



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté de génie électrique et d'informatique

Département d'informatique

Mémoire de Fin d'études

En vue de l'obtention du diplôme Master en Informatique

Option : Conduite De Projets Informatique

Thème :

**Une solution basée sur le méta-modèle
pour l'intégration de systèmes**

Cas: E_Learning

Proposé et dirigé par :

M^r : KERBICHE M'hand

M^{elle} : Mansouri Hayat

Réalisé par :

M^{elle} : Ouazar Souhila

Promotion 2013/ 2014

Remerciements

Nos plus vifs remerciements vont à notre promoteur Mr KERBICHE Mohand pour nous avoir proposé ce sujet et nous avoir dirigés tout au long de sa réalisation. Sa compétence, ses critiques et son bon sens nous ont largement aidé à réaliser ce travail et nous lui exprimons ici nous sincères reconnaissances.

Nous tenons aussi à exprimer toute notre gratitude aux membres de jury d'avoir bien voulu accepter de juger notre travail.

Nous tenons à remercier toute personne qui nous a aidés pour l'aboutissement de notre travail.

Dédicaces

A Mes très chers parents.

*A mes frères et mes sœurs et mon neveu Abed arahmane et
mon cousin.*

A tout (es) mes ami (es).

Souhila Ouazar

Dédicaces

A Mes très chers parents.

A mes frères et mes sœurs.

A tout (es) mes ami (es).

Mansouri hayat

Sommaire

Introduction générale	1
I.1 Introduction:	2
I.2 Les Systèmes d'information:	3
I.2.1 Définition de système.....	3
I.2.2 Définition de l'information	3
I.2.3 Systèmes d'information.....	4
I.1.4 Les différentes natures du système d'information	4
I.3 L'interopérabilité	5
I.3.1 Définition de l'interopérabilité	5
I.3.2 Niveau d'interopérabilité.....	6
I.3.3 Définitions des concepts.....	8
I.3.4 Principe de l'interopérabilité en informatique	9
I.3.5 Complexité de l'interopérabilité en informatique :	10
I.3.6 Les données véhiculées dans les interfaces :	10
I.3.7 Interfaces de programmation :	11
I.3.8 Outils d'interopérabilité (.NET Framework) :	11
I.3.8.1 Définition d'un Framework :	11
I.3.8.2 L'outil Framework :	12
I.3.9. Approches de l'interopérabilité	13
I.3.10 Les inconvénients de l'interopérabilité sont les suivants :	13
I.3.11 Les avantages de l'interopérabilité sont les suivants :	14
I.4 Intégration	14
I.4.1 Définition de l'intégration	14
I.4.2 La classification de l'intégration	15
I.4.3 Les niveaux d'intégration	16
I.4.4 Intégration d'applications d'entreprise	17
I.4.5 Avantages de l'intégration	18
I.4.6 Les type d'intégrations	19
I.4.6.1 L'intégration des applications	19
I.4.6.2 L'intégration des données	19
I.4.6.3 L'intégration des anciens systèmes	20


I.4.6.4 L'intégration de la sécurité et des identités	20
I.5 Conclusion	21
II. 1 Introduction	36
II.2 Le modèle	37
II.2.1 définition	37
II.2.2 Notion de l'abstraction	37
II.2.3 Avantages d'un modèle	37
II.2.4 La modélisation	37
II.3 La méta-modèle	39
II.3.1 Définition	39
II.3.2 Ingénierie Dirigée par les Modèles	40
II.3.2.1 Définition	40
II.3.2.2 Concepts de l'architecture MDA	41
II.3.2.3 MDA (Model Driven Architecture)	43
II.3.2.4 IDM vs MDA (Model Driven Architecture)	45
II.3.2.5 Standards d'OMG pour MDA	45
III.1 Introduction	47
III.2 Les différents systèmes d'enseignement assisté par ordinateur	47
III.2.1 Enseignement programmé	47
III.2.2 Enseignement Assisté par Ordinateur (EAO).....	48
III.2.3 Enseignement Intelligent Assisté par Ordinateur (EIAO)	50
III.2.4 Environnement Intelligent d'Apprentissage par ordinateur	52
III.2.5 Environnement Informatique pour l'Apprentissage Humain	53
III.2.6 La formation et l'enseignement en ligne	55
III.2.6.1 Présentation	55
III.2.6.2 Les différents aspects de la formation en ligne	55
III.2.6.3 Quelques exemples de formules d'apprentissage mixte	56
III.2.6.4 Les domaines de la formation en ligne	56
III.3 Notion de la FAD (Formation à Distance) ou EAD (Enseignement à Distance)	57
III.3.1 Origine de la FAD	57
III.3.2 Evolution vers les plates formes informatiques pour la formation à distance.....	58
III.3.3 Modalités de la FAD	59
III.3.3.1 Formation à distance asynchrone	59
III.3.3.2 Formation à distance synchrone	60
III.3.4 Architecture matérielle d'une plate-forme de FAD.....	60

