système. Ils apprécient de pouvoir sélectionner un lien de navigation même si le système ne le conseille pas.

Riding [Riding et al., 01] a montré que dans certaines situations, la performance d'un élève était affectée par une interaction entre son style d'apprentissage et les facteurs suivants :

- la structure utilisée pour le matériel pédagogique (position du titre, présence de sous-titre, ajout de présentations générales et leur position),
- son mode de présentation (texte, image, multimédia),
- son type de contenu (concret, abstrait).

Une autre étude [Parkinson et al. 02] conduit à la conclusion que les AIWBES ont un énorme potentiel à offrir aux élèves sur le plan pédagogique lorsque leur conception prend en compte le style d'apprentissage d'un individu. Certains auteurs pensent que la motivation d'un élève pour suivre une séquence pédagogique peut être augmentée si elle est conçue dans le but de répondre à plusieurs styles d'apprentissage.

Néanmoins, la prise en compte du style d'apprentissage d'un individu dans le processus de formation n'est pas le seul facteur à considérer. Il a été aussi mis en évidence que le type de navigation, contrainte ou conseillée devait être corrélée au niveau de connaissances de l'élève pour obtenir une performance maximale dans la phase d'apprentissage. Un élève avec aucun pré-acquis préférera une séquence dirigée, alors qu'à l'inverse un élève avec certaines connaissances sur le domaine aura tendance à vouloir naviguer librement dans les contenus pédagogiques.

7.4. Style d'apprentissage et documents adaptatifs

L'adaptation aux styles d'apprentissage est une tâche qui doit être prise en compte lors de la phase de conception des séquences pédagogiques ainsi que lors de leur enchaînement durant la phase d'enseignement à proprement dite. On distingue donc la création de contenus multimédia adaptatifs et leur séquençement et présentation adaptatifs. Plusieurs travaux ont porté sur ces deux aspects séparément. Cependant, il y en a peu qui traite le problème d'une manière globale.

Hui Min Lee [Hui Min Lee et al., 04] conclut dans son étude que la qualité de l'enseignement se trouve améliorée lorsque l'auteur de la séquence pédagogique cherche à exploiter les différentes possibilités offertes par les supports hypermédias (séquences vidéo, audio,...) afin de contenter les caractéristiques et préférences individuelles des élèves.

Mayer [Mayer, 03] a mené plusieurs expériences qui ont montré par exemple l'importance que pouvait revêtir la position dans la page d'une légende associée à une image. Plus la distance qui les sépare est grande, plus il sera difficile pour l'élève de comprendre le contenu. De même, il a montré que l'ajout d'éléments multimédia non pertinent dans une page produisait un effet négatif sur l'apprentissage de l'élève en perturbant son niveau de concentration.

La majeure partie des travaux concernant les documents multimédia adaptables aux styles d'apprentissage porte sur les médias à utiliser lors de l'adaptation pour la présentation. Très peu se consacre à la relation entre le style et la nature de l'activité et des tâches d'apprentissage. Moallem [Moallem, 01] précise que la conception du cours doit aussi être réalisée en fonction des pédagogies que l'auteur vise et pas seulement sur des considérations liées aux médias utilisés comme support de l'information.

La tâche de créer des contenus adaptatifs est très délicate et fastidieuse du fait de la complexité des critères à prendre en considération. Les règles d'adaptation doivent être spécifiées pour chaque utilisation de chaque composant pédagogique, pour chaque situation particulière. Pour aider les auteurs dans leur entreprise, certaines recherches prônent la création de logiciels auteurs [Creastea, 04 a] tel que le système MOT (Multimedia Object Transfe) qui offre des fonctions d'assemblage de composants pédagogiques en précisant pour chacun la stratégie d'adaptation à utiliser. L'adaptation peut être programmée par l'auteur du cours grâce à un langage d'adaptation intégré au système. De même, eCAD est un système d'aide à l'analyse et la conception [Timothy et al., 04] d'un cours formulé sur la base d'un ensemble d'objectifs pédagogiques respectant la taxinomie de Bloom (est un psychologue américain spécialisé en pédagogie. Il était également professeur, chercheur, éditeur littéraire et examinateur en éducation. Il est surtout connu pour ses importantes contributions au classement des objectifs pédagogiques et pour sa taxonomie de Bloom, utile pour évaluer la progression de l'apprentissage.).

Les systèmes adoptant les styles d'apprentissage dans les deux phases de conception et enseignement sont rares. Dans les auteurs présentent un modèle de document à trois couches qui offrent une adaptation de la navigation ainsi que de la présentation en considérant le style d'apprentissage. Castillo [Castillo et al., 03] montre un début d'intérêt pour les réseaux bayésiens dans la modélisation de prédiction de tâches, en se basant sur les résultats des travaux de Felder [Felder, 93].

8. Conclusion

Nous avons consacré la première partie de ce chapitre à la notion d'objet pédagogique. La puissance du concept de l'objet, réside dans sa modularité et dans la réutilisation de ses éléments constitutifs. Par la suite, nous avons introduit les styles d'apprentissage, leurs modèles et approches.

Chapitre 2 : Méta-heuristique d'optimisation

1. Introduction

L'optimisation reste toujours un sujet très important dans le domaine d'informatique, elle vise à déterminer une solution optimale, prenant en compte un ensemble de contraintes et de variables inhérentes au problème. Dans la pratique, il n'existe pas de solution optimale parce que la représentation mathématique d'un problème n'est jamais exacte. Il faut donc déterminer de bonnes solutions en se basant sur de bons modèles qui se doivent d'être fiables et robustes.

La solution optimale à un problème d'optimisation ne peut que très rarement être déterminée en un temps polynomial. Il est donc souvent nécessaire de trouver des modes de résolution qui fournissent une solution de bonne qualité dans un laps de temps raisonnable : c'est ce que font les heuristiques. Depuis une vingtaine d'années, les heuristiques les plus populaires, et également les plus efficaces, sont des techniques générales, appelées méta-heuristiques, qu'il s'agit d'adapter à chaque problème particulier. [Kirkpatrick et al., 83]

2. Les méta-heuristiques :

Les méta-heuristiques sont apparues dans les années 1980 et forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels on ne connaît pas de méthode classique plus efficace. Elles sont généralement utilisées comme des méthodes génériques pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé.

2.1. Définition

- **L'heuristique** : est une méthode, une technique ou un critère de guidage ou de décision, en général empirique ou obtenu par approximation, permettant de choisir la voie la plus prometteuse de recherche de la solution au problème posé, ou d'éliminer les voies les moins intéressantes, sans garantie sur la validité ou la précision de l'information ainsi fournie [Bouri, 08].

- **Les méta-heuristiques** : Etymologiquement parlant de ce mot est composé dans un premier temps du préfix méta qui signifie « au-delà » ou « plus haut » en grec puis de « heuristique » qui signifie « trouver ». Cette décomposition permet de facilement comprendre le but premier de ces algorithmes : trouver des solutions à des problèmes en utilisant plusieurs (métas) heuristiques [Bouri, 08].

Les méta-heuristiques sont souvent inspirées des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essais particuliers). [Bouri, 08].

2.2. Objectif des méta-heuristiques

Une méta-heuristique a pour but de résoudre un problème d'optimisation donné : elle cherche un objet mathématique (une permutation, un vecteur, etc.) minimisant (ou maximisant) une fonction objectif, qui décrit la qualité d'une solution au problème. L'ensemble des solutions possibles forme l'espace de recherche. L'espace de recherche est au minimum borné, mais peut être également limité par un ensemble de contraintes.

Les méta-heuristiques manipulent une ou plusieurs solutions, à la recherche de l'optimum, la meilleure solution au problème. Les itérations successives doivent permettre de passer d'une solution de mauvaise qualité à la solution optimale. L'algorithme s'arrête après avoir atteint un critère d'arrêt, consistant généralement en l'atteinte du temps d'exécution imparti ou en une précision demandée. Une solution ou un ensemble de solutions est parfois appelé un état, que la méta-heuristique fait évoluer via des transitions ou des mouvements. Si une nouvelle solution est construite à partir d'une solution existante, elle est sa voisine. Le choix du voisinage et de la structure de donnée le représentant peut être crucial.

Lorsqu'une solution est associée à une seule valeur, on parle de problème mono-objectif, lorsqu'elle est associée à plusieurs valeurs, de problème multi-objectifs (ou multicritères). Dans ce dernier cas, on recherche un ensemble de solutions non dominées (le « front de Pareto »), solutions parmi lesquelles on ne peut décider si une solution est meilleure qu'une autre, aucune n'étant systématiquement inférieure aux autres sur tous les objectifs.

2.3. Concepts généraux :

En fait, on peut citer plusieurs concepts liés aux méta-heuristiques :

- Les méta-heuristiques ne nécessitent pas de connaissances particulières sur le problème optimisé pour fonctionner, le fait de pouvoir associer une (ou plusieurs) valeurs à une solution est la seule information nécessaire.
- En pratique, elles ne devraient être utilisées que sur des problèmes ne pouvant être optimisés par des méthodes mathématiques. Utilisées en lieu et place d'heuristiques spécialisées, elles montrent généralement de moins bonnes performances.
- Les méta-heuristiques sont souvent employées en optimisation combinatoire, mais on en rencontre également pour des problèmes continus ou mixtes (problèmes à variables discrètes et continues).
- Certaines méta-heuristiques sont théoriquement « convergentes » sous certaines conditions. Il est alors garanti que l'optimum global sera trouvé en un temps fini, la principale condition de convergence est de considérer que l'algorithme est ergodique (qu'il peut atteindre n'importe quelle solution à chaque mouvement), mais on se satisfait souvent d'une quasi-ergodicité (si la méta-heuristique peut atteindre n'importe quelle solution en un nombre fini de mouvements).

2.4. Organisation générale

D'une manière générale, les méta-heuristiques s'articulent autour de trois notions :

- **Diversification /exploration** : désigne les processus visant à récolter de l'information sur le problème optimisé [Siarry, 03].
- **L'intensification/exploitation** : vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche [Siarry, 03].
- **La mémoire** : est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi les optimums locaux [Siarry, 03].

Les méta-heuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

2.5. Les avantages et les inconvénients

Les méta-heuristiques étant très généralistes, elles peuvent être adaptées à tout type de problème d'optimisation pouvant se réduire à une « boîte noire ». Elles sont souvent moins puissantes que des méthodes exactes sur certains types de problèmes. Elles ne garantissent pas non plus la découverte de l'optimum global en un temps fini.

Cependant, un grand nombre de problèmes réels ne peuvent être optimisé de manière efficace par des approches purement mathématiques, les méta-heuristiques peuvent alors être utilisées avec profit.

La notion d'efficacité se rapporte généralement à deux objectifs contradictoires : la vitesse et la précision. La vitesse est souvent mesurée en nombre d'évaluations de la fonction objectif, qui est la plupart du temps la partie la plus gourmande en temps de calcul. La précision se rapporte à la distance entre l'optimum trouvé par la méta-heuristique et l'optimum réel, soit du point de vue de la solution, soit de celui de la valeur. Bien souvent, un algorithme rapide est peu précis, et inversement. [Roumane, 12]

2.6. Applications pour les méta-heuristiques

- Conception de réseaux de télécommunications.
- Mise en œuvre de réseaux de télécommunications.
- Bio-informatique : alignement multiple, recherche de chaîne médiane, ...
- Construction d'emploi du temps.
- Tournées de véhicules (Vehicle Routing Problem).
- Problème du voyageur de commerce (TSP).
- Problème de k-coloriage de graphe [Galinier, 99].
-

3. Les algorithmes de colonies de fourmis

3.1. Définition

- **Les fourmis**

Les fourmis sont capables de résoudre collectivement des problèmes complexes, comme trouver le plus court chemin entre deux points dans un environnement accidenté. [Solnon, 00] Pour cela, elles communiquent entre elles de façon indirect, grâce à une hormone volatile, appelée phéromone.

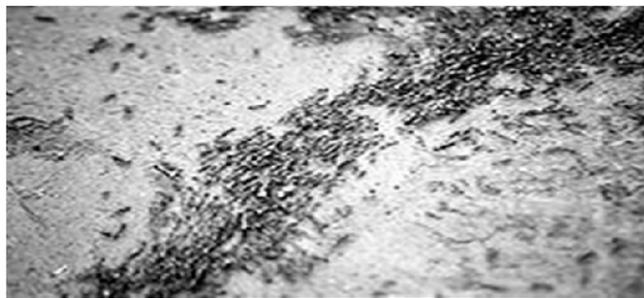


Figure 2.1 : colonie des fourmis [1]

- **Les algorithmes de colonies de fourmis** : sont des algorithmes inspirés du comportement des fourmis et qui constituent une famille de méta-heuristiques d'optimisation. [1]

3.2. Origine

Initialement proposé par Marco Dorigo et al. Dans les années 1990, pour la recherche de chemins optimaux dans un graphe, le premier algorithme s'inspire du comportement des fourmis recherchant un chemin entre leur colonie et une source de nourriture. Il existe cependant plusieurs familles de méthodes s'inspirant du comportement des fourmis, ces différentes approches sont regroupées sous les termes : algorithmes de colonies de fourmis, optimisation par colonies de fourmis, fourmis artificielles, ou diverses combinaisons de ces variantes [1].

L'idée originale provient de l'observation de l'exploitation des ressources alimentaires chez les fourmis. En effet, celles-ci, bien qu'ayant individuellement des capacités cognitives

limitées, sont capables collectivement de trouver le chemin le plus court entre une source de nourriture et leur nid [1].

Des biologistes ont ainsi observé, dans une série d'expériences menées à partir de 1989, qu'une colonie de fourmis ayant le choix entre deux chemins d'inégale longueur menant à une source de nourriture avait tendance à utiliser le chemin le plus court [1].

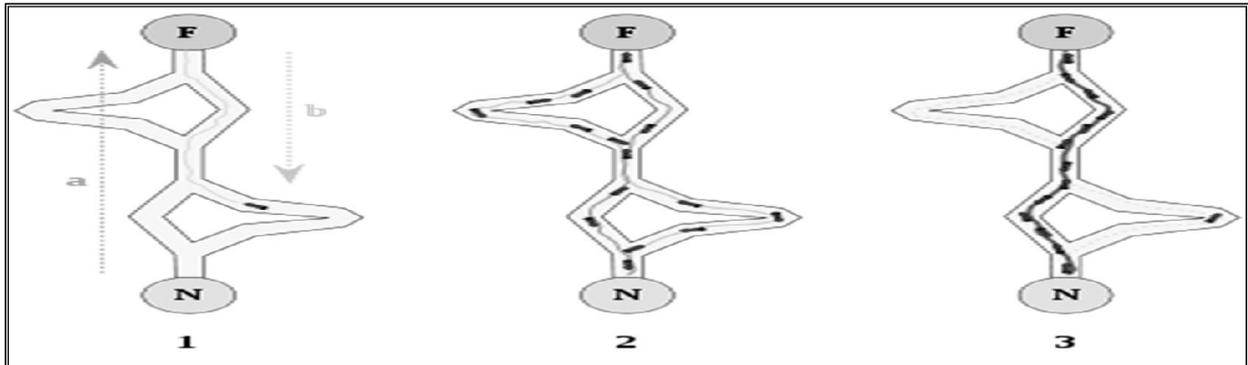


Figure 2.2 : Le routage par colonie de fourmis : (Ant colony routing) ACR [1]

1. la première fourmi trouve la source de nourriture (**F**), via un chemin quelconque (**a**), puis revient au nid (**N**) en laissant derrière elle une piste de phéromone (**b**).
2. les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement de la piste rend plus attractif le chemin le plus court.
3. les fourmis empruntent le chemin le plus court, les portions longues des autres chemins perdent leur piste de phéromones.

3.3. Un modèle expliquant ce comportement est le suivant

- une fourmi (appelée « éclaireuse ») parcourt plus ou moins au hasard l'environnement autour de la colonie.
- si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones.
- ces phéromones étant attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe, cette piste.
- en revenant au nid, ces mêmes fourmis vont renforcer la piste.

- si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps, parcourue par plus de fourmis que la longue piste.
- la piste courte sera donc de plus en plus renforcée, et donc de plus en plus attractive.
- la longue piste, elle, finira par disparaître, les phéromones étant volatiles.
- à terme, l'ensemble des fourmis a donc déterminé et « choisi » la piste la plus courte [1]

Ce système repose sur des rétroactions positives (le dépôt de phéromone attire d'autres fourmis qui vont la renforcer à leur tour) et négatives (la dissipation de la piste par évaporation empêche le système de s'emballer). Théoriquement, si la quantité de phéromone restait identique au cours du temps sur toutes les branches, aucune piste ne serait choisie. Or, du fait des rétroactions, une faible variation sur une branche va être amplifiée et permettre alors le choix d'une branche. L'algorithme va permettre de passer d'un état instable où aucune branche n'est plus marquée qu'une autre, vers un état stable où l'itinéraire est formé des « meilleures » branches.