III.4 Vers des Environnements Interactifs d'Apprentissage à Distance (EIAD)	61
III.4.1 Définition des EIAD.....	61
III.4.2 Les situations d'interactions dans les EIAD.....	61
III.4.3 L'apprentissage collectif	62
III.4.3.1 Coopération et collaboration	62
III.4.3.2 Apprentissage coopératif et apprentissage collaboratif	62
III.4.3.3 Apprentissage coopératif	63
III.4.3.4 Apprentissage collaboratif	63
III.4.4 Pratiques avec les plates-formes informatiques pour la formation à distance.....	63
Un certain nombre de plates-formes informatiques pour la formation à distance existent actuellement. Ces plates-formes proposent des outils de communication entre apprenants. Les outils de forum de discussion et de courriers électroniques sont les plus répandus (communication asynchrone), actuellement ces plates-formes mettent davantage l'accent sur la diffusion des cours que sur la mise en place d'activités pédagogiques.	63
III.4.5 Apprentissage collectif soutenu par ordinateur : le domaine des CSCL	64
III.5 Notion d'E-Learning	64
III.5.1 Définition d'E-Learning	64
III.5.2 Evolution de l'e-learning.....	64
III.5.3 Différents aspects de l'e-learning.....	65
III.5.4 Les avantages de l'apprentissage.....	66
III.5.5 Les inconvénients de l'apprentissage en ligne	67
III.6 Conception	67
III.6.1.2 Meta-modèle pour rôle d'enseignant.....	70
III.6.1.3 Meta-modèle pour la gestion d'une matière	71
Pour la gestion de la matière nous trouvons les modules qui constitues de l'ensemble de cours et un cours peut avoir des exercices ou pas; pour chaque module aura un test.....	71
III.6.1.4 Méta-modèle pour gestion de l'inscription d'un apprenant à un module	72
III.6.2 Transformation de diagramme de classe à un modèle relationnelle.....	73
III.6.3 Modèle relationnel conforme à notre Méta-modèle	73
Conclusion	74
VI.1 Introduction	77
VI.2 L'environnement de développement	77
VI.2.1 IDE (NetBeans 8.0).....	77
VI.2.2 Le langage de programmation java :	78
VI.2.3 J2EE :	82
VI.2.4 Représentation de servlet :	82

VI.2.5 Présentation des JSP.....	82
VI.2.6 Représentation de MySQL.....	83
VI.2.8 Apache tomcat.....	86
VI.3 Quelques captures d'écran de notre application.....	88
Premier site web :.....	88
Conclusion générale	Erreur ! Signet non défini.
Bibliographie :	92

Tables des illustrations

Figure I. 1 Représentation des sous-systèmes d'une organisation.....	7
Figure I. 2: Les trois niveaux d'intégration d'un EAI.....	31
Figure II. 1 Les trois perspectives de la modélisation.....	36
Figure II. 2: l'Architecture à quatre Niveaux modélisation par l'OMG	38
Figure II. 3: 4niveaux de modélisation dans un contexte de bases de données relationnelles.	39
Figure II. 4: le cycle de développement en Y	44
Figure III. 1: Les composants d'un système d'enseignement intelligent.....	52
Figure III. 2: Schéma récapitulatif des systèmes d'enseigne	55
Figure III. 3: Schéma général des plates-formes pour la formation à distance.....	63
Figure III. 4: Architecture matérielle d'une plate-forme de FAD.....	65
Figure III. 5: Les situations d'interactions dans les EIAD.....	66
Figure III. 6: Méta- modèle pour le rôle de l'administrateur.....	69
Figure III. 7: Meta-modèle pour rôle d'enseignant.....	71
Figure III. 8: Meta-modèle pour la gestion d'une matière.....	72
Figure III. 9: Méta-modèle pour gestion de l'inscription d'un apprenant à une matière	73
Figure III. 10: diagramme de classe rôle de l'administrateur	73
Figure VI. 1: Aperçu NetBeans (IDE)	76
Figure VI. 2: Aperçu MySQL	83
Figure VI. 3 : Aperçu fichier XML.....	85
Figure VI. 4: Apache tomcat.....	86
Figure VI.5: Page d'accueil du 1 ^{er} Site	88
Figure VI.6: Liste des formateurs du 1 ^{er} Site	88
Figure VI.7: Page d'accueil du 2 ^{eme} Site	89
Figure VI.8: Liste des formateurs du 2 ^{eme} Site	89
Figure VI.9: Page d'accueil d'application d'intégration.....	90
Figure VI.10: Liste des formateurs d'application d'intégration.....	90



Introduction Générale

Introduction générale

Au fil des années, les environnements informatiques sont devenus plus complexes et plus hétérogènes en raison de la diversité des besoins des clients et des innovations grandissantes de l'industrie informatique. Les infrastructures publiques et privées du secteur informatique sont de plus en plus multiplateformes, multifournisseurs et distribuées à grande échelle.

L'évolution des nouvelles technologies de l'information et de la communication et entre autres l'Internet et les systèmes d'informations distribués a permis à un nombre important de cette communauté de partager des sources d'informations hétérogènes et d'une grande masse.

La cohabitation entre des applications et des technologies différentes est devenue une activité critique du point de vue de la garantie de la qualité des services et de l'efficacité organisationnelle et économique des entreprises. Dans ce contexte, nombreux sont les professionnels de l'informatique considérant que l'interopérabilité est un aspect aussi important que la sécurité et la fiabilité pour la gestion de leurs systèmes informatiques.