3.4. Principe d'un algorithme de colonie de fourmis

Un algorithme de colonies de fourmis est un algorithme itératif à population où tous les individus partagent un savoir commun qui leur permet de guider leurs futurs choix et d'indiquer aux autres individus des directions à suivre ou au contraire à éviter. Fortement inspiré du déplacement des groupes de fourmis, cette méthode a pour but de construire les meilleures solutions à partir des éléments qui ont été explorés par d'autres individus. Chaque fois qu'un individu découvre une solution au problème, bonne ou mauvaise, il enrichit la connaissance collective de la colonie. Ainsi, chaque fois qu'un nouvel individu aura à faire des choix, il pourra s'appuyer sur la connaissance collective pour pondérer ses choix. [Schwartz, 08]

Pour reprendre la dénomination naturelle, les individus sont des fourmis qui vont se déplacer à la recherche de solutions et qui vont sécréter des phéromones pour indiquer à leurs congénères si un chemin est intéressant ou non. Si un chemin se retrouve fortement phéromonné, cela signifiera que beaucoup de fourmis l'ont jugé comme faisant partie d'une solution intéressante et que les fourmis suivantes devront la considérer avec intérêt [Schwartz, 08]

Un risque apparaît lorsqu'un chemin non optimal est marqué. En effet, les fourmis qui s'en trouveront à proximité seront tentées d'y passer augmentant encore le niveau de phéromone de ce chemin. Pour diminuer le risque d'enfoncer la colonie dans un minimum local du problème, on pourra prendre soin de diminuer automatiquement le niveau de phéromone de tout le système, pour rehausser l'intérêt des autres chemins qui pourraient faire partie de la solution optimale. Ce paramètre, correspondant au taux d'évaporation des phéromones, est l'un des paramètres principaux de l'algorithme [Schwartz, 08].

De la même manière, aucun chemin ne devra être inondé de phéromones et aucun chemin ne devra être totalement invisible, on pourra donc aussi contrôler le niveau de phéromone de chaque chemin pour le maintenir entre des bornes minimum et maximum. Un chemin inondé de phéromones masquerait tous les autres à proximité et un chemin pas du tout phéromonné ne serait jamais choisi par une fourmi, en conséquence nous devons conserver ces chemins avec des valeurs raisonnables. Ces bornes min et max sont aussi des paramètres de l'algorithme [Schwartz, 08].

4. Déroulement global de l'algorithme

4.1. Itérations et déplacement des fourmis

Nous n'avons pas de notion de générations de population, les itérations de l'algorithme correspondent aux déplacements des fourmis. Pour aller d'un nœud du graphe à un autre, chaque fourmi aura besoin d'un nombre d'itérations dépendant de la taille de l'arc de graphe à parcourir. Ce mode d'itérations va aussi privilégier les plus courts chemins puisque les fourmis auront besoin de moins d'itérations pour en arriver au bout [Schwartz, 08]

4.2. Cycle de Vie d'une fourmi

Chaque fourmi doit connaître la liste des nœuds qu'elle a déjà parcourus et les nœuds encore à parcourir. De plus elle doit mesurer le temps qu'elle passe sur la solution qu'elle explore. À chaque nœud, la fourmi va étudier les arcs possibles en observant leurs niveaux de phéromone respectifs. Elle n'a ensuite qu'à choisir au hasard, en privilégiant les arcs fortement phéromonnés. Une fois arrivée à destination, la fourmi connaît la longueur totale de la solution qu'elle a trouvée, elle peut refaire le chemin en sens inverse pour marquer le chemin avec ses phéromones et enrichir la connaissance collective de la colonie [Schwartz, 08].

4.3. Le dépôt de phéromones

L'heuristique de dépôt de phéromones peut grandement modifier le mode de convergence de l'algorithme. D'un point de vue purement naïf, on peut tout à fait déposer la même quantité de phéromones sur chaque chemin. Les fourmis engagées dans des chemins longs iront déposer moins de phéromones puisqu'elles pourront essayer moins de chemins. Et au contraire, les fourmis engagées sur les plus courts chemins pourront très vite essayer d'autres chemins. Tout naturellement, les chemins les plus courts vont se retrouver plus phéromonés que les autres [Schwartz, 08].

Cependant on peut aussi utiliser d'autres modes de dépôt de phéromones. Une idée intéressante est de déposer plus de phéromones à mesure que la solution testée est bonne. Une fonction décroissante basée sur la taille du chemin trouvé peut faire l'affaire. Et bien sûr, rien ne vous oblige à prendre une fonction linéaire, histoire de privilégier encore davantage les meilleurs chemins.

4.4. Le moteur de l'algorithme

Le moteur est un simple conteneur de fourmis qui déroule les itérations en observant d'éventuelles exceptions venant des fourmis. Il analyse les exceptions levées et lorsqu'il s'agit d'une fourmi coincée dans une impasse, il détruit purement et simplement cette fourmi, et quand il s'agit d'une fourmi qui a trouvé un chemin, il note la solution dans un coin et crée une nouvelle fourmi pour remplacer celle qui vient de trouver un chemin [Schwartz, 08].

Le moteur de l'algorithme dépend de 2 paramètres :

- le nombre de fourmis à faire évoluer
- le nombre d'itérations à jouer

4.5. Les données du problème

Le problème contient simplement le graphe à explorer, ainsi que les phéromones de chaque arc. On peut aussi lui donner des paramètres comme les bornes min et max des phéromones à utiliser ainsi que le taux d'évaporation. C'est lui qui va contenir la connaissance collective de la colonie, sous la forme d'une matrice de phéromones.

Résultats : Le programmeur a la main sur plusieurs paramètres pour affiner la recherche de solutions [Schwartz, 08] :

- Le nombre de fourmis.

- Le taux d'évaporation des phéromones.

5. Conclusion

Les méthodes de résolution sont extrêmement nombreuses, elles sont basées sur des principes totalement différents, chacune explore et exploite l'espace de recherche selon des techniques qui lui sont propres. Comparer ces méthodes entre elles n'est pas une chose facile. Toutefois, il n'existe pas de méthodes de recherche qui soit véritablement plus performante qu'une autre sur l'ensemble des problèmes. Pour notre étude, nous avons retenu les colonies de fourmis parce qu'elles sont extrêmement performantes dans de nombreux domaines. C'est une méthode très efficace lorsqu'il s'agit d'exploiter une zone de l'espace de recherche.

Chapitre3 : les méthodes de classification

1. Introduction

Classifier, c'est regrouper entre eux des objets similaires selon tel ou tel critère. Tous Les diverses techniques de classification visent à répartir n individus, caractérisés par p variables X_1, X_2, \dots, X_p en un certain nombre m de sous-groupes aussi homogènes que possible.

On distingue deux grandes familles de techniques de classification :

- La classification non hiérarchique ou partitionnement, aboutissant à la décomposition de l'ensemble de tous les individus en m ensembles disjoints ou classes d'équivalence ; le nombre m de classes est fixé.
- La classification hiérarchique : pour un niveau de précision donné, deux individus peuvent être confondus dans un même groupe, alors qu'à un niveau de précision plus élevé, ils seront distingués et appartiendront à ces deux techniques de classification. Dans la suite du chapitre, nous allons aborder brièvement ces différentes méthodes de classification, [Lebart , 00].

2. Les méthodes de la classification hiérarchiques

Le terme de classification sert à désigner soit une partition soit une hiérarchie. On obtient une partition si l'on partage un ensemble I en un système de classes non vides de telle sorte que tout *individu* i appartienne à une classe et une seule. Si l'ensemble I est divisé en un nombre fini de classes, dont chacune est divisée en un nombre fini de classes,...etc, on parle alors d'une hiérarchie de classes emboîtées. L'exemple de classe plus connu et sans doute le plus cité est celui fourni par les sciences naturelles : "les êtres vivants sont partagés en deux règnes : le règne animal et le règne végétal ; chacun de ces deux règnes est lui-même subdivisé. Par exemple, parmi les animaux, on distingue : vertébrés, invertébrés ; puis parmi les vertébrés : mammifères, oiseaux, reptiles batraciens et poissons".

2.1 Les méthodes hiérarchiques ascendantes

Les méthodes hiérarchiques ascendantes consistent à rassembler, à chaque étape, les éléments (objets ou clusters) les plus similaires au sein d'un même nouveau cluster.

L'objectif de l'algorithme de cette méthode est d'obtenir une hiérarchie, c'est-à-dire une collection de groupes d'observations. [Lebart , 00]

La figure ci-dessous présente c'est quoi une hiérarchie ascendant.

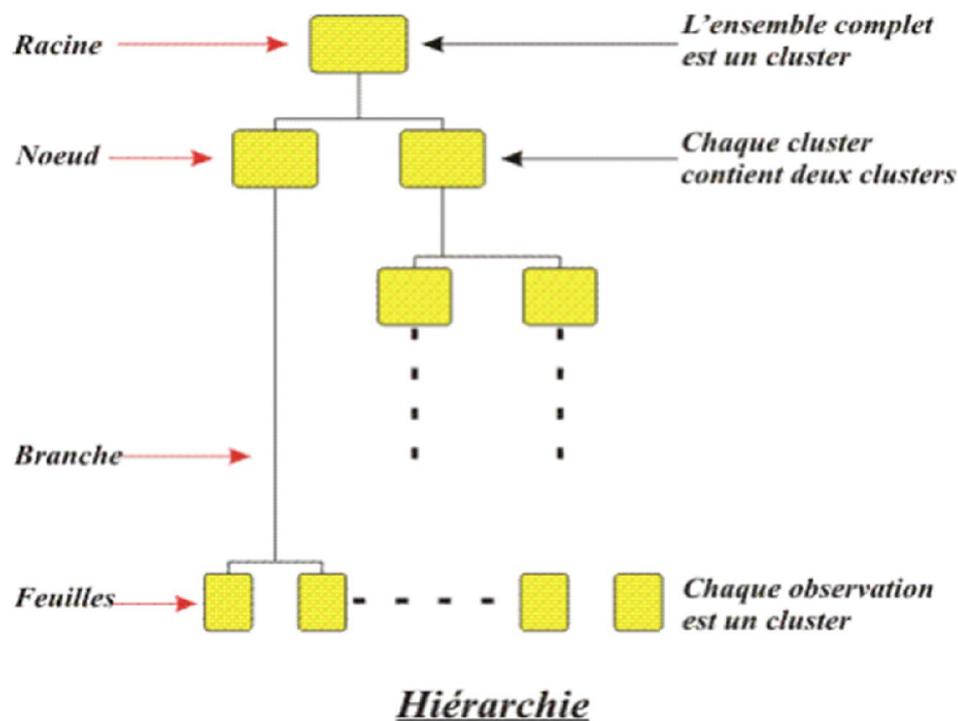


Figure 3.1 : hiérarchie ascendant. [Lebart , 00]

L'algorithme est composé de trois étapes phases qui sont :

- **Étape 1** : il y a n éléments à classer (qui sont les n individus) ;
- **Étape 2** : on construit la matrice de distances entre les n éléments et l'on cherche les deux plus proches, que l'on agrège en un nouvel élément. On obtient une première partition à $n-1$ classes ;
- **Étape 3** : on construit une nouvelle matrice des distances qui résultent de l'agrégation, en calculant les distances entre le nouvel élément et les éléments restants (les autres

distances sont inchangées). On se trouve dans les mêmes conditions qu'à l'étape 1, mais avec seulement **(n-1)** éléments à classer et en ayant choisi un critère d'agrégation. On cherche de nouveau les deux éléments les plus proches, que l'on agrège. On obtient une deuxième partition avec **n-2** classes et qui englobe la première;

- **Etape m** : on calcule les nouvelles distances, et l'on réitère le processus jusqu'à n'avoir plus qu'un seul élément regroupant tous les objets et qui constitue la dernière partition. [Lebart , 00]

2.2 Les méthodes hiérarchiques descendantes

Les méthodes hiérarchiques descendantes déterminent, à chaque étape, le groupe courant le moins homogène et le divisent en deux sous-groupes. Leur schéma général est le suivant :

1. Rassembler tous les objets dans un même cluster. Définir une valeur seuil de distance (ou de dissimilarité) au-dessus de laquelle deux objets ne pourront pas être considérés comme appartenant à un même groupe.
2. Comparer tous les objets deux à deux dans chaque cluster et marquer la paire d'objets ayant la plus grande distance (ou dissimilarité).
3. Si cette distance (ou dissimilarité) est supérieure à la valeur seuil, diviser le cluster correspondant en deux et retourner au point 2. Sinon, fin de la procédure.

Contrairement aux méthodes agglomératives, seule la définition d'une distance (ou dissimilarité) entre deux objets est nécessaire pour les méthodes divisives. Les diverses méthodes se distinguent alors uniquement sur la manière de diviser un cluster en deux sous clusters. [seridi , 13].

3. Les méthodes de classification non hiérarchique (partition) :

Dans le cas du clustering par partition, plusieurs méthodes se distinguent fortement : le clustering basé sur l'optimisation d'inertie est une famille de méthodes de partition la plus antique, par exemple : l'algorithme de regroupement autour de centres mobiles [Forgy, 65]. Le clustering statistique est basé sur l'hypothèse que les données ont été générées en suivant une certaine loi de distribution, le but étant alors de trouver les paramètres de cette distribution, ainsi que les paramètres cachés déterminant l'appartenance des objets aux différentes composantes de

cette loi. Le clustering stochastique consiste à parcourir l'espace des partitions possibles selon certaines heuristiques, et à sélectionner celle qui optimise un critère donné ; le clustering basé sur la densité a pour but d'identifier dans l'espace les zones de forte densité entourées par des zones de faible densité pour la formation des clusters ; comme son nom l'indique. Le clustering basé sur les grilles utilise une grille pour partitionner l'espace de description des objets en différentes cellules, puis identifie les ensembles de cellules denses connectées pour former les clusters ; Le clustering basé sur les graphes consiste à former le graphe connectant les objets entre eux et dont la somme des valeurs des arcs, correspondant aux distances entre les objets, est minimale, puis à supprimer les arcs de valeurs maximales pour former les clusters ; enfin, la base du clustering spectral consiste à projeter itérativement les objets dans des sous-espaces de variance maximum, puis à utiliser une méthode de partitionnement dans de tels sous-espaces pour séparer les données. Dans les paragraphes suivants, nous présenterons brièvement les principales méthodes de clustering par partition, ainsi qu'une liste des caractéristiques principales qui peuvent leur être associées. [seridi , 13].

Il y a plusieurs méthodes de la classification de partition :

- Le clustering basées sur l'optimisation d'inertie.
 - Méthode des centres mobiles
 - Méthode des nuées dynamiques
 - **Méthode des k-moyennes (K-means)**
- Le clustering statistique.
- Le clustering stochastique.
- Clustering basé sur la densité.
- Clustering basé sur les grilles.
- Clustering par la théorie des graphes.
- Clustering spectral

3.1 Méthode des k-moyennes (K-means)

L'algorithme de regroupement autour de centres mobiles est généralement imputé à Forgy [Forgy, 65]. En réalité, de nombreux travaux ont été menés parallèlement sur le thème des centres mobiles, introduisant des variantes [Ball, 67], ou des généralisations. Cette méthode est connue, en anglais, sous le nom de k-means. La méthode des k-moyennes est imputée à MacQueen [J.MacQueen, 65] dont l'algorithme commence

également par un tirage des centres, mais contrairement à la technique des centres mobiles, chaque réaffectation d'un point entraîne une modification immédiate du centre correspondant.

3.2 Algorithme des K-means

La méthode des K-means est très couramment utilisée dans bon nombre d'applications. Il est donc utile d'en discuter un peu plus longuement que d'un simple point de vue descriptif. L'algorithme peut se présenter comme :

1. Choisir au hasard le centroïde de chacun des k clusters.
2. Attribuer chaque objet au cluster dont le centroïde lui est le plus proche.
3. Recalculer les positions des nouveaux centroïdes et mettre à jour les assignations des objets aux clusters en fonction de leur proximité aux nouveaux centroïdes.
4. Répéter les étapes 2 et 3 jusqu'à convergence, c'est-à-dire jusqu'à ce que les centroïdes ne bougent plus (ou ne subissent pas de déplacement significatif).

La figure ci-dessous présente un exemple de clustering par K-means.

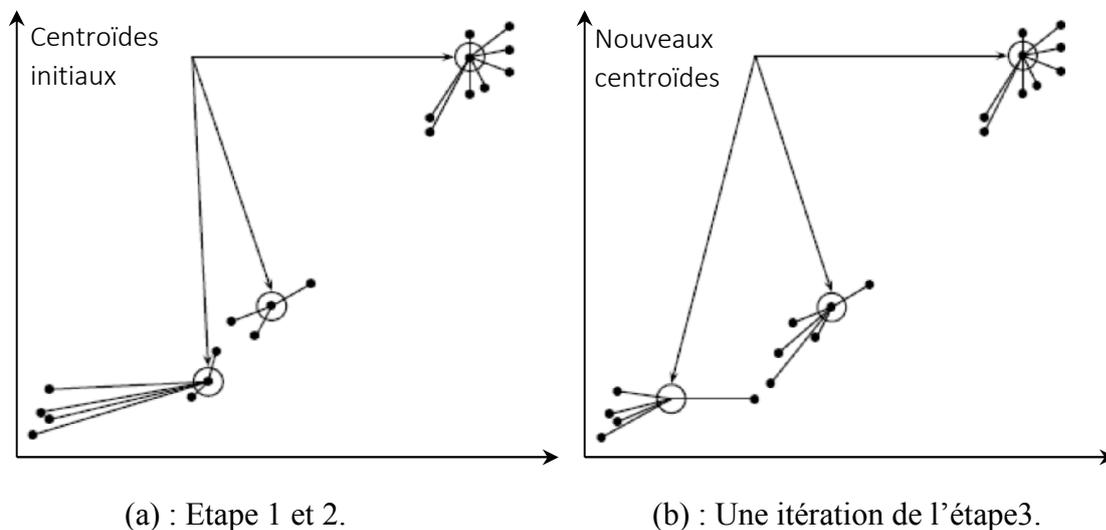


Figure.3.2: clustering k-means.

3.3 Remédiation aux problèmes des K-means

La méthode des K-means étant d'usage fréquent (surtout suite à l'avantage de sa complexité linéaire), différentes variantes ont été développées pour remédier aux difficultés

présentées au paragraphe précédent. Pour permettre de choisir un nombre de clusters le plus cohérent possible avec le jeu de données, on utilisera [Y. De-smet, 05]:

- **une analyse des données** : visualisation des données par analyse discriminante [Lebart, 00].
- **une phase de « split and merge »** : qui consiste à permettre des séparations et fusions dans les clusters à la sortie de la méthode. Typiquement, un cluster sera divisé si sa variance dépasse une valeur seuil. D'autre part, deux clusters seront fusionnés lorsque la distance séparant leur centroïde est inférieure à un certain seuil. Une méthode s'appuyant sur cette démarche est la méthode ISODATA (Iterative Self Organizing Data Analysis Technique) [A.K.Jain, 99].

Pour faire face à la dépendance de la solution finale avec le choix des centres initiaux des clusters (optimum local), deux approches sont reprises ci-dessous [Y. De-smet, 05] :

- **approche simple aléatoire** : k étant supposé choisi convenablement, on exécute à plusieurs reprises la méthode, avec le même nombre de clusters, mais avec des positions initiales des centres différents. On garde la solution donnant la plus petite valeur à la fonction critère.
- **approche par groupements forts** : pour un nombre k fixé, on exécute plusieurs fois la méthode. Des groupes d'objets restant constamment associés à travers les différentes partitions générées sont appelés groupes forts (ou stables). L'idée est alors d'utiliser les centroïdes de ces groupes parmi les positions initiales des centroïdes à fixer afin de relancer la méthode une dernière fois. Notons que cette démarche peut aussi permettre de corriger une valeur de k qui s'avérerait inappropriée.

4. Conclusion

Dans le cadre de classification, il y'a plusieurs méthodes de classification hiérarchiques et non hiérarchiques. Parmi les méthodes de classification les plus utilisées, nous rencontrons la méthode k-means. Dans le chapitre qui suit, nous présenterons les démarches de conception de notre solution préconisée.

Chapitre 4 : La conception

1. Introduction

Un EIAH est un environnement informatique conçu dans le but de favoriser l'apprentissage humain, utilisé dans des situations d'interaction présentes ou à distance. Il y a différents types de EIAH parmi les environnements d'aide à la réalisation d'une activité pédagogique et aussi un Outil de présentation de l'information (hypermédia) et c'est la raison pour laquelle nous avons emprunté cette approche.

Notre travail se concentre sur la conception et la réalisation d'un système hypermédia adaptatif et dynamique, basé sur l'emploi hybride de l'algorithme *des colonies des fourmis* et d'un algorithme *de classification*.

2. Les objectifs de notre système

Notre système est conçu pour atteindre certains objectifs, dont on peut citer :

- Un cours est une liste ordonnée d'Objets Pédagogiques, constituant un chemin vérifiant les objectifs pédagogiques. Le système fournit ainsi à un apprenant particulier les connaissances nécessaires pour atteindre les objectifs visés.
- La génération automatique des contenus pédagogiques adaptés et composés dynamiquement suivant l'algorithme de *colonie des fourmis* et en s'appuyant sur les spécificités caractérisant le profil de l'apprenant (le niveau, le style d'apprentissage) et les caractéristiques des objets pédagogiques.
- La classification des chemins d'OPs générés par l'algorithme des colonies des fourmis et vérifiant un certain nombre de conditions.

3. Architecture générale du système

L'architecture de notre système est composée d'un ensemble d'outils conviviaux, faciles à utiliser pour les usagers qui manipulent le système à savoir :

- les enseignants auteurs responsables des modules.
- les apprenants ont suivi une session d'apprentissage.
- l'administrateur du système qui gère et veille sur toutes les tâches et activités opérés dans le système.

La figure 4.1 illustre explicitement l'architecture logicielle de notre système :

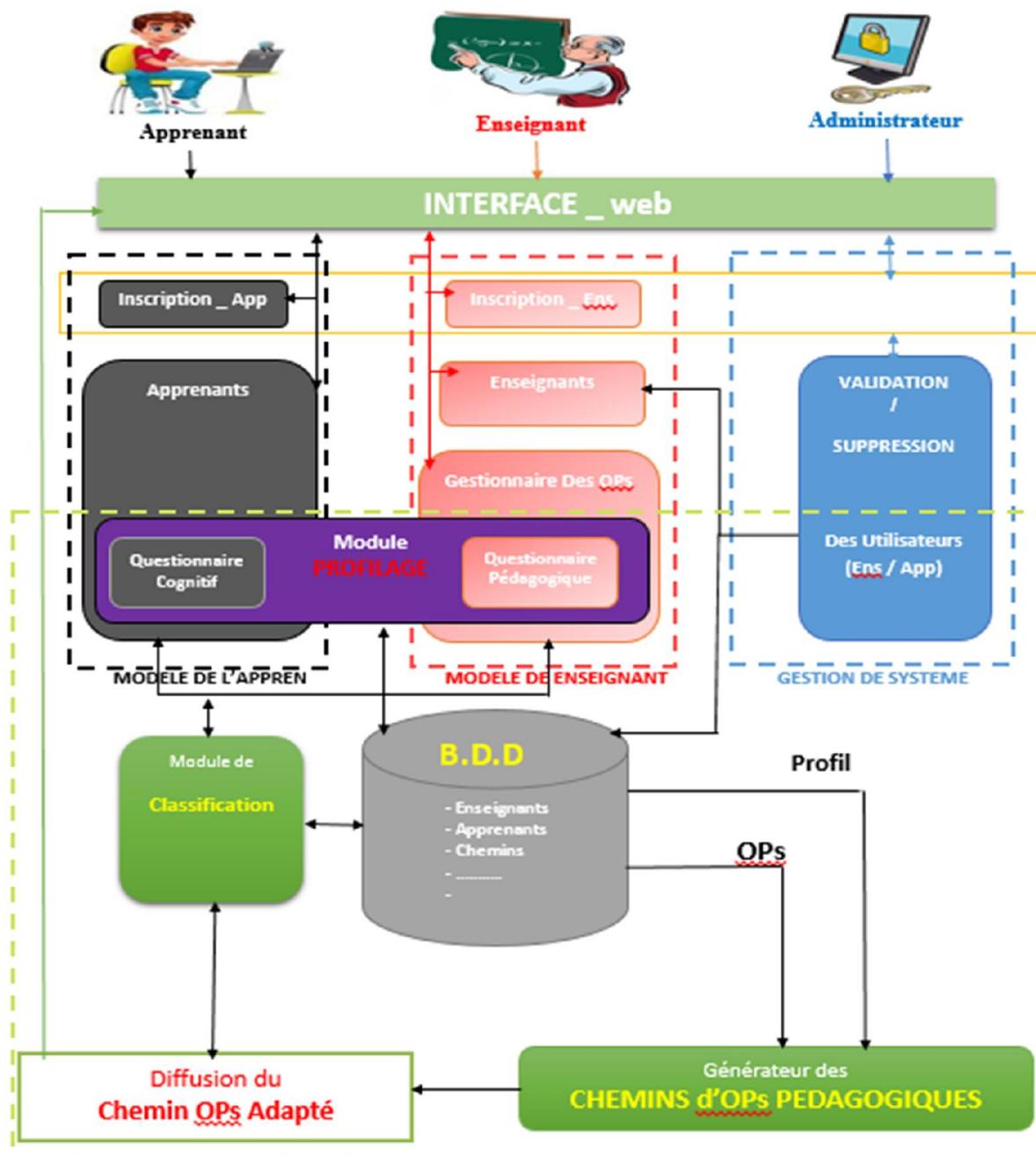


Figure 4.1 : Architecture générale du système

Notre système est composé de :

- **Interfaces** : permettant au système de communiquer avec les différents acteurs à savoir, l'auteur de cours (enseignant) en mode enseignant, l'apprenant en mode apprenant et avec l'administrateur en mode administration.
- **Un modèle apprenant** : Qui contient des informations sur l'apprenant (le niveau cognitif et le style d'apprentissage) stocké dans la base des données.
- **Un modèle enseignant** : Permettant à l'enseignant, l'ajout des cours sous forme d'objets pédagogiques dans une base d'objet pédagogique, alors qu'il peut aussi créer des questionnaires sous forme de QCM.
- **Un Générateur de contenu** : s'appuyant sur un moteur de composition dynamique d'objets pédagogiques permettant la sélection, l'optimisation et la diffusion de contenu pédagogique en rapport avec son profil.

Et ce dernier est composé de trois modules, afin de faciliter sa tâche, qui sont :

- **Module de profilage** : qui est chargé de concevoir un profil pour chacun des apprenants selon son niveau et ses préférences qui sont recueillis par un formulaire d'inscription et les deux questionnaires (psycho & pédagogique).
- **Générateur de chemins pédagogiques** : c'est le cœur de notre système, sa tâche est la plus importante, c'est la génération des chemins d'OPs, de façon dynamique pour chaque apprenant selon *l'algorithme de colonie de fourmis*. (voir figure 4.5)
- **Module de classification** : qui est chargé de comparer le profil du nouveau apprenant avec ceux qui sont déjà inscrits dans la base et ayant suivi une session d'apprentissage, afin de lui affecter le chemin pédagogique approprié sans passer par **Le module générateur de chemins pédagogiques**. (voir figure 4.5)

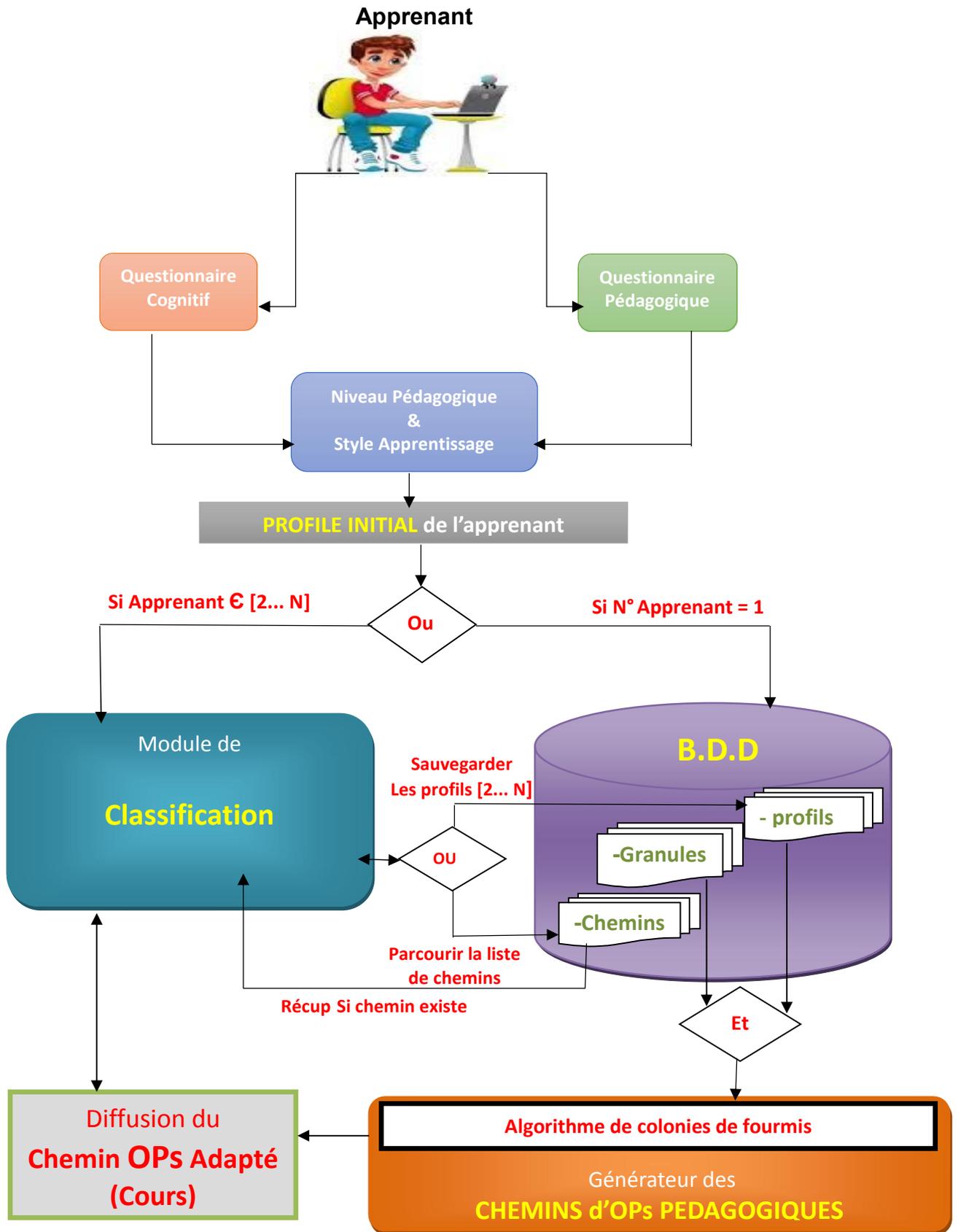


Figure 4.2 : Schéma détaillé de la structure modulaire de notre système.

3.1. Interface

Notre système est constitué d'une interface composée de :

3.1.1. Une interface *administrateur* qui permet de :

- valider l'inscription des apprenants et des enseignants.
- consulter, ajouter ou supprimer les différents acteurs (apprenants, enseignants) du système.
- organiser les parcours d'apprentissage (ajout ou suppression de module de formation).

3.1.2. Une interface *mode enseignant* permettant à l'enseignant de :

- S'inscrire dans le système
- Se connecter à son espace par son Pseudonyme et son mot de passe.
- Déposer, générer et supprimer les objets Pédagogiques.
- Saisir un nombre illimité de questions de types QCM qui vont apparaître dans le questionnaire Pédagogique.
- Créer des questionnaires de type QCM pour évaluer chaque chapitre.

3.1.3. Une Interface en *mode apprenant* lui permettant de :

- S'inscrire dans le système.
- Se connecter à son espace par un Pseudonyme et un mot de passe.
- Suivre un apprentissage progressif et adaptatif ou non, en fonction de son niveau et ses besoins.