La consultation, la mise à jour et la recherche d'informations par les différents utilisateurs nécessitent des outils et des moyens spécifiques se basant sur la standardisation de l'univers de discours. Ceci devient indispensable ces dernières années. Pour remédier à l'insuffisance de gestion de l'information, les départements de technologies d'information des organisations ont opté pour des solutions d'intégration. Notre solution se base sur la construction d'un méta-modèle pour l'intégration des applications d'E-Learning (Formation en ligne). Notre travail s'articule autour de quatre chapitres qui sont les suivants:

- ✚ Le premier chapitre est consacré pour les généralités sur les systèmes d'information, l'interopérabilité et l'intégration.
- ✚ Nous abordons dans le deuxième chapitre la notion du MDA, IDM, Méta-modèle.
- ✚ Le troisième chapitre consiste en la présentation d'évolution l'enseignement en informatique, E-Learning, la FAD. Puis passer à l'analyse et la conception.
- ✚ Le quatrième chapitre : « Réalisation » consacré à la réalisation et l'implémentation de l'application.

Chapitre I :
Généralités sur
l'interopérabilité et
l'intégration des systèmes.

I.1 Introduction:

Pour l'utilisateur final, la question de l'interopérabilité en matière de technologie de l'information et de la communication (et plutôt celle de la non-interopérabilité, l'interopérabilité n'étant pas « visible » lorsqu'elle fonctionne bien) est le plus souvent ressentie comme un obstacle technique à l'usage d'un produit ou d'un service. Cet obstacle est d'autant plus mal ressenti que dans un monde comme le monde Internet, l'interopérabilité se veut totale, permettant tous les échanges de contenus de toute nature que ça soit information ou service.

L'intégration est souvent considérée comme allant plus loin que l'interopérabilité, en forçant une certaine dépendance fonctionnelle des applications. Alors que des systèmes interopérables peuvent fonctionner indépendamment, un système intégré peut perdre certaines fonctionnalités si certains de ses services sont interrompus. Un système intégré est donc composé d'applications interopérables mais des applications interopérables ne réalisent pas nécessairement un système intégré. En complément de ces deux vues systémiques, certains systèmes peuvent n'être que compatibles. Ces systèmes n'interfèrent pas directement entre eux. Cela n'implique donc pas qu'ils soient capables d'échanger des services. Des systèmes interopérables sont, par définition, compatibles, au moins en partie, mais l'inverse ne l'est pas.

L'interopérabilité est un moyen d'assurer l'intégration. La différence entre intégration et interopérabilité est clarifiée par le standard ISO 14258 (1999) qui considère que des modèles peuvent faciliter : (1) l'intégration quand il existe un standard ou une représentation pivot qui les formalise ; (2) l'unification quand il existe un méta-modèle commun assurant une équivalence sémantique entre eux ; (3) la fédération lorsque les modèles existent par eux-mêmes mais que des correspondances entre les concepts qu'ils modélisent peuvent être définis à un niveau ontologique pour formaliser la sémantique de leur interopérabilité.

Problématique :

En informatique, il s'agit de composants qui fonctionnent ensemble mais qui n'ont pas été conçus pour cela, par exemple parce qu'ils sont écrits dans des langages différents ou parce qu'ils utilisent des représentations de données différentes (concernent les normes pour présenter, stocker, échanger, traiter et communiquer les données via les moyens informatiques). Ces composants peuvent être des outils logiciels ou des composants matériels qui peuvent communiquer entre eux. Dans notre étude, on ne s'intéresse qu'à l'interopérabilité au niveau des composants logiciels. Théoriquement, la solution au problème de l'intégration au niveau des composants logiciels consiste à trouver une stratégie afin de faire communiquer des logiciels, ou des composants, différents. Plusieurs stratégies peuvent être envisagées pour résoudre un tel problème.

I.2 Les Systèmes d'information:

I.2.1 Définition de système

Un système est un ensemble d'éléments interagissant entre eux selon certains principes ou règles. Un système est déterminé par :

- la nature de ses éléments constitutifs ;
- les interactions entre ces derniers ;
- sa frontière, c'est-à-dire le critère d'appartenance au système (déterminant si une entité appartient au système ou fait au contraire partie de son environnement).

I.2.2 Définition de l'information

L'information est un facteur clé de succès de toute organisation, d'où l'importance du système d'information mis en place. Elle ne prend son intérêt, par-delà le système de communication et de traitement, que lorsqu'elle rencontre l'utilisateur intéressé. Ce qui est important, c'est la bonne information au bon moment pour le bon utilisateur. L'information pour être efficace, doit aller du producteur c.-à-d. de la source, vers ses utilisateurs potentiels. Mais, il n'y a rien de plus indéfinissable et multiforme que l'information. On la définit comme des données enregistrées, classées, organisées pour avoir une signification dans un cadre déterminé. L'information prend une signification et une valeur propre au contexte et à l'instant.