3.2. Modèle Apprenant

Pour qu'un système d'apprentissage soit intelligent, il faut qu'il soit capable de s'adapter à l'apprenant qui se trouve devant la machine, ceci ne peut être atteint que par la connaissance du modèle de l'apprenant, l'ensemble des informations recueillies (les résultats des deux questionnaires) aidera à définir les caractéristiques des apprenants d'une façon à ce qu'il soit plus facile de leurs adapter les contenus.

Nous avons donc décidé d'organiser les informations du modèle apprenant en quatre catégories en plus de ses informations identitaires (nom, âge, adresse email, prénom, pseudonyme et mot de passe).

3.2.1. Niveau Cognitif

Lors de la phase d'apprentissage, les apprenants subissent un test pédagogique qui va les répartir en cinq catégories (*Mauvais, Passable, Bien, Très bien, Excellent*). Ce test est un ensemble de cinq questions sous forme de QCM proposées par l'enseignant. L'évolution du niveau de connaissance de l'apprenant durant une session d'apprentissage est induite par une méthode d'évaluation informative à la fin de chaque chapitre consulté.

3.2.2. Style d'apprentissage

Va permettre à l'apprenant de spécifier les types de média préférés, modalité de perception et le traitement de l'information. L'apprenant doit passer un test psychologique qui va déterminer son style d'apprentissage ce dernier va l'accompagner jusqu'à la fin.

3.3. Modèle Enseignant

Nous avons fragmenté le contenu du cours en objets pédagogiques. Pour cela, il était nécessaire de spécifier les paramètres décrivant ces fragments (nom de chapitre, niveau de l'apprenant, nouveau chapitre ou chapitre déjà existant, titre d'OPs, bloc et source d'OPs) et fournir ainsi des fonctionnalités facilitant la recherche, le filtrage et la construction (assemblage) du contenu.

3.3.1. Le cours :

Le cours est un regroupement d'objets pédagogiques tel qu'un objet est identifié par bloc (introduction, fondement, conclusion, td, tp) (*voir figure 4.3*), niveau (*Mauvais, Passable, Bien, Très bien, Excellent*), style (image, son, texte, vidéo), chapitre et par module.

Nous proposons une granularité assez fine du cours afin qu'il puisse constituer une source efficace aux diverses activités proposées aux apprenants, nous affirmons que le contenu est fragmenté en petites unités, ces unités seront plus partageables entre les acteurs du système donc plus réutilisables ce qui va augmenter et favoriser énormément l'adaptation aux apprenants.

Chaque chapitre doit respecter la structure suivante :

- **Le premier bloc** : appeler introduction contient tous les objets présentent une introduction pour chaque chapitre.
- **Un deuxième bloc** : appelé fondamentale, il contient le contenu du chapitre.

- **Le troisième bloc** : contient les conclusions.
- **Le quatrième bloc** : contient des travaux pratiques
- **Le cinquième bloc** : contient des travaux dirigés.

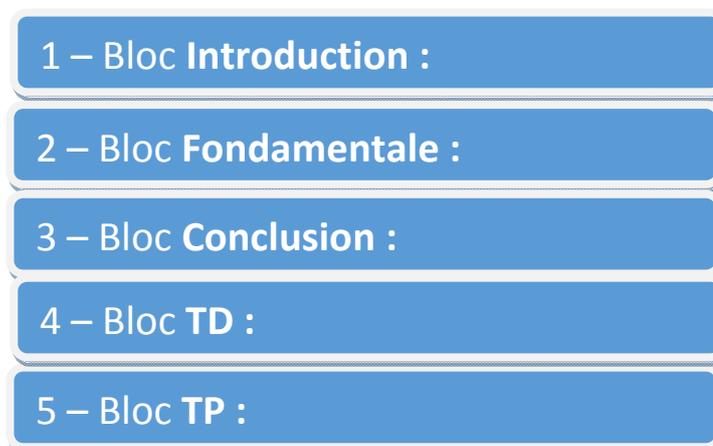


Figure 4.3 : Structure d'un chapitre

Remarque :

Après la consultation de chaque chapitre l'apprenant doit passer un test d'évaluation, concernant ce dernier, le test doit prendre en compte le niveau de l'apprenant. La figure suivante présente la structure d'un cours d'une façon assez simple.

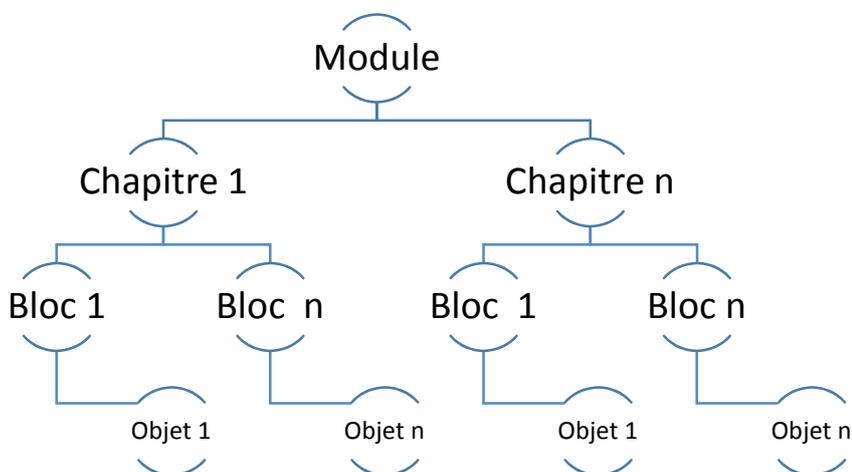


Figure 4.4 : Représentation hiérarchique d'un cours.

Le déroulement d'une session d'enseignement suit les étapes suivantes :

Etape 1 : L'enseignant commence une session d'apprentissage par l'accès au système via n'importe quel navigateur standard tel que : Internet explorer ou Mozilla FireFox ;

Etape 2 : Le système ouvre alors un dialogue pour demander à l'enseignant s'il est déjà inscrit ou non.

1- Si l'enseignant n'est pas encore inscrit (nouveau enseignant), le système affiche un formulaire d'inscription à l'enseignant, afin de saisir toutes les données nécessaires à son inscription. Cette demande d'inscription sera envoyée à l'administrateur pour la valider.

2- Si l'enseignant est déjà inscrit dans le système (ancien enseignant), le système se charge de lui demander de sélectionner le module qui n'est pas encore affecté à aucun enseignant dans le système, après le choix du module, l'enseignant commence l'ajout de tous les objets pédagogiques nécessaire qui composent les chapitres de chaque module. Ensuite, il doit poser les questions de type **QCM** au nombre illimité, ou bien accéder à la base des objets pédagogiques pour la réutilisation ou la modification des objets pédagogique attaché à son module.

3.4. Générateur de contenu

Afin de mieux élaborer notre système, nous aurons besoin de rechercher, de stocker, et de mettre à jour d'une part, les objets pédagogiques, et d'autre part les profils apprenants.

Le générateur de contenu a comme objectifs :

- La sélection du profil de l'apprenant et les attributs des objets pédagogiques qui sont l'entrée du *générateur de chemin pédagogique*,
- La composition dynamique des objets qui vont être présentées à l'apprenant sous forme d'un cours adaptatif.

3.4.1. Module de profilage

Le module profilage est un module qui permet de gérer des informations provenant des questionnaires cognitifs et pédagogiques afin de construire un profile apprenant. Cela permet à l'apprenant de suivre un parcours adapté à ses compétences (**son niveau**) et ses préférences (**son style d'apprentissage**).

Le fonctionnement du **module profilage** sur une session d'apprentissage commence à l'arrivée d'un apprenant jusqu'à la sauvegarde des données dans la base du modèle apprenant. Le déroulement d'une session d'apprentissage d'un apprenant suit les étapes suivantes :

Etape 1 : L'apprenant commence une session d'apprentissage par l'accès à notre système via n'importe quel navigateur standard tel que : Google chrome ou Mozilla Firefox, ou autres...;

Etape 2 : Le système se charge et ouvre une fenêtre pour demander à l'apprenant s'il est déjà inscrit ou non. Dans ce cas, on trouve deux catégories d'utilisateurs :

1- Nouveau apprenant : Si l'apprenant n'est pas encore inscrit (nouveau apprenant), le système lui demande de saisir toutes les informations nécessaires pour l'inscription du nouveau apprenant suivant un formulaire prédéfini. Cet apprenant ne devient membre qu'après validation de son inscription par l'administrateur.

2- Ancien apprenant : c'est un utilisateur déjà inscrit dans le système, dans ce cas, l'utilisateur accède à son espace par son pseudo et son mot de passe.

Etape 3 : Dans l'espace apprenant et après la sélection du module à suivre durant la session d'apprentissage, le système lance le module profilage qui a pour rôle de déterminer le profil initial de l'apprenant (niveau cognitive dans la matière sélectionnée et styles d'apprentissage). Le dialogue initié par le module profilage se traduit par deux types de questionnaires soumis à l'apprenant et qui définissent son point de départ dans la session d'apprentissage (l'un pédagogique sous forme de QCM en rapport avec la matière choisie, l'autre représente le questionnaire VARK (3.4.1.1.A)).

Ces deux questionnaires seront détaillés dans la section (3.4.1.1), les renseignements tirés par le module profilage formeront le profil de l'apprenant et seront enregistré dans le modèle de l'apprenant.

3.4.1.1. Les tests d'entrées

A. Questionnaire psychologique

C'est la première phase d'évaluation de l'apprenant. On ne peut pas aller directement au test pédagogique avant que nous ayons effectué une étude psychologique détaillée. Notre système utilise en entrée les réponses du questionnaire psychologique (case de multiple choix). Ce questionnaire est développé en étroite collaboration avec des spécialistes en psychologie, afin de mettre l'apprenant dans les meilleures conditions. L'approche que nous

avons utilisé repose sur la notion de préférence d'apprentissage, un modèle de style d'apprentissage **VAR**K (**v**isual /**a**ural, **r**ead/**w**rite/**k**inesthetic).

Ce questionnaire vous renseignera sur le style d'apprentissage que vous favorisez pour acquérir et utiliser les idées et les informations. Choisir la réponse qui décrit le mieux votre préférence et encrer la lettre correspondante. Vous pouvez encrer plus d'une lettre si une réponse unique ne vous suffit pas. Ne pas répondre à la question si aucune solution ne vous convient. [2]

B. Questionnaire pédagogique :

Ce questionnaire permet de vérifier que l'apprenant a les connaissances ou les compétences nécessaires pour pouvoir suivre la session d'apprentissage avec plus de succès. Ce questionnaire est composé des questions sous forme QCM proposé par l'enseignant responsable du module. En cas où le test d'entrée (questionnaire pédagogique) n'est pas rempli, l'apprenant peut consulter tous les OPs déposés par l'enseignant (tous les niveaux). Dans ce cas, on aura un apprentissage libre non adapté.

3.4.2. Module de classification

Ce module est aussi très important dans notre application, son but est de classer les chemins d'OPs générés par l'ACO et ayant conduit l'apprenant à une réussite lors d'une session d'apprentissage précédente.

L'algorithme de k-means est le plus connu dans le domaine de classification. Pour utiliser k-means nous devons fixer à l'avance le nombre de classe et le nombre d'effectif, chose qui n'est pas possible dans notre cas, vu que le nombre d'apprenants évolue dans le temps et le nombre de classe aussi. Donc deux paramètres dynamiques qui nous obligent à choisir un algorithme de classification plus utile et plus efficace pour notre système, ce dernier est décrit comme suit :

1. **NbApp = 1** : Pour le premier apprenant il y'a deux cas possible :

- Si l'apprenant a passé tous les tests d'évaluations du premier chapitre au dernier. Donc, ce chemin est considéré optimale pour l'apprenant en question, alors il sera classé. (**NbClass = 1**).

- Si l'apprenant a échoué dans au moins un test d'évaluation, ça veut dire, que le chemin généré ne lui convient pas. Dans ce cas, le chemin ne sera pas classé ($NbClass = 0$).

2. $NbApp \geq 2$: dès l'arrivée d'un nouveau apprenant. On a deux scénarios :

L'apprenant a passé les deux tests cognitif et psychologique. Le module de classification va chercher dans la BBD s'il y a un chemin pédagogique sauvegardé qui correspond au profil du nouvel apprenant.

- Si on trouve un, on ignore la phase du module générateur des chemins (**ACO**).
- Si on ne trouve pas un, on passe par la phase du module générateur des chemins (**ACO**). Si par la suite, on trouve que ce chemin n'est pas classé, alors, ce dernier sera le nouveau chemin à classer et donc le nombre de classe augmente de plus 1 ($NbClass = NbClass + 1$).

3.4.2.1. Comment classer un chemin

Pour classer un chemin, nous avons mis la condition suivante :

- l'apprenant doit garder un niveau constant pendant son parcours. Pour cela, il faut qu'il obtienne plus de 60 % dans tous les tests d'évaluation de chaque chapitre.

Remarque

En cas d'échec dans l'un des tests d'évaluation au cours de son parcours d'apprentissage, l'apprenant sera sanctionné en dégradant son niveau pédagogique. Dans ce cas, on dit que le chemin recommandé n'est pas un chemin optimal pour cet apprenant. Donc, le chemin ne sera pas considéré dans la phase de classification.

3.4.3. Générateur de chemin

La *figure 4.5* montre le fonctionnement du processus d'adaptation et de génération en détails

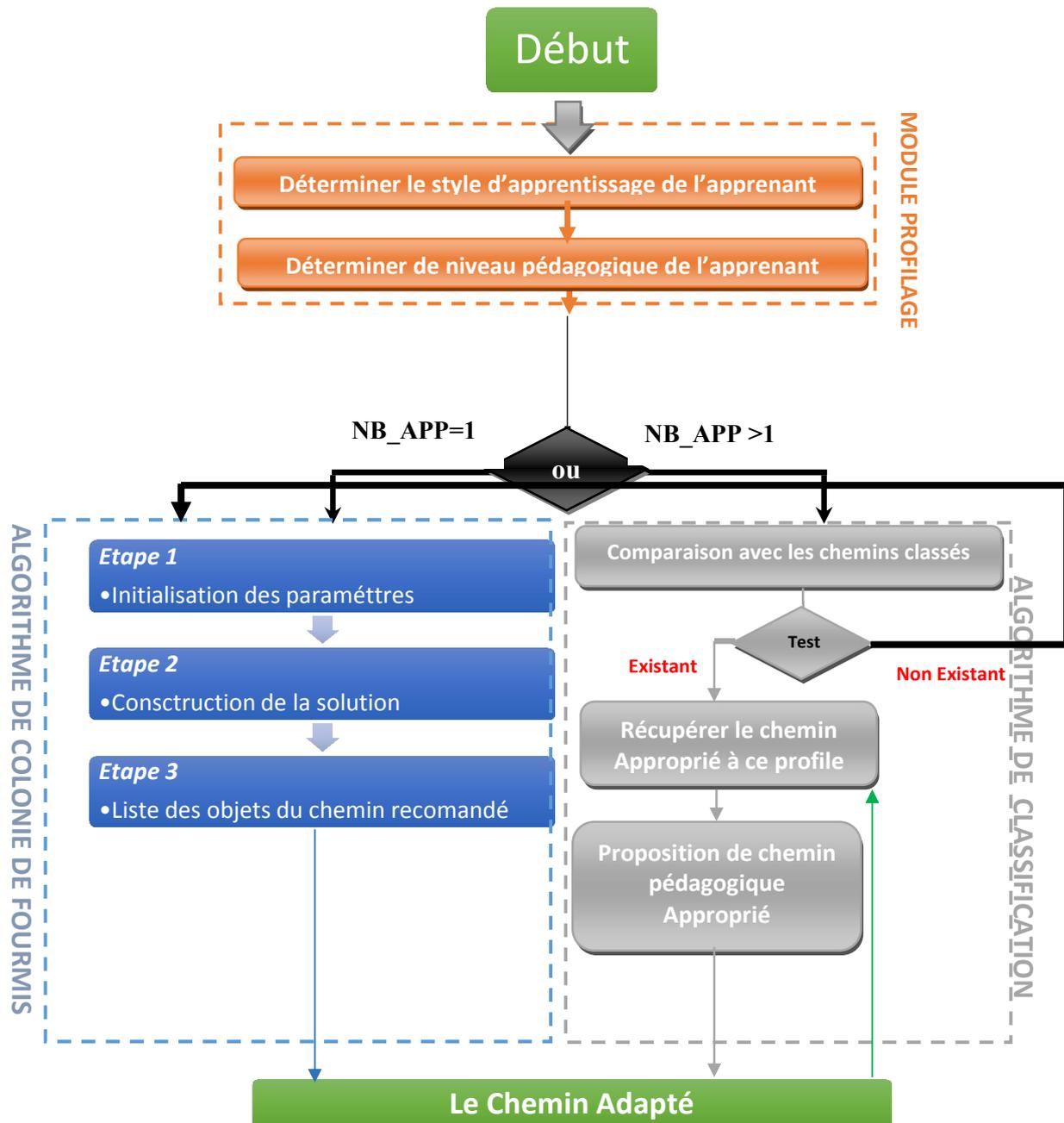


Figure 4.5 : fonctionnement du processus d'adaptation et de génération de chemins d'OPs.

On résume le fonctionnement du processus d'adaptation et de génération schématisé dans la *figure4.5*, dans les étapes suivantes :

Etape 1 : le système commence la génération du cours par l'accès à la base du profil de l'apprenant pour sélectionner et récupérer les informations concernant son niveau et ses préférences. On utilise les règles suivantes pour déterminer son niveau initial :

❖ **Remarque :** La variable «*Note_{AP}*» signifie une variable contenant la note du questionnaire pédagogique.

- Si : $Note_{AP} \leq 6$: l'apprenant à un *Niveau mauvais*
- Si : $6 < Note_{AP} \leq 10$: l'apprenant à un *Niveau passable*.
- Si : $10 < Note_{AP} \leq 13$: l'apprenant à un *Niveau bien*.
- Si : $13 < Note_{AP} \leq 16$: l'apprenant à un *Niveau Très bien*.
- Si : $Note_{AP} \geq 16$: l'apprenant à un *Niveau excellent*.

Etape 2 : dans cette étape le module de classification va chercher dans la BBD s'il y a un chemin pédagogique sauvegardé qui correspond au profil de l'apprenant. Si on le trouve, on passe directement à l'étape 4 sinon on passe à l'étape suivante.

Etape 3 : Après que le *profil* de l'apprenant a été déterminé, le module *Générateur de chemins OPs* utilise ces informations comme données d'entrées pour son algorithme (colonies de fourmis) de calcul. Le résultat du traitement apparaît comme étant une liste d'OPs recommandés pour ce profil. Autrement dit, il génère un *chemin* OPs adapté à l'apprenant en question.

Etape 4 : Affichage du cours à l'apprenant sous forme d'OPs.

Etape 5 : Après une session d'apprentissage d'un chapitre. L'apprenant passe à des tests d'évaluations. Les résultats obtenus décideront du classement des chemins d'OPs doivent être classé ou pas. Si les résultats sont considérés comme favorables alors les chemins d'OPs générés seront enregistrés dans la base des données afin que le système puisse les réutiliser une autre fois, pour un nouveau profil similaire.

Remarque

Le premier apprenant ne passe pas par l'étape 2.

4. Structure de la base des données

4.1. Le modèle conceptuel des données

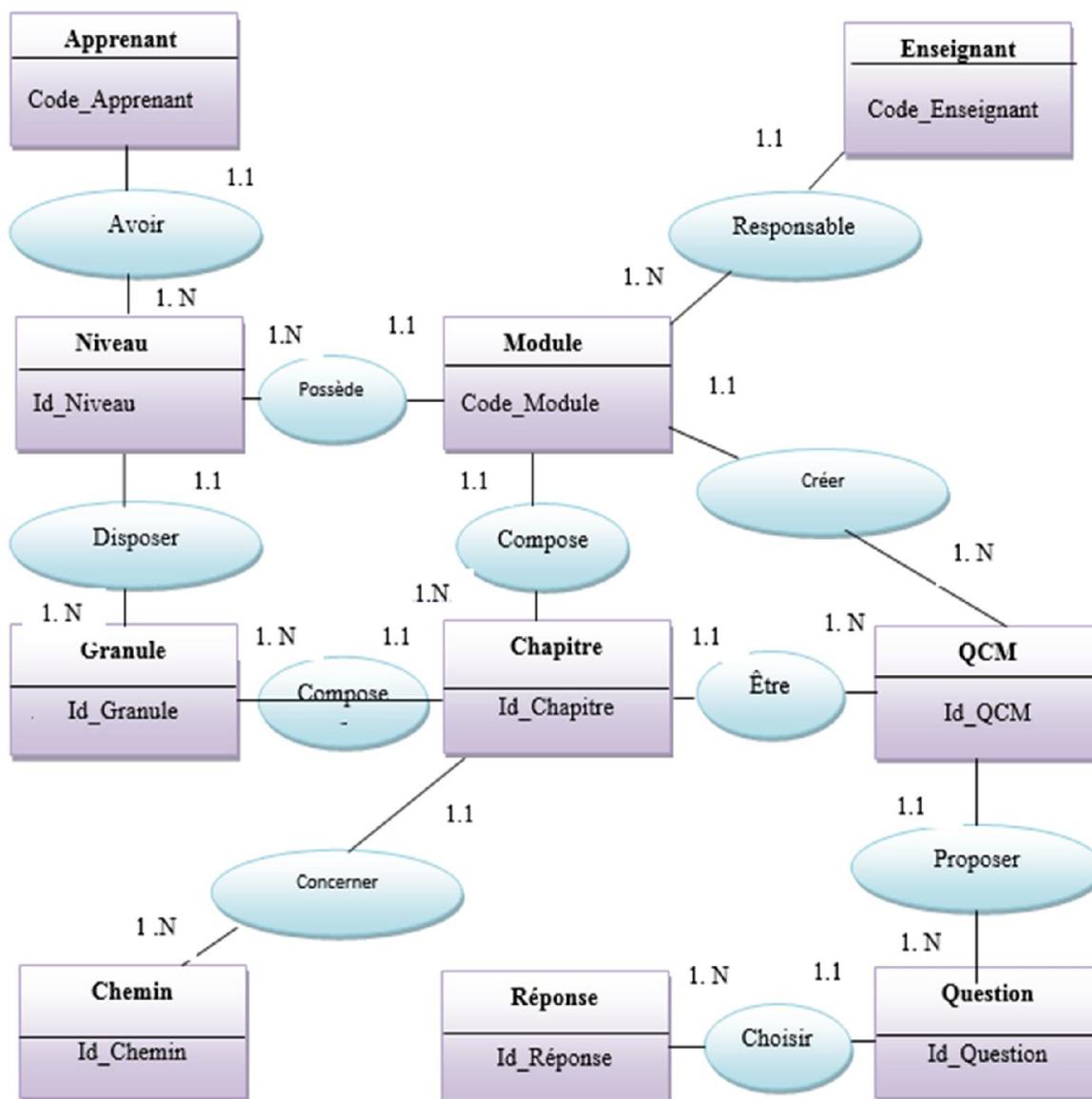


Figure 4.6 : Le modèle conceptuel des données (M.C.D).

4.2. La liste des entités

Table	Les champs	L'identifiant	Type
Apprenant	-Code_app -Nom_app -Prenom_app -Pseudo_app, -Mot_app -Adr_email_app -Age_app -étas_app	Code_app	Entier Caractère Caractère Caractère Caractère Caractère Entier Entier
Enseignant	-Code_ens -Nom_ens -Prenom_ens -Pseudo-ens, -Mot-ens -Adr_email -étas_ens	Code_ens	Entier Caractère Caractère Caractère Caractère Caractère Entier
Module	-Code-mod -Nom_mod -étas_mod	Code-mod	Entier Caractère Entier
Granules	-id-gran -titre_gra -source_gra -bloc_gra -preg_gra -ext_gra	Code-gran	Entier Caractère Caractère Caractère Caractère Caractère
Chapitre	-id-cha -titre_chapitre	id_cha	Entier Caractère
Question	-id-question -txt_question	id_question	Entier Caractère
QCM	-id-QCM -txt_QCM	id_QCM	Entier Caractère
Niveau	-id-niveau -val_niveau -grandu_niveau	id_niveau	Entier Entier Entier

Chemin	<u>-id-chemin</u> -style_chemin -path_chemin -clas_chemin	id-chemin	Entier Entier Entier Caractère
---------------	--	-----------	---

Tableau 4.1 : La liste des entités

4.3. Le modèle logique des données (M.L.D)

Apprenant : (Code_app, Nom_app, Prenom_app, Pseudo-app, Mot-app , Adr_email_app , Age_app , étas_app)

Enseignant : (Code_ens, Nom_ens, Prenom_ens, Pseudo-ens, Mot-ens , Adr_email_ens , Age_ens , étas_ens)

Module : (Code-mod, Nom_mod, étas_mod, #code_ens)

Granule : (id-gran ,_titre_gra ,_source_gra ,_bloc_gra ,_preg_gra ,_ext_gra , #id_chapitre , #id_niveau)

Chapitre : (id-chapitre , titre_chapitre ,#code_module)

Question : (id-question , txt_question , #id_QCM)

QCM : (id-QCM , titre_QCM , #id_chapitre , #code_module)

Niveau : (id-niveau, val_niveau , grandu_niveau , #code-app , #code_mod)

Chemin : (id-chemin , style_chemin ,path_chemin ,clas_chemin , #id_chapitre)

5. Conclusion

Dans ce chapitre, on a présenté d'une manière détaillée la partie conception de notre système. Premièrement, nous avons proposé un système de composition dynamique pour générer des cours pédagogiques. La composition est faite à l'aide de l'algorithme ACO. Notre système est composé d'un ensemble d'interfaces et modèles afin d'assurer son bon fonctionnement. Chaque interface est associée à un acteur particulier. Dans la prochaine section, nous détaillerons notre implémentation du futur système.

Chapitre 5 : Implémentation

1. Introduction

L'implémentation de notre application consiste à la création d'un site web qui va supporter la plateforme e-Learning, tout on se basant sur système hypermédia adaptatif pour la mise en œuvre d'un ensemble de parcours adaptatifs que les apprenants vont suivre.

Ce chapitre a pour objectif de présenter la réalisation d'un environnement informatique. Ce dernier est mis à la disposition des apprenants à travers un système d'intranet. Les apprenants vont poursuivre une formation à distance adaptée à leurs besoins. Les enseignants vont déposer leurs supports pédagogiques sous forme de granules (objets pédagogiques).

2. Environnement du travail

2.1. Langage de programmation

2.1.1. PHP

C'est un langage de script HTML, qui fonctionne côté serveur. Ce qui distingue le PHP des langages de script comme le JavaScript est que le code est exécuté sur le serveur. La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données. Réaliser une page web dynamique interfaçant une base de données est extrêmement simple. Il se pose maintenant en plate-forme orientée objet, dotée de plus de 3000 fonctions différentes et de capacités de manipulation du langage XML de PHP [Olivier Heurtel, 07].

- **Apache http serveur :**

Le logiciel apache http serveur, souvent appelé apache, est un serveur http produit par Apache Software Foundation. C'est le serveur HTTP le plus populaire du World Wide Web. C'est un logiciel libre.

Apache fonctionne principalement sur les systèmes d'exploitation Unix et Windows. La version Windows n'est pas considérée comme stable que depuis la version 2.0 d'apache.

Apache est conçu pour supporter de nombreux modules lui donnant des fonctionnalités supplémentaire : interprétation du langage Perl, PHP et Python. [3]

- **MySQL :**

Est un gestionnaire de base de donnée relationnelles libre. Il est très utilisé dans les projets libres. C'est un SGBD développé dans un souci de performances élevées. Il est multiutilisateur. MYSQL fonctionne sur beaucoup de plateformes différentes, incluant Unix, Windows 95, 98, NT 200 et XP....etc. [4]

2.1.2. JavaScript:

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

Le langage a été créé en 1995 par Brendan Eich pour le compte de Netscape Communications Corporation. Le langage, actuellement à la version 1.8.2 est une implémentation de la 3^e version de la norme ECMA-262 qui intègre également des éléments inspirés du langage Python. La version 1.8.5 du langage est prévue pour intégrer la 5^e version du standard ECMA. [5]

2.1.3. Ajax

Ajax (Asynchrones JavaScript and XML) est une manière de construire des applications Web et des sites web dynamiques basés sur diverses technologies Web ajoutées aux navigateurs dans les années 1990.

Ajax est la combinaison de technologies telles que JavaScript, CSS, XML, le DOM et le XMLHttpRequest. AJAX permet aux pages web d'être mis à jour de manière asynchrone par échange de petites quantités de données avec le serveur dans les coulisses. Cela signifie qu'il est possible de mettre à jour les parties d'une page web, sans avoir à recharger toute la page. [6]

Pages web classique, (qui n'utilisent pas AJAX) il faut recharger la page entière si le contenu doit changer.

Exemples d'applications utilisant AJAX : Google Maps, Gmail, YouTube, Face book.

2.2. Environnement de développement

2.2.1. EasyPHP

EasyPHP fut le premier package WAMP à voir le jour (1999). Il s'agit d'une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL phpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP. Par défaut, le serveur Apache crée un nom de domaine virtuel (en local) 127.0.0.1 ou local host. Ainsi, quand on choisit « Web local » dans le menu d'EasyPHP, le navigateur s'ouvre sur cette URL et affiche la page index. PHP de ce site qui correspond en fait au contenu du dossier www d'EasyPHP. [7]

3. Description des tables de la base de données

Notre base de données est réalisée en MySQL. Cette base est structurée sous la forme de tables :

- **Table d'apprenant** : elle permet de stocker tous les informations qui concernent l'apprenant, nous avons utilisés un champ code_app pour identifier chaque apprenant de façon unique.
- **Table enseignant** : elle permet de stocker les informations relatives aux enseignants qui ont des comptes dans notre système.
- **Table module** : cette table contient les modules et les informations qui les concernent.
- **Table granule** : cette table permet à l'enseignant de stocker les objets pédagogiques des cours (sous forme de granules).
- **Table chapitre** : elle permet de stocker les informations relatives aux chapitres de chaque module.

- **Table question** : elle permet de stocker les questions de qcm.
- **Table réponse** : elle permet de stocker les réponses du qcm.
- **Table QCM** : elle responsables des qcm du system.

4. Structure de l'application :

Le site web qu'on a développé contient une page d'Accueil qui comporte des liens hypertexte vers les trois interfaces du système (administrateur, enseignant et apprenant).

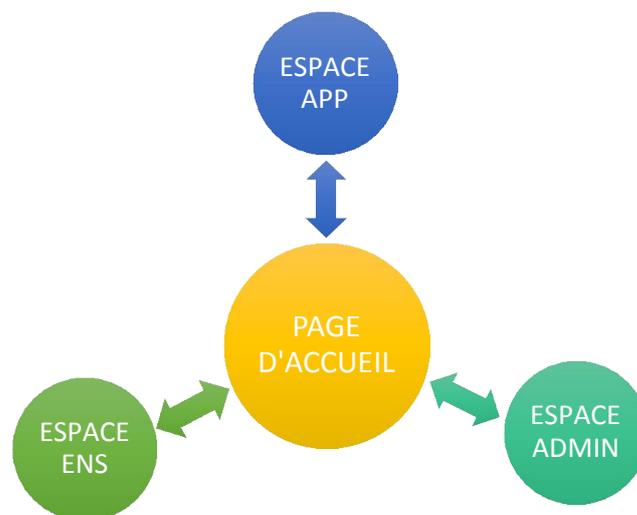


Figure 5.1 : Les liens de la page d'accueil.



Figure 5.2 : Interface principale du système (Page D'accueil).

4.1. Description du fonctionnement et présentation du logiciel :

L'accès à notre système, se fera à travers l'interface principale **PAGE D'ACCUEIL**. Elle contient les options suivantes :

- **Connexion** : c'est le lien qui permet d'accéder à l'interface des acteurs.
- **Inscription** : c'est le lien qui permet d'inscrire les acteurs.
- **Notre système** : Pour avoir une idée générale sur notre système.
- **Aide** : Pour faciliter l'utilisation de notre système.
- **Contactez-nous** : pour contactez l'administrateur.