I.2.3 Systèmes d'information

SI est un ensemble organisé de ressources (matériels, logiciels, personnel, données et procédures) qui permet de regrouper, classifier, traiter et diffuser de l'information dans un environnement donné.

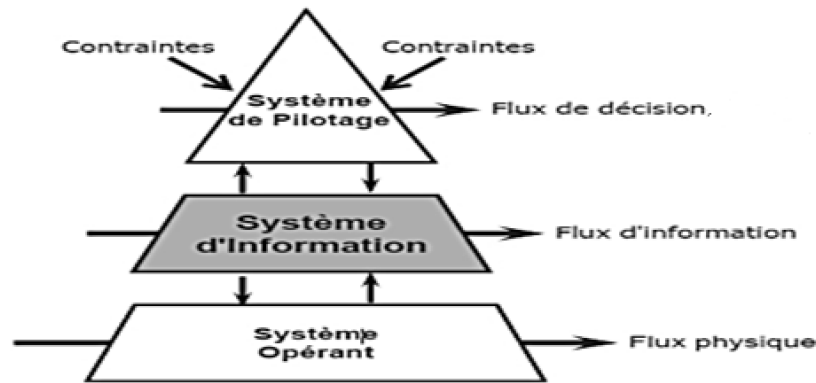


Figure I. 1 Représentation des sous-systèmes d'une organisation.

Le système d'information est un sous-système de l'organisation qui s'occupe de collecter, stocker, traiter et partager l'information dans le système opérant et dans le système de pilotage.

- Dans le système opérant, l'information est une vision actuelle des données de l'entreprise (bons de commandes, facture, catalogue de produit). Cette information est très détaillée.
- Dans le système de pilotage, l'information est plus synthétique car elle doit permettre la prise de décision (par exemple liste des 10 produits les plus vendus en Décembre 2008)

Le SI doit apporter à tous les acteurs de l'organisation dans l'entreprise, l'information dont ils ont besoin pour agir et décider. Le SI est une représentation de la réalité.

I.1.4 Les différentes natures du système d'information

Système d'information et finalité de la chose : Né dans les domaines de l'informatique et des télécommunications, le concept de SI s'applique maintenant à l'ensemble des organisations, privées ou publiques. Le terme système d'information (ou SI) possède les significations suivantes :

- Un ensemble organisé de ressources (personnel, données, procédures, matériel, logiciel, ...) permettant d'acquérir, de stocker, de structurer et de communiquer des informations sous forme de textes, images, sons, ou de données codées dans des organisations. Selon leur finalité principale, on distingue des systèmes d'information supports d'opérations (& traitement de transaction, contrôle de processus industriels, supports d'opérations de bureau et de communication) et des systèmes d'information supports de gestion (aide à la production de rapports, aide à la décision...).
- Un système ou sous-système d'équipements, d'informatique ou de télécommunication, interconnectés dans le but de l'acquisition, du stockage, de la structuration, de la gestion, du déplacement, du contrôle, de l'affichage, de l'échange (transmission ou réception) de données sous forme de textes, d'images, de sons, et/ou, faisant intervenir du matériel et des logiciels.
- Un SI est un réseau complexe de relations structurées où interviennent hommes, machines et procédures qui a pour but d'engendrer des flux ordonnés d'informations pertinentes provenant de différentes sources et destinées à servir de base aux décisions.
- Un SI est un ensemble d'éléments matériels ou immatériels (hommes, machines, méthodes, règles) en interaction transformant en processus des éléments (les entrées) en d'autres éléments (les sorties)

I.3 L'interopérabilité

I.3.1 Définition de l'interopérabilité

L'interopérabilité c'est la capacité technique que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer et opérer ensemble, sans ambiguïté, sans conflit de systèmes ou de contenu.

Interopérabilité des systèmes d'information se réfère à la capacité de partager des informations dans des environnements informatiques distribués, notamment

- ⊕ Pour rechercher et obtenir des informations, quand ils sont nécessaires, indépendamment de leur localisation physique.

- ⊕ Pour comprendre les informations découvertes, quelle que soit la plate-forme qui les supportent, locale ou distante.

I.3.2 Niveau d'interopérabilité

→ Interopérabilité au niveau matériel

L'interopérabilité au niveau matériel est le problème le plus ancien dans le monde de la communication entre les composants informatiques ; elle apparaît de plus en plus cruciale dans le domaine des réseaux informatiques. Ce type d'interopérabilité concerne la communication entre les ordinateurs, dans un réseau informatique ou avec le réseau global. La présence des protocoles de communication et des architectures de réseaux peuvent alourdir ce problème. Les frontières entre la compatibilité au niveau matériel et logiciel sont très minces, notamment à cause du fait que des parties logicielles peuvent interférer avec la communication des composants matériels

→ Interopérabilité au niveau logiciel

Au niveau logiciel, l'interopérabilité concerne des composants logiciels qui doivent travailler ensemble. Il s'agit de faire coopérer les applications/composants écrit(e)s dans des langages de programmation différente, ou de les réutiliser au sein d'une nouvelle application.

Et dans ce niveaux on peut distinguer 03 domaine d'interopérabilité :

⊕ L'interopérabilité technique

L'interopérabilité technique désigne le recours à la définition et l'utilisation d'interfaces technologiques, des normes et des protocoles, en vue de créer des systèmes d'information collaboratifs fiables, efficaces et performants capables d'échanger l'information.