4.1.1. Interface administrateur

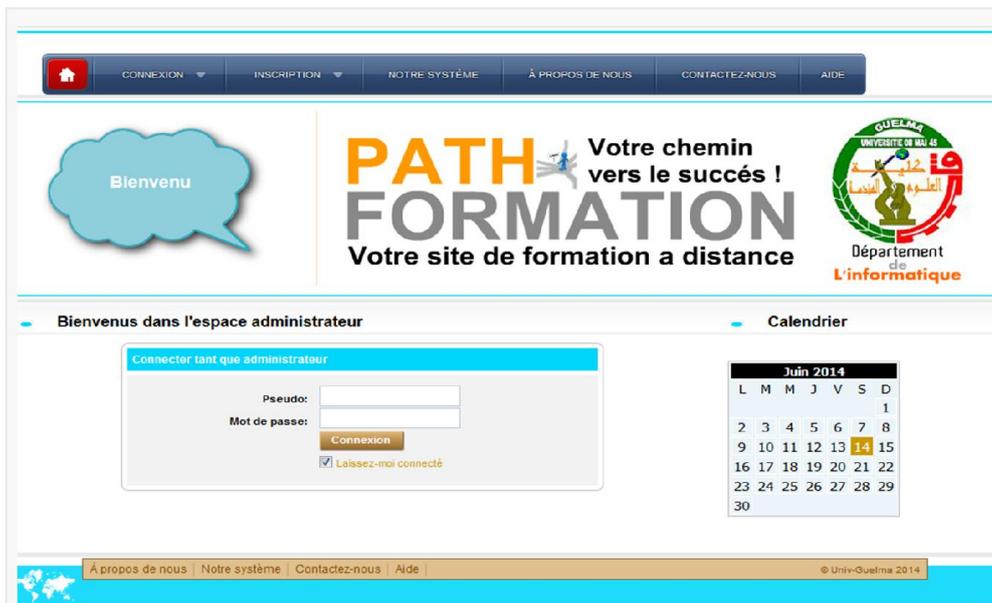


Figure 5.3 : Interface d'accès de l'administrateur.

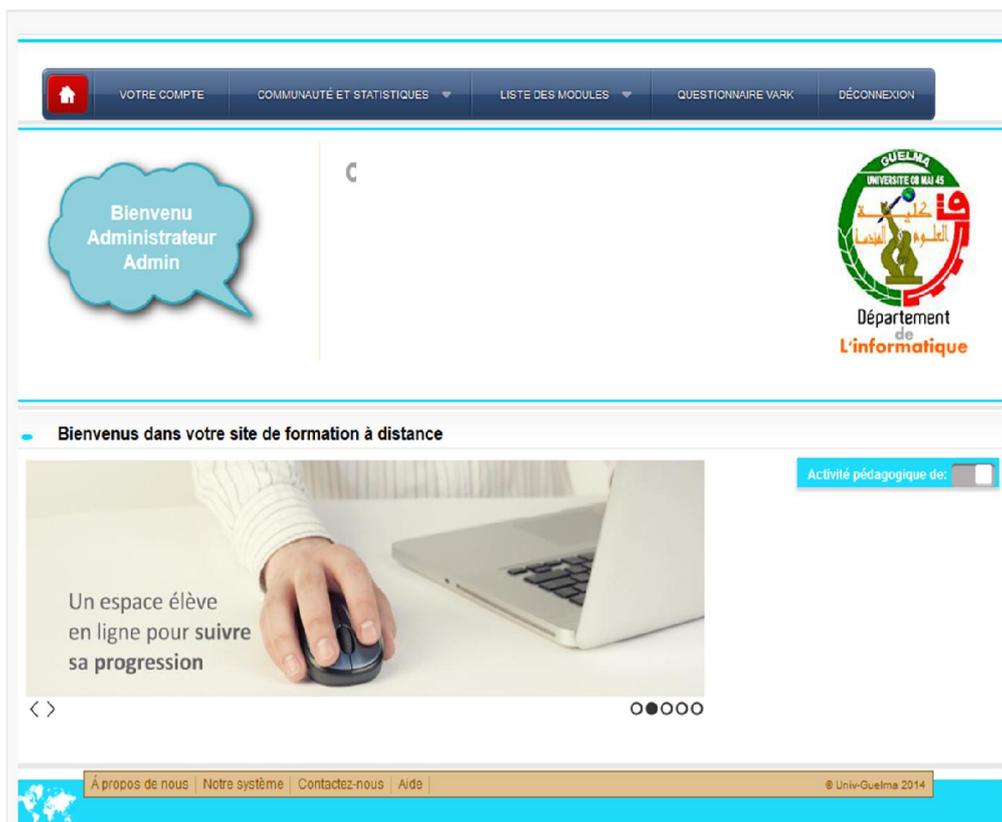


Figure 5.4 : Interface de l'espace administrateur.

On résume le fonctionnement de l'administrateur (l'utilisation de cinq boutons) par la structure hiérarchique suivante :

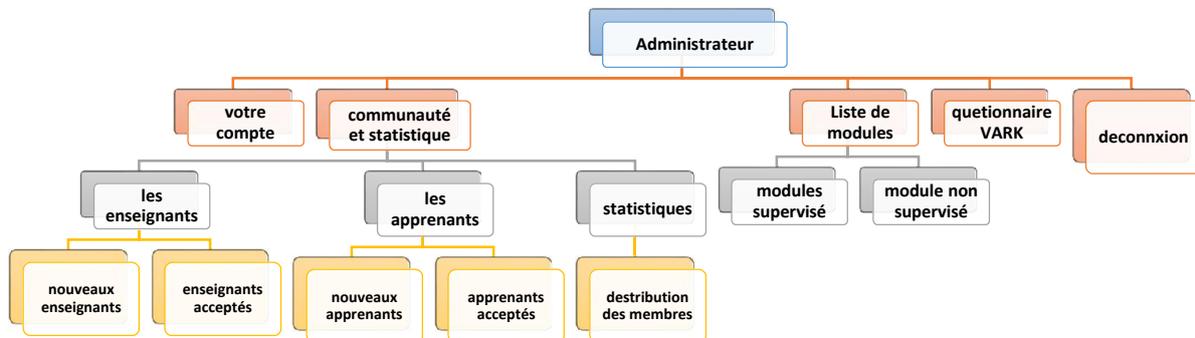


Figure 5.5 : Architecture de l'interface administrateur.

- **Nouveaux apprenants** : c'est un lien qui permet d'afficher la liste des nouveaux apprenants récemment inscrits et attendant leurs validation par l'administrateur (accepter / refuser).

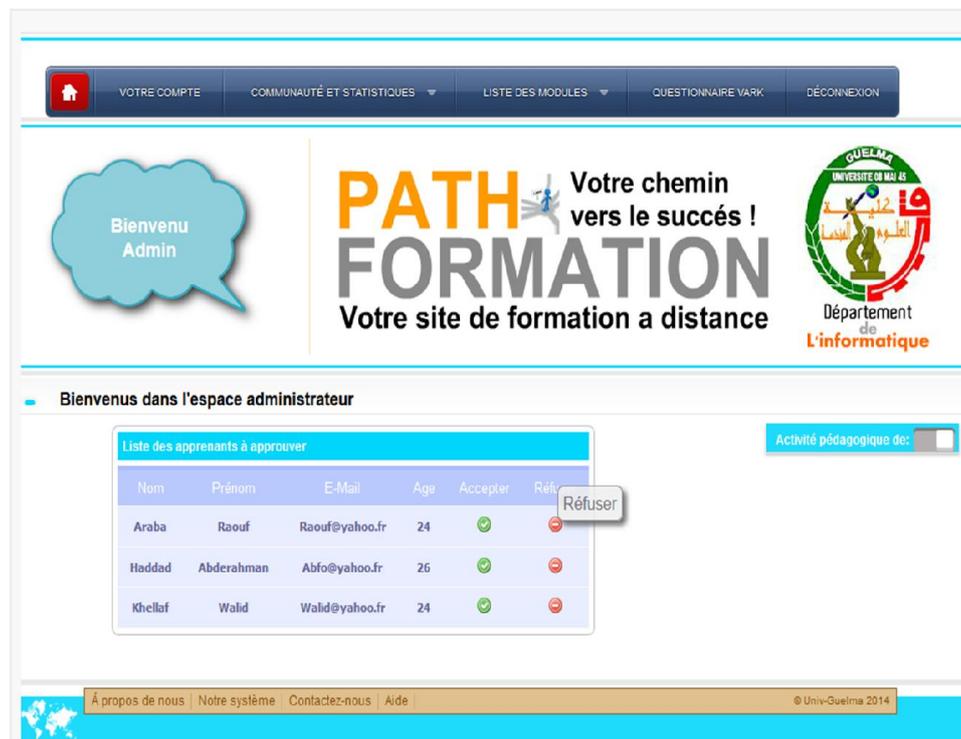


Figure 5.6 : Liste des nouveaux apprenants.

- **Apprenant accepté** : c'est un lien qui permet d'afficher un fichier contenant la liste des apprenants membres avec une possibilité de suppression.



Figure 5.7 : Liste des apprenants acceptés.

- **Nouveaux enseignants** : c'est un lien qui permet d'afficher un fichier contenant la liste des nouveaux enseignants inscrit dans le système et qui attendent leurs validations.

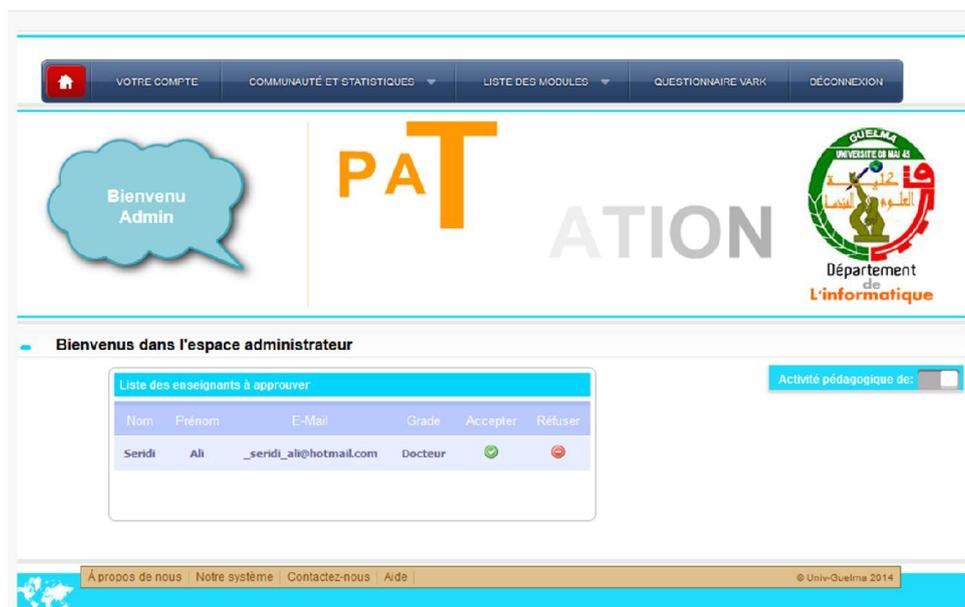


Figure 5.8 : Liste des nouveaux enseignants.

- **Enseignant accepté** : c'est un lien qui permet d'afficher un fichier contenant la liste des enseignants membre dans le système avec la possibilité de supprimé les derniers.



Figure 5.9 : Liste des enseignants acceptés.

- **Statistique** : dans cette page, on présente un graphe distribution des acteurs (apprenants, enseignants).

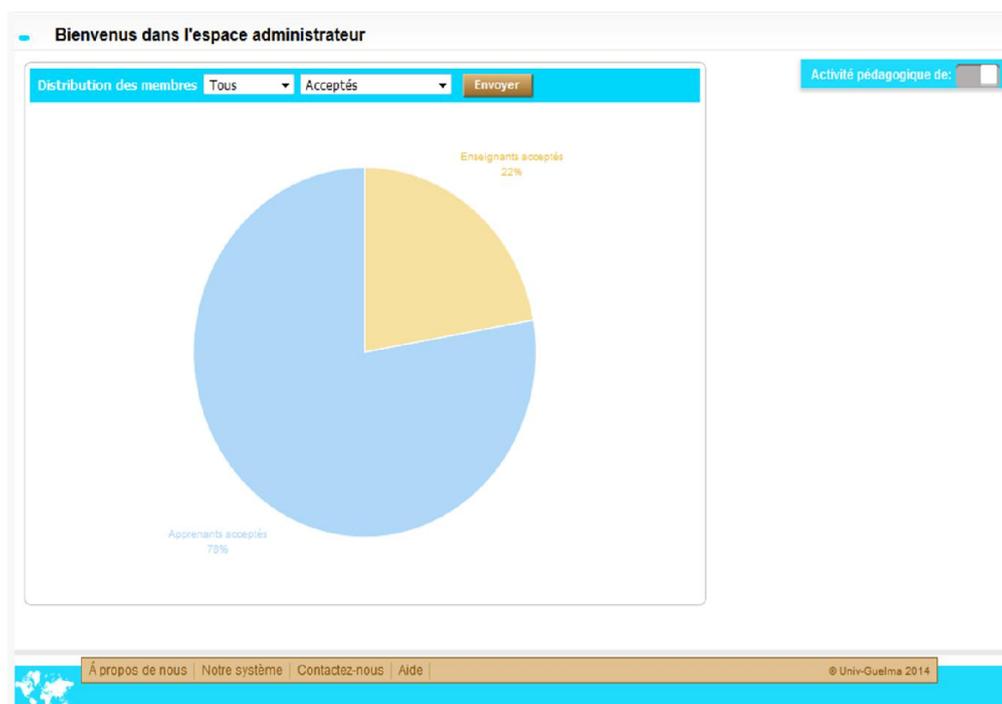


Figure 5.10 : Distribution des membres

- **Liste module :** c'est un lien qui permet d'afficher deux interfaces permettant à l'administrateur d'ajouter des nouveaux modules, une pour le module supervisé. Et l'autre pour les modules non supervisés (voir Figure 5.12).

Nom module	Enseignant	E-Mail	Grade	Supprimer
Algorithmic	Bourbia	Bourbia_riad@hotmail.com	Docteur	⊖
TALN	Bourbia	Bourbia_riad@hotmail.com	Docteur	⊖
Genie logiciel	Guouassmi	Guouassmi_noui@hotmail.com	Docteur	⊖

Figure 5.11 : Liste des modules supervisés

Nom module	Liste des enseignants	Affecter au enseignant	Supprimer
RDF	Enseignant	✓	⊖

Figure 5.12 : Liste des modules non supervisés

- ❖ **Remarque :** il y a un petit espace intitulé « les activités d'enseignants ». Il affiche toutes les traces et les activités d'un enseignant sélectionné.

4.1.2. Interface apprenant :

On résume le fonctionnement de l'apprenant dans son espace par l'architecture suivante :

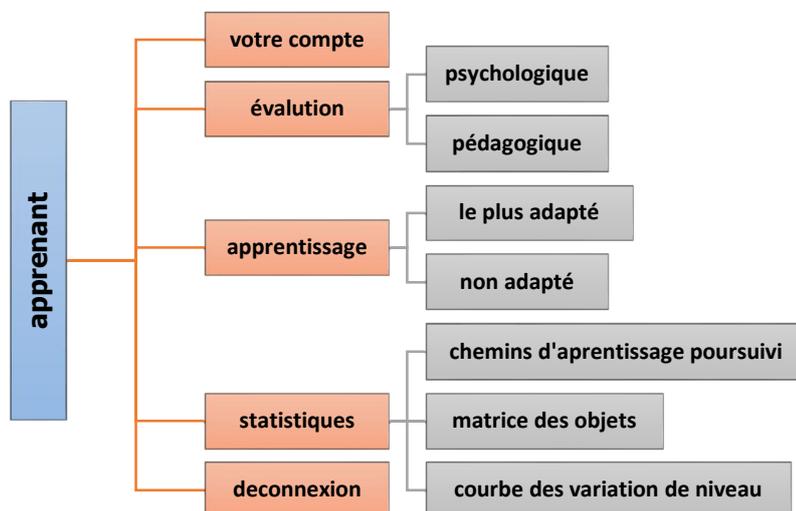


Figure 5.13 : architecture de l'interface apprenant.



Figure 5.14 : Interface d'accès pour les apprenants.

- **Inscription et identification :** L'apprenant saisi ses informations dans le formulaire d'inscription, si un champ n'est pas rempli, une alerte surgit, sinon l'inscription s'effectue. Une fois, l'apprenant est inscrit, il suffit d'introduire son mot de passe et son pseudo pour accéder à son propre espace, et ceci après validation.