Sur un plan technique, l'interopérabilité se réalise à trois niveaux techniques complémentaires

1. Une description des ressources avec sémantiques communes issues de différents jeux de métadonnées standardisées.
2. Un contexte générique d'implémentation de ces descriptions dans des langages structurés standardisés, interprétables par les machines.
3. Des protocoles informatiques d'échange de ces données normalisées.

⊕ **L'interopérabilité sémantique:**

L'interopérabilité sémantique consiste à faire en sorte que la signification des informations échangées ne soit pas perdue dans la procédure d'interopérabilité et qu'elle soit préservée et comprise par les personnes, les applications et les institutions concernées (les formats des messages, la structuration, la sémantique des constituants). Elle nécessite que l'interopérabilité technique soit effective. L'interopérabilité sémantique se base sur trois principes nécessaires qui sont :

a. Métadonnées

Les métadonnées se sont des informations descriptives sur les ressources. L'utilisation de métadonnées descriptives et standardisées améliore la recherche des informations pertinentes dans un réseau de ressources.

b. Ontologie

L'ontologie est une description formelle des concepts, des rôles et des relations qui existent pour un agent ou une communauté d'agents. Elles fournissent une compréhension commune d'un domaine qui peut être communiquée, tout en jouant un rôle majeur dans les échanges d'information.

c. Médiateurs

Un médiateur est un adaptateur de données situées sur un réseau entre un client et un serveur de données (le client peut être une autre base de données). Par ailleurs, un médiateur est un composant logiciel qui résout les conflits schématiques et sémantiques

⊕ **L'interopérabilité syntaxique**

Concerne le format des données. L'interopérabilité syntaxique constitue le premier niveau et concerne le format de représentation des connaissances. Le fait qu'une ressource soit exprimée ou puisse être traduite sous un format standardisé ou normé conduit vers l'interopérabilité syntaxique

➔ **Interopérabilité au niveau des entreprises**

Un troisième genre d'interopérabilité concerne les composants matériels et logiciels qui doivent coopérer au sein des applications d'une même entreprise. Ce niveau d'interopérabilité est incontournable pour assurer la pérennité économique de l'entreprise. A ce stade, l'interopérabilité peut être vue comme « la capacité des entreprises à structurer,

formaliser et présenter leurs connaissances et savoir-faire, afin d'être en mesure de les échanger ou de les partager ». Le problème d'interopérabilité des entreprises passe souvent par celui de ses systèmes d'information, étant donné la forte dépendance entre les entreprises et leurs systèmes d'information.

I.3.3 Définitions des concepts

Norme : (du latin norma, équerre, règle) désigne un état habituellement répandu moyen considéré le plus souvent comme une règle à suivre. Une norme est une entente consensuelle établie par les partenaires d'un organisme international reconnu officiellement pour fournir une solution normative à des problèmes communs de description, d'indexation et de classification des informations, des processus ou des services.

Le processus d'élaboration d'une norme est appelé normalisation.

Standard: un ensemble de recommandations développées et préconisées par un groupe représentatif d'utilisateurs. C'est par exemple le cas des RfC (Request For Comments) de l'IETF ou des recommandations du W3C, de l'IEEE, etc.

Format ouvert : Un format ouvert, ou spécification ouverte, est un format de données interopérable et dont les spécifications techniques sont publiques et sans restriction d'accès ni de mise en œuvre, par opposition à un format fermé. Son mode de représentation a été rendu public par son auteur et qu'aucune entrave légale ne s'oppose à sa libre utilisation.

Le secteur public ne peut adopter un format fermé pour ses échanges avec les administrés ou les usagers sans imposer l'utilisation d'un logiciel particulier, ce qui revient à donner à son éditeur une position dominante sur le marché, qui pourrait alors en abuser, et rendrait ce secteur dépendant de cet éditeur pour l'accès à ses propres données et outils.

Dans Les principaux formats ouverts on trouve

- Compression : gzip, LZMA, 7z (ZZip)
- Image : png, svg, odg (OpenDocument Drawing), bmp, jpeg
- Tableur : OpenDocument Calc (ods)
- Texte brut : ASCII (txt)
- Texte formaté : odt (OpenDocument Text), htm ou .html (Hypertext Markup Language),

- XHTML(.xhtml), Feuilles de style en cascade (.css)
- Un format fermé est un format de données qui se définit par opposition au format ouvert.

Format fermé : est lorsqu'il n'est pas libre d'utilisation, soit que ses spécifications ne sont pas connues et que donc, on ne peut pas créer de logiciel qui sache lire ou écrire ce type de format, soit que des restrictions légales à son utilisation existent (à travers une licence d'utilisation). L'avantage pour le propriétaire d'un tel format est que celui-ci n'est parfaitement lisible qu'avec un logiciel particulier ou une famille de logiciels. Ainsi, il augmente le coût de la migration de ses clients vers d'autres solutions logicielles et maintient une emprise sur eux.

Si le format est accepté comme standard, le propriétaire peut obliger ses concurrents à lire et/ou écrire son format, rendant leurs applications plus complexes, tout en s'assurant que leurs performances seront moindres aux yeux des consommateurs. Le format de fichier constitue souvent un enjeu commercial important pour lequel de fortes sommes d'argent sont mobilisées.

Exemples de formats fermés

- Fichiers produits par Microsoft Office : doc (Word), xls (Excel), ppt (PowerPoint), pub (Publisher)
- image : gif, wmf, tiff
- audios : wav, mp3, wma
- codec vidéo : avi (DivX)
- Les formats de compression de données: zip, rar, ace
- Format exécutable : exe

I.3.4 Principe de l'interopérabilité en informatique

L'interopérabilité en informatique est une capacité juridique offerte au citoyen d'utiliser l'informatique sans se soucier d'aspects techniques. Cette capacité doit permettre à tout citoyen, sans préjudice, d'obliger par les ordres, qu'il donne à un ordinateur, par l'intermédiaire d'un programme, de toute nature, de se coordonner, de coopérer et d'être piloté par tout autre programme d'une autre nature quel que soit le lieu, le matériel et le langage

utilisé. Il doit le faire si le service attendu l'exige. Le service rendu doit être de même valeur satisfaisante qu'il eut été fait par l'un ou l'autre des programmes, dès lors que le service correspond, sans obstacle contractuel ni obstacle technique.

I.3.5 Complexité de l'interopérabilité en informatique :

L'informatique pose le problème de l'interopérabilité en des termes nouveaux. Elle met en évidence certaines contradictions entre les intérêts commerciaux d'entreprises fournissant produits et services, et les exigences nouvelles des consommateurs de ces produits et services. Du fait des outils informatisés, de l'expertise acquise par des groupes d'utilisateurs, de la communication facilitée, l'interopérabilité devient une problématique plus concrète aux yeux d'un nombre grandissant de personnes, qui en comprennent mieux les tenants et aboutissants notamment les enjeux du choix et de la protection des données.

Ce mouvement est vu comme une avancée démocratique par les partisans d'une interopérabilité « ouverte », mais cet avis n'est pas partagé par tous. Nombre d'entreprises défendent à l'inverse un modèle plus classique où l'interopérabilité reste le fruit de l'initiative privée et subit un contrôle strict. De par les enjeux qui lui sont aujourd'hui liés, dans les domaines du travail ou dans la sphère privée par exemple, l'interopérabilité informatique va certainement jouer un rôle de catalyseur des changements futurs, quels qu'ils soient.

I.3.6 Les données véhiculées dans les interfaces :

En pratique, l'interopérabilité touche tous les domaines de l'informatique. Ce sont les règles de cohérence des données véhiculées qui gouvernent l'interopérabilité. Les données de référence employées par plusieurs applications sont généralement celles qui pilotent l'interopérabilité.

Dans des contextes où coexistent les données structurées (celles des bases de données) et les données non structurées (les documents, textes, images), on considère généralement aujourd'hui que les données communes sont constituées par des « métadonnées ». À l'origine, c'étaient des mots clés qui étaient introduits dans les langages de balisage tels que SGML, HTML. Le langage XML est aujourd'hui considéré comme le langage qui permet d'accéder à l'ensemble des ressources informatiques par le Web, en utilisant ces métadonnées, dans le cadre RDF défini par le W3C en 1999. En pratique, l'interopérabilité repose sur la

description de Schémas XML, qui permettent de vérifier que les documents XML se conforment aux contraintes d'un schéma. La souplesse des schémas XML vient de ce qu'il est possible de définir des espaces de noms et des types de données pour caractériser les éléments de données échangés.

Autant l'interopérabilité est nécessaire en intelligence économique pour les gouvernements et les entreprises en réseau, autant l'utilisation sans précaution des métadonnées dans les composants informatiques peut comporter des risques de pertes d'informations pour les communautés qui les emploient, souvent sans avoir conscience de leur importance stratégique. La tenue de registres de métadonnées conformément aux règles normatives (ISO/CEI 11179) limite les risques liés aux métadonnées.

I.3.7 Interfaces de programmation :

En informatique, une interface de programmation (souvent désignée par le terme API pour *Application Programming Interface*) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

En informatique l'utilisation des API est importante pour l'interopérabilité. Par exemple, la spécification J2EE pour le langage de programmation Java comporte de nombreux types d'API, qui véhiculent des métadonnées (des données structurées exemple: bases de données ou non structurées exemple : photo, fichier ...etc. Ces API peuvent s'appliquer à différents types de ressources informatiques (bases de données) ou applications (Progiciel de gestion intégré).

I.3.8 Outils d'interopérabilité (.NET Framework) :

I.3.8.1 Définition d'un Framework :

En programmation informatique, un *Framework* est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de

tout ou d'une partie d'un logiciel (architecture). Un *Framework* se distingue d'une simple bibliothèque logicielle principalement par :

- son caractère générique, faiblement spécialisé, contrairement à certaines bibliothèques ; un *Framework* peut à ce titre être constitué de plusieurs bibliothèques chacune spécialisée dans un domaine. Un *Framework* peut néanmoins être spécialisé, sur un langage particulier, une plateforme spécifique, un domaine particulier : Reporting, mapping, etc.
- le cadre de travail (traduction littérale de l'anglais : *Framework*) qu'il impose de par sa construction même, guidant l'architecture logicielle voire conduisant le développeur à respecter certains *patterns* ; les bibliothèques le constituant sont alors organisées selon le même paradigme.

Les Framework sont donc conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web, des middlewares et des composants logiciels. Les Framework sont acquis par les ingénieurs, puis incorporés dans des logiciels applicatifs mis sur le marché, ils sont par conséquent rarement achetés et installés séparément par un utilisateur final.

I.3.8.2 L'outil Framework :

Les outils fournis dans cette section simplifient et réduisent la quantité de code nécessaire à l'exécution de tâches d'interopérabilité telles que permettre aux clients COM d'utiliser des classes .NET Framework qui dépendent du Common Language Runtime.

Entre autres choses, ces outils convertissent et génèrent des bibliothèques de types, et ils créent et enregistrent des classes et des assemblies. Ils permettent aux développeurs d'utiliser des méthodologies de conception et de programmation existantes, que les types dépendent du Common Language Runtime ou de COM.

Ces outils sont installés automatiquement avec Visual Studio et avec le Kit de développement logiciel Windows. Le meilleur moyen d'exécuter ces outils consiste à utiliser l'invite de commande de Visual Studio ou du Kit de développement logiciel Windows.

I.3.9. Approches de l'interopérabilité

L'interopérabilité nécessite que les communications obéissent à des normes clairement établies et univoques. Ces documents techniques définissent souvent des exigences, parfois accompagnées de recommandations plus ou moins optionnelles. Si la norme est correctement écrite, deux systèmes qui satisfont aux exigences doivent dialoguer ensemble sans souci particulier.

Ils peuvent ainsi évoluer librement sans risque de casser cette possibilité de communication, tant qu'ils respectent la norme définissant leurs interfaces.

1. Les normes ouverts et communautaires
2. Les formats fermés et propriétaires.
3. Entre les deux : les normes à distribution restreinte :

Entre ces deux mondes, il existe également un grand nombre d'organismes plus ou moins ouverts dans la sélection de leurs membres, souvent orientés vers les entreprises et ayant des cotisations ou des droits d'entrées conséquents, dont les publications ne sont pas librement accessibles, mais payantes. C'est le cas de la majorité des organismes d'État, notamment. On peut citer les organismes ISO, ANSI, AFNOR,UIT...

Parfois, c'est une très petite assemblée de personnes, voire une seule, qui décide d'une norme. Elle peut être plus ou moins à l'écoute des suggestions, bien sûr, de ses utilisateurs. Des exemples courants sont RAR (algorithme de compression), PDF (format de document pour l'impression), Java (langage de programmation), Flash (format d'animation pour le Web), etc.

I.3.10 Les inconvénients de l'interopérabilité sont les suivants :

- Une certaine perte d'autonomie de l'exploitant de réseau combinée à des démarches pouvant consommer beaucoup de temps et visant la négociation d'enjeux associés aux méthodes et aux contrats;

- Dans certains cas, le besoin en équipement coûteux en raison d'exigences fonctionnelles supplémentaires (par exemple, équipement routier ou appareils embarqués respectant plusieurs normes);

- Le coût de passage de systèmes non compatibles à des systèmes compatibles (par exemple le renouvellement de systèmes de péage électronique pour être conformes aux nouvelles normes);
- Coûts supplémentaires des mesures de sécurité prises en raison des risques accrus de fraude.

I.3.11 Les avantages de l'interopérabilité sont les suivants :

- Le confort accru pour les voyageurs qui peuvent se servir de leur équipement «domestique» et de leurs propres moyens de règlement lorsqu'ils vont à l'étranger;
- L'économie pour les propriétaires d'automobiles qui n'ont pas à se doter de différents appareils embarqués pour exécuter les mêmes fonctions partout;
- Les soumissions plus concurrentielles présentées dans un vaste marché commun lorsque les exploitants de réseaux lancent des appels d'offres d'équipement; sources multiples plutôt que d'une seule source.

I.4 Intégration

Au fil des années, les environnements informatiques sont devenus plus complexes et plus hétérogènes en raison de la diversité des besoins des clients et des innovations grandissantes de l'industrie informatique. L'intégration des applications dans les entreprises est une question critique du point de vue de l'augmentation de l'efficacité et de la réduction des coûts.

I.4.1 Définition de l'intégration

L'intégration est un processus de combinaison d'éléments qui au premier abord semblent incompatibles ou même conflictuels, mais qui après un peu d'analyse et de resynthèse (menant à la reformulation ou la réorientation), s'avèrent être plutôt complémentaires.

"L'astuce" essentielle permettant l'intégration, (utilisée incomplètement et intuitivement dans le compromis), est que l'on se rend compte que chaque forme concrète d'idée/projet n'est pas l'aspect essentiel, mais une tentative de traduire quelques valeurs "sous-jacentes" plus importantes en une forme concrète adaptée à l'évidence disponible. Le

problème avec la sélection et le compromis est la supposition inconsciente apparemment fausse, que la valeur essentielle d'une idée ou d'un projet réside en sa forme réelle, résultant en des conflits du type "c'est à prendre ou à laisser".