Bienvenu

Chers
ETUDIANTS

Voulez vous que vous ameliorez vos connaissances
sans que vous déplacez d'1 mètre de chez vous

UNIVERSITE DE GUELMA
الجامعة الوطنية
Département
de
L'informatique

Inscription d'un nouveau apprenant

Nom:

Prénom:

Pseudo:

Courrier électronique:

Mot de passe: Confirmer mot de passe:

Age:

S'inscrire

À propos de nous | Notre système | Contactez-nous | Aide | © Univ-Guelma 2014

Figure 5.15 : Formulaire d'inscription

- Interface apprenant :

MÉTRER À JOUR VOTRE COMPTE | EVALUATION | APPRENTISSAGE | STATISTIQUES | DÉCONNEXION

Bienvenu Apprenant Raouf

Bienvenus dans votre site de formation à distance

Activité et trace pédagogiques

Un pont cognitif pour vous aider à améliorer vos performance et enrichir vos connaissances

À propos de nous | Notre système | Contactez-nous | Aide | © Univ-Guelma 2014

Figure 5.16 : Interface de l'espace apprenant

C'est une interface qui fournira à l'apprenant tous les outils nécessaires pour suivre un cours. Si l'apprenant est inscrit pour la première fois il doit suivre le parcours suivant :

- **Evaluation**

A. Evaluation psychologique

Ce bouton affiche automatiquement le questionnaire VARK pour aider le système à connaître le style d'apprentissage par l'affichage d'un questionnaire à choix multiple. L'apprenant doit remplir tous les cases ou les champs qui s'affichent obligatoirement pour continuer sa session d'apprentissage ou le système doit afficher tous les styles. Comme le montre la *figure 5.16*.



Figure 5.17 : Evaluation psychologique

B. Questionnaire pédagogique

La validation de la phase évaluation psychologique ouvre une autre page qui contient un QCM associé au module choisi. L'apprenant doit répondre à toutes les questions pour connaître son niveau pédagogique. Comme le montre la *figure 5.18*.



Figure 5.18 : Questionnaire pédagogique.

- **Apprentissage** : ce bouton offre à l'apprenant deux possibilités :

A. Apprentissage adapté

La validation du questionnaire pédagogique ouvre un tableau qui contient le module qui est sélectionné. Après la sélection, le système affiche le cours sous forme de table des chapitres avec une évaluation à la fin de chaque chapitre consulté. (Figure 5.15)

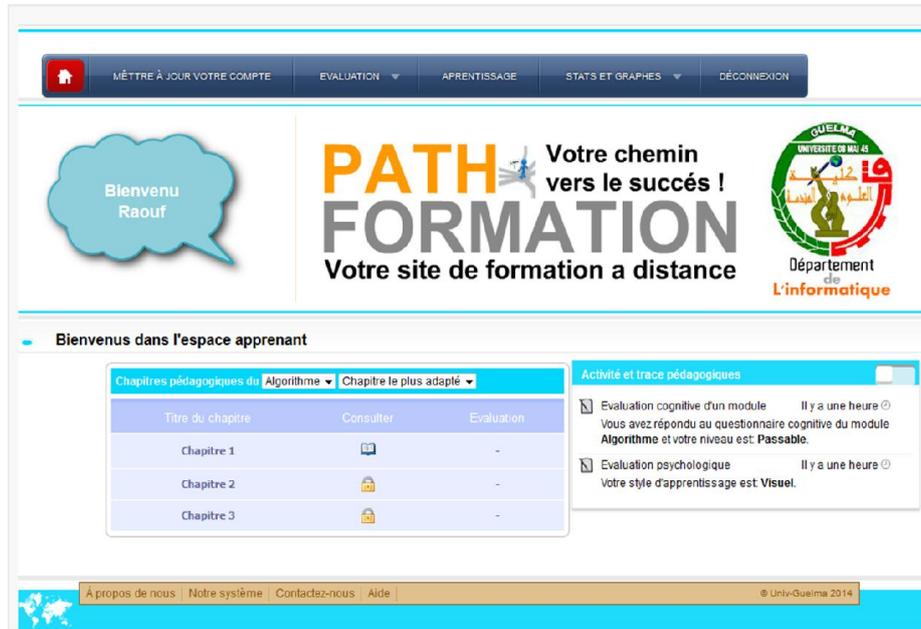


Figure 5.19 : le chemin le plus adapté.

Un seul clic sur consulter, et le system affiche le premier chapitre.

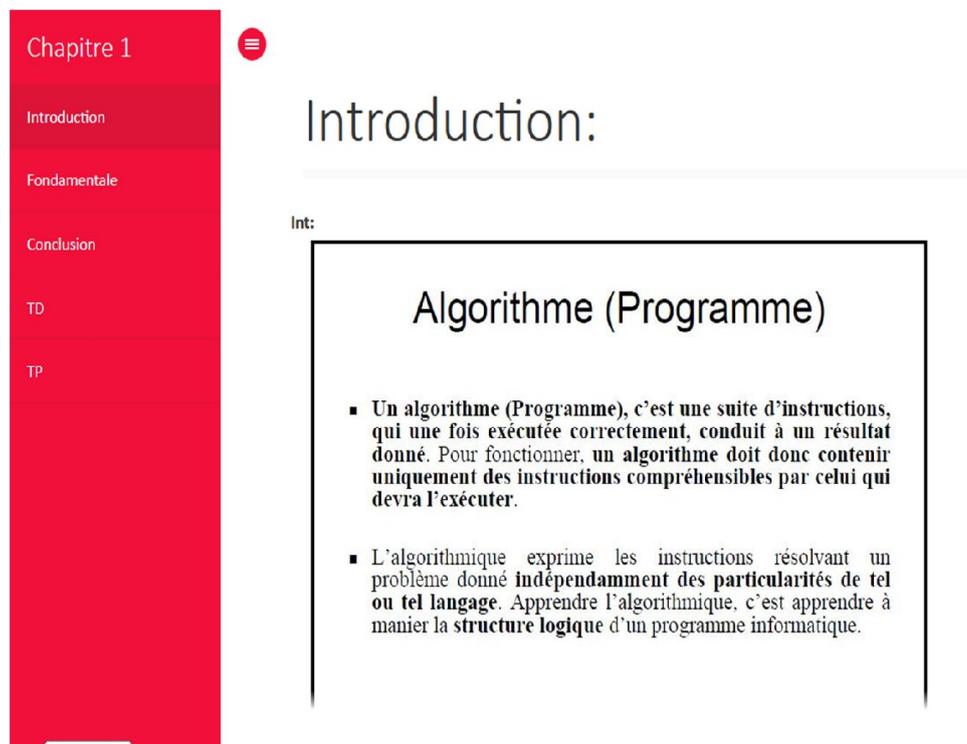


Figure 5.20 : Affichage du chapitre.

B. Chemin non adapté

On peut afficher tous les objets pédagogiques avant évaluation (psychologique, pédagogique), si l'apprenant répond au test psychologique, le system affiche tous les objets pédagogiques liés à son style d'apprentissage.

The screenshot shows a web interface for teachers. At the top, there is a navigation bar with buttons for 'MÉTRÉ À JOUR VOTRE COMPTE', 'EVALUATION', 'APRENTISSAGE', 'STATS ET GRAPHES', and 'DÉCONNEXION'. Below this, a blue speech bubble says 'Bienvenu Walid'. To the right, the text 'Vous êtes ENSEIGNANTS' is displayed, along with the logo of the 'Département de L'informatique' at 'UNIVERSITÉ DE MAI 45'. The main content area is titled 'Bienvenus dans l'espace apprenant' and features a table of learning objects and a calendar for June 2014.

Module	Titre du chapitre	Titre d'objet	Type de block	Niveau	Style d'apprentissage	Consulter
Algorithme	Chapitre 1	Int	Introduction	Passable	Aural (audio)	
Algorithme	Chapitre 1	Int	Introduction	Bien	Aural (audio)	
Algorithme	Chapitre 1	Fon	Fondamentale	Excellent	Aural (audio)	
Algorithme	Chapitre 1	Fon	Fondamentale	Mauvais	Aural (audio)	
Algorithme	Chapitre 1	Fon	Fondamentale	Passable	Aural (audio)	

Calendar: Juin 2014

L	M	J	V	S	D
					1
2	3	4	5	6	7
8	9	10	11	12	13
14	15	16	17	18	19
20	21	22	23	24	25
26	27	28	29	30	

Figure 5.21 : Chemin non adapté.

- **Statistiques**

A. chemin d'apprentissage poursuivi

Cette option permet de présenter sous forme de graphe le chemin d'apprentissage poursuivi par l'apprenant associé chapitre consulté.

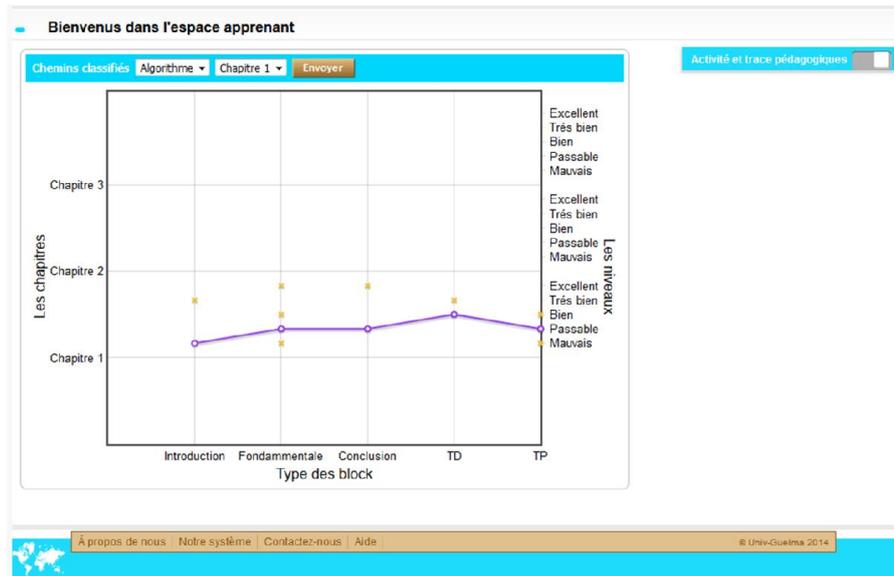


Figure 5.22 : chemin d'apprentissage poursuivi.

B. matrice des objets

Elle présente tous les objets pédagogiques associés au style de l'apprenant.



Figure 5.23 : Matrice des objets pédagogiques.

C. courbe des variations de niveau

C'est une courbe liée au parcours de l'apprenant. Elle affiche ses variations de niveaux pédagogiques au cours d'un apprentissage.

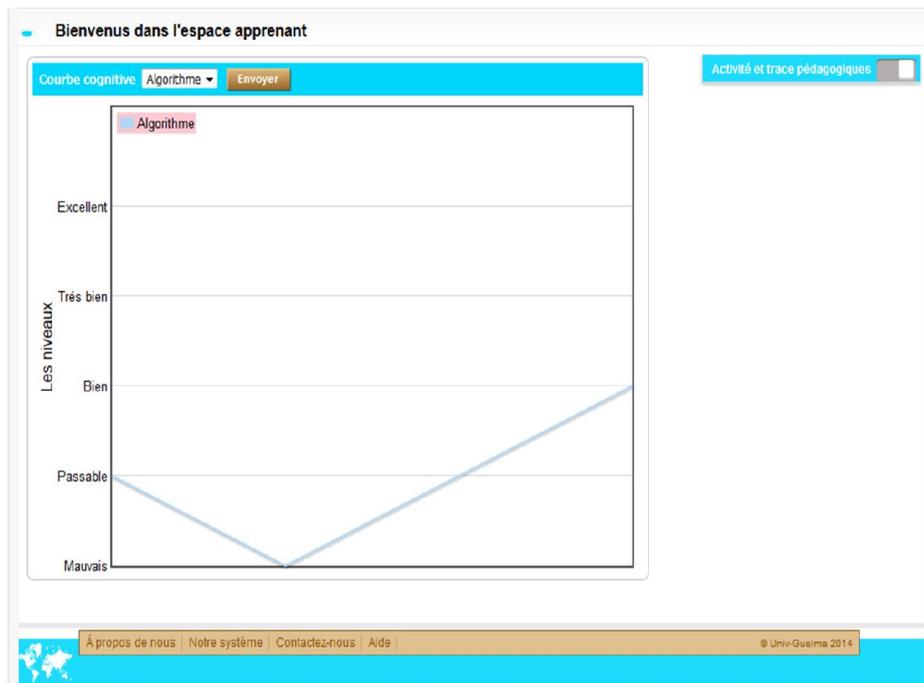


Figure 5.24 : Courbe des variations de niveau.

4.1.3. Interface enseignant

On résume le fonctionnement de l'espace enseignant par l'architecture suivante :

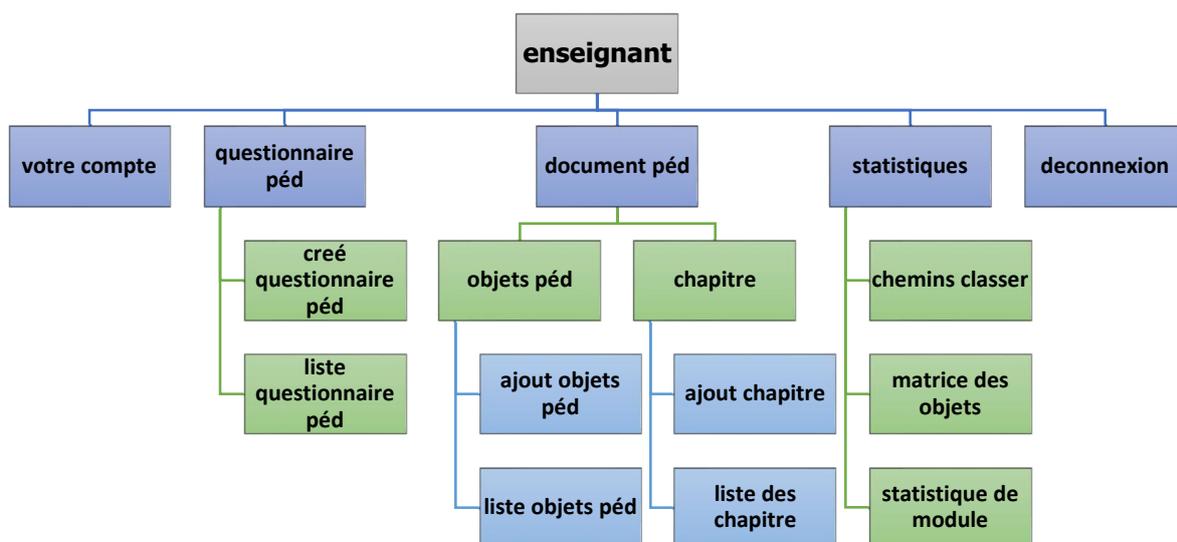


Figure 5.25 : Architecture de l'interface enseignant.

- **Inscription et identification :** L'enseignant saisi ses informations dans le formulaire d'inscription, si un champ n'est pas rempli, une alerte surgit, sinon l'inscription s'effectue. Une fois que l'enseignant est inscrit, il suffit d'introduire son mot de passe et son pseudo pour accéder à son propre espace après validation.

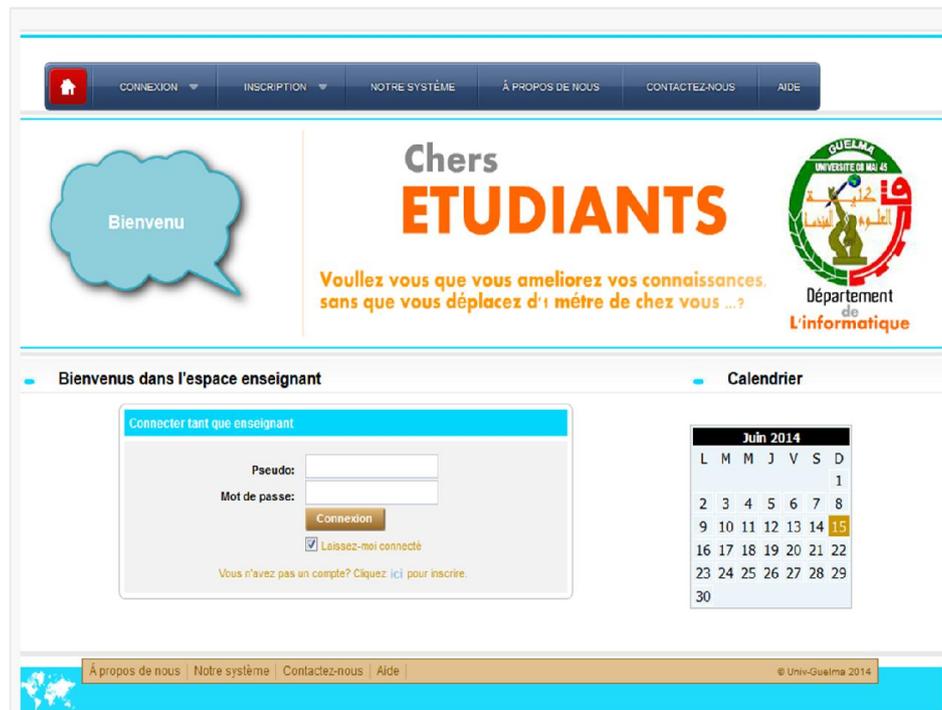


Figure 5.26 : Interface d'accès pour les enseignants.