I.4.2 La classification de l'intégration

Il semble y avoir deux domaines principaux d'application pour des processus d'intégration : **conceptuel** et **factuel**. Ceux-ci peuvent parfois être liés l'un à l'autre, par exemple quand l'intégration conceptuelle entre les gens mènent à l'intégration factuelle de leurs activités. Mais l'intégration factuelle peut se produire sans précéder l'intégration conceptuelle, par un processus itératif d'essai et d'erreur, par exemple entre des organismes sans intelligence.

- ➔ L'intégration **conceptuelle** est un moyen de pensée, par lequel l'information partielle ou des aperçus apparemment contradictoires (ou des plans, des hypothèses, des paradigmes, des théories, des projets, des programmes, etc) peuvent être reformulés, afin que ces éléments puissent harmonieusement être combinés en un élément d'information ou un éclairage plus complet, ayant plus de valeur. Le résultat de ce processus combine chaque élément de valeur de l'original, des éclairages partiels et/ou conflictuels. Par conséquent, le résultat de ce processus logique présente toujours plus de valeur et est plus plausible que chaque éclairage séparé, tout du moins si les règles d'intégration sont correctement appliquées.
- ➔ L'intégration **structurelle** est un moyen d'interaction, par laquelle deux (ou plus) systèmes (des organismes, des molécules, etc) effectuent des adaptations structurelles et/ou procédurales dans leur mode de fonctionnement -- le plus souvent par tâtonnement --, pour que tous les éléments impliqués atteignent une meilleure réalisation de leurs "besoins" individuels qu'avant le processus d'intégration. La symbiose, la collaboration, la coopération, la complexification, la socialisation, la synergie constituent ses exemples frappants de ce phénomène. Le processus évolutionnaire cosmique dans son entier peut être décrit en termes d'intégration. Seuls des systèmes intelligents comme les êtres humains sont capables de comprendre une intégration aussi factuelle comme le résultat de l'intégration conceptuelle consciente.

I.4.3 Les niveaux d'intégration

Fusionner deux entreprises ne consiste pas seulement à additionner des chiffres d'affaires et des parts de marché, mais également à mélanger des technologies, des hommes, des méthodes de travail et des cultures. Selon les opérations et le contexte dans lequel elles se situent, l'intégration sera plus ou moins planifiée. La notion de compatibilité peut exister ou non entre les entreprises à intégrer. Ce triptyque renvoie aux différentes manœuvres stratégiques possibles dans le cadre des fusions-acquisitions et plus précisément aux incompatibilités pouvant exister entre les activités des entreprises impliquées. Si la compatibilité entre les organisations apparaît donc comme essentielle, trois niveaux d'intégration émergent de la littérature mettant en perspective une polysémie de l'intégration. Trois niveaux sont donc définis : une intégration stratégique, une intégration organisationnelle et une intégration technologique. Il convient donc de prendre en compte ces trois niveaux d'intégration afin de pouvoir étudier le processus d'intégration des systèmes d'information.

➔ L'INTÉGRATION STRATÉGIQUE

Ce niveau d'intégration stratégique touche directement les systèmes d'information. En effet, la comptabilité et le contrôle, la communication externe, les circuits de distribution ainsi que la gestion des stocks, ont tous en commun une étroite relation avec la fonction système d'information qui permet soit de formaliser directement ces activités, soit d'en supporter le fonctionnement. Les entreprises, indépendantes et autonomes avant l'opération de fusion, ont développé des stratégies spécifiques. Les pilotages de ces stratégies peuvent différer très fortement, rendant nécessaire et impératif le changement d'habitudes et de comportement. L'intégration stratégique est donc un impératif pour les entreprises qui souhaitent se rapprocher.

➔ L'INTÉGRATION ORGANISATIONNELLE

De la même façon, l'intégration organisationnelle implique des difficultés qui touchent les composantes essentielles de l'entreprise. Les composants à intégrer sont principalement la répartition des pouvoirs décisionnels, les modes de fonctionnement ainsi que les structures organisationnelles issues des entreprises en présence. Plusieurs difficultés d'intégration sont recensées. Une multiplicité des réunions et des projets d'intégration émerge du rapprochement des personnels afin de donner une cohérence à la nouvelle entreprise fusionnée. De même, les responsabilités sont diluées et des fonctions peuvent alors être doublées inutilement donnant lieu à des rationalisations de postes et des licenciements. Des équipes de travail informelles se reconstituent et génèrent alors des apparitions de clans au sein de la nouvelle structure. Les

composants à intégrer sont surtout les domaines du contrôle des résultats financiers, de la communication interne et de la planification stratégique. En effet, on assiste à des difficultés.

➔ L'INTÉGRATION TECHNOLOGIQUE ET INFORMATIONNELLE

Le terme intégration apparaît dans cette dimension comme la simple incorporation d'une information au sein d'un système (matériel et/ou logiciel) qui a pour objet de la traiter. En effet, plusieurs processus peuvent utiliser des données communes, telles que les informations relatives aux clients