- **Interface enseignant :**

C'est une interface qui fournira à l'enseignant le droit d'ajouter des chapitres d'un module, Aussi, le droit d'ajouter tous les objets pédagogiques qui composent ce module, et créer les questionnaires associés aux modules et pour l'évaluation.



Figure 5.27 : Interface de l'espace enseignant.

B. Ajouter de nouveaux chapitres

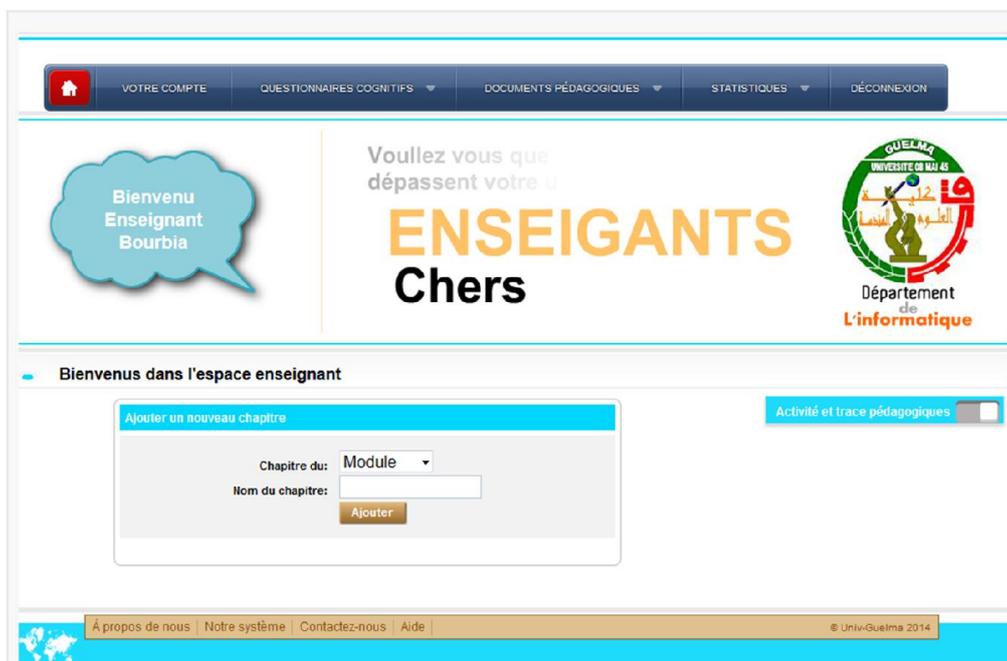


Figure 5.28 : Ajouter un chapitre.

C. Ajouter des objets pédagogiques :

L'enseignant peut déposer son propre cours (sous formes de granules) à l'aide d'un formulaire qui contient : Le nom du module, le nom du chapitre, le bloc qui contient l'objet (introduction, fondamentale, conclusion,..), le nom de l'objet pédagogiques et le niveau de difficulté (mauvais, passable, bien, très bien, excellent):

Figure 5.29 : Formulaire d'ajout des objets pédagogiques.

D. Création de QCM

Ce formulaire est utilisé pour créer les différents questionnaires reliés aux modules.

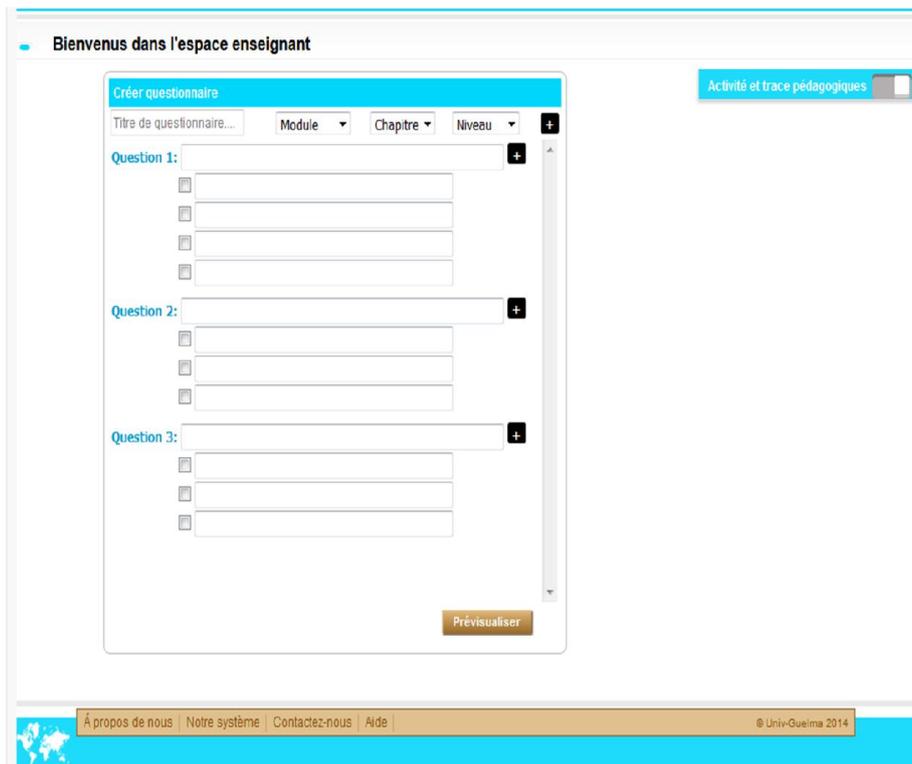


Figure 5.30 : Ajouter des QCM

5. Conclusion

Dans ce chapitre, nous avons pu présenter l'environnement et le processus de développement. Nous avons exposé ainsi le résultat de développement à l'aide des aperçus écran. Nous avons clôturé par une évaluation du travail réalisé.

Conclusion générale

Le travail présenté dans ce mémoire est approprié au domaine du *E-learning* et plus précisément aux travaux des systèmes adaptatifs dynamiques. Nous avons focalisé le travail sur l'optimisation et la classification d'un système adaptatif dynamique, et de rendre son processus de génération du contenu pédagogique à partir des unités plus fines (les objets pédagogiques ou granules) amélioré, avec l'algorithme de colonie de fourmis et un algorithme de classification, pour surmonter quelques problèmes déjà connus dans les systèmes adaptatifs dynamique non optimisés.

Après avoir développé cette plate-forme d'apprentissage à distance, baptisée « PATH FORMATION ». On a employé une technique d'optimisation basée sur l'algorithme de colonie de fourmis, pour la collecte et la recherches des objets ayant les mêmes attributs que les apprenants. Ceci est considéré comme une opportunité pour les apprenants qui ne savent qu'est ce qu'ils veulent ou comment procéder. Cette technique offre beaucoup d'avantages en termes de performances et de temps par rapport aux systèmes non optimisés. L'autre technique utilisée dans notre système, est un algorithme de classification qui permet de classer les chemins d'apprentissage ayant conduit des apprenants à une réussite, de les recommander à nouveaux pour des apprenants de même profils.

Perspectives

Nous avons conçu et réalisé cette première version d'un système adaptatif dynamique basé sur les techniques d'optimisations et la classification, et nous croyons bien avoir atteint les objectifs que nous nous sommes fixés au départ. Vu que les résultats obtenus sont satisfaisants, donc des améliorations et des extensions peuvent être apportées au travail présenté :

- Pour les deux algorithmes employés dans notre système, nous les avons implémentés en combinant le langage de balise PHP, et le langage de requêtes SQL (effectivement une base de données SQL), qui sont deux langages serveurs client, c'est-à-dire, ils prennent beaucoup de temps, qui n'es pas bien aimé dans les systèmes de grand trafic et de grandes masses d'informations qui circulent, donc pour cela, il est bien préférable dans le futur d'utiliser autres langages de console, pour l'implémentation, comme : C,

JAVA, et d'utiliser un autre SGBD, comme par exemple ORACLE, qui gère très bien les base de données volumineuses.

- Faire en apprentissage collaboratif entre des apprenants via un forum ou un chat room .

Bibliographie

- [**A.K.Jain, 99**] A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: a review, *ACM Comput. Surv.* 31 (1999) (3), pp. 264–323.
- [**Alenou ,09**] Alenou atenkeng laurette liliane, développement d'un portail de diffusion des ressources pédagogiques de mathématiques en classe de troisième au cameroun,2009
- [**Amouche, 12**] Amouche hakima, hybridations des métahauristiques , univ tizi-ouzou ,2012.
- [**Ball, 67**] G. H. Ball and d. j. Hall, “A clustering technique for summarizing multivariate data”, *Behavioral Science*, vol.12, pp. 153-155, 1967.
- [**Bourda, 2001**] Bourda Y., (2001), Objets pédagogiques, vous avez dit objets pédagogiques ? Cahier de Gutenberg 39-40, mai 2001, pp 71-79.
- [**Bouri, 08**] Sihem Bouri, optimisation de la production et de la structure d'énergie électrique par les colonies de fourmis, Thèse de doctorat 2008. Sidi Bel-Abbes.
- [**Brusilovsky, 04**] Brusilovsky P. (2004). Adaptive navigation support: From adaptive hypermedia to the adaptive Web and beyond. *Psychology Journal*, 2004, volume 2 (Number 1), pp.7-23.
- [**Cristea, 04**] Cristea A. (2004a). Adaptive Course Creation for All, International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 1, 2004, pp.718.
- [**De Bra., 04**] De Bra P., Nejdl W., (2004) « Adaptive Hypermedia and Adaptive Web-Based Systems», Third International Conference, AH 2004, Eindhoven, The Netherlands, August 23- 26.
- Felder R.M., Silverman L.K. (1988). Learning and Teaching Styles in Engineering Education, *Engr. Education*, 78(7), pp.674-681 (1988).
- [**Felder, 93**] Felder R.M., Felder G.N., Dietz E.J. (2002). The Effects of Personality Type on Engineering Student Performance and Attitudes. *J. Engr. Education*, 91(1), pp.3-17 (2002).
- [**Forgy, 65**] E. W. Forgy, Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications”, *Biometrics*, 21, pp. 768-769, 1965.
- [**Galinier, 99**] Philippe Galinier, Michel Habib et Jin-Kao Hao, Méta-heuristiques pour l'optimisation combinatoire et l'affectation sous contraintes, 1999. [Gardner, 99] Gardner H. (1999). *Intelligence Reframed : Multiple Intelligences for the 21st Century*, Basic Books.

- [**Gérard, 04**] Puimatto Gérard (2004). Petit glossaire à usage commun pour le pilotage et la conduite de projets d'espaces numériques d'éducation. [En ligne]. Les dossiers de l'ingénierie éducative, n° 46.
- [**Gomez, 05**] Gomez M., (2005), Métadonnées pour les ressources d'enseignement et de formation, Journée de l'information AFNOR CN36 GE4.
- [**George, 01**] S. George, "Apprentissage collectif à distance.SPLACH : un environnement informatique support d'une pédagogie de projet", Thèse de doctorat de l'Université du Maine.2001
- [**Grasha, 96**] Grasha A. (1996). Teaching with Style: A practical guide, Pittsburgh: Alliance Publishers.
- [**Hodgins 00**] Hodgins, H.W. (2000). The future of learning objects in D.A. Wiley (Ed.). The Instructional Use of Learning Objects [on-line]. Available: <http://reusability.org/read/>.
- [**Honey et Mumford, 92**] Honey P., Mumford A. (1992). The Manual of Learning Styles, 3 rd Ed., Maidenhead, Peter Honey.
- [**Hui Min Lee et al. 04**] Hui Min Lee C., Cheng Y.W ., Rai S., Depickere A. (2004). What affect student cognitive style in the development of hypermédia learning system? Computer & education, 2004, Elsevier.
- [**IEEE 02**] IEEE Standard for Learning Object Metadata, version électronique : <http://ltsc.ieee.org/>
- [**IEEE, 05**] IEEE (2005). IEEE-Learning Technology Standards Committee (LTSC), WG12: Learning Object Metadata, <http://ltsc.ieee.org/wg12/>, (consulté en décembre 2005).
- [**Iles et al, 08**] Iles N., Chikh A., Mothe J., Choutti M., (2008), Un modèle distribué d'entrepôt pédagogique Utilisation de métadonnées LOM et d'annotations sémantiques, Revue CEMAFORAD, Volume 4
- [**J.MacQueen, 65**] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, *Proc. 5th Berkeley Symp.*, 1965, pp. 281-297.
- [**Kirkpatrick et al., 83**] Kirkpatrick, S., Gelatt, C., et Vecchi, M. Optimisation by simulated annealing. *Science*, tome 220, n 4598, pages 671-680, 1983.
- [**Kolb, 85**] Kolb D.A. (1985). LSI Learning-Style Inventory, McBer & Company.
- [**Lebart, 00**] L. Lebart, A. Morineau, M. Piron, Statistique exploratoire multidimensionnelle, Dunod, 2000.

- [Mayer, 03]** Mayer R.E. (2003). The promise of multimedia learning: using the same instructional design methods across different media, *Learning and instruction* 13 (2003) 125-139, 2003, Elsevier Science.
- [Moallem, 01]** Moallem M. (2001). Applying Constructivist and Objectivist Learning Theories in the Design of a Web-Based Course: Implications for Practice, *Educational Technology & Society* 4 (3), 2001.
- [Olivier Heurtel, 07]** Olivier Heurtel, Php5.2, Développer un site web dynamique et interactif
- [Parkinson et al. 02]** Parkinson A., Redmond J. (2002). Do cognitive styles affect learning performance in different computer media? *ACM SIGCSE Bulletin*, Proceedings of the 7th annual conference on Innovation and technology in computer science education, Volume 34 Issue 3, June 2002
- [Pernin, 04]** Pernin J-Ph, Objets pédagogiques : unités d'apprentissage, activités et ressources ? *Revue Sciences et Techniques éducatives*, Numéro Spécial "Ressources numériques, XML et éducation".
- [Riding et al, 01]** Riding R., Rayner, S. (2001). *Cognitive Styles and Learning Strategies*, David Fulton Publishers London, 2001.
- [Roumane, 12]** Roumane bencherki, *La Méthode de Recherche à Voisinage Variable (RVV)* Université USTO MB, 2012.
- [Sarasin, 98]** Sarasin L. C. (1998). *Learning Style Perspectives: Impact in the Classroom*, Madison: Atwood Publishing.
- [Schwartz, 08]** Pierre Schwartz, *Application d'un algorithme de colonie de fourmis au problème du voyageur de commerce*, 2008.
- [Seridi, 13]** support de cour de Pr.seridi H ,module Analyse des données ,master 1 , année 2012/2013
- [Solnon, 00]** Christine Solnon, *Résolution des problèmes combinatoires*, 2000.
- [Timothy et al, 04]** Timothy J., Hafner W., Mitropoulos F. (2004). eCAD : A Knowledge-Based Course Engineering System, proceedings of the 37th Hawaii International Conference on System Sciences, 2004.
- [Y. De-smet, 05]** Y. De-smet, C. Lamboray, *Multicriteria clustering : à few approaches*, Service de Mathématiques de la Gestion, Université Libre de Bruxelles, 2005.

Webographie

- [1] http://www.kimonte.com/algorithmes_de_colonies_de_fourmis.php.
Dernière date de consultation : le 30/3/2014
- [2] <http://www.vark-learn.com/French/page.asp?p=questionnaire>
Dernière date de consultation : le 3/05/2014
- [3] <http://www.artnetwalk.com/node/16>
Dernière date de consultation : le 07/06/2014
- [4] <http://www.webotheque.fr/glossaire-web/sbdb-mysql.php>
Dernière date de consultation : le 07/06/2014
- [5] <http://charlie-ens.over-blog.com/article-java-script-118420415.html>
Dernière date de consultation : le 07/06/2014
- [6] <http://www.eric-page.info/Projets/chefProjetMW/L-ajax.php>
Dernière date de consultation : le 09/06/2014
- [7] http://clg.becquerel.free.fr/sites/club_info/spip.php?article25
Dernière date de consultation : le 09/06/2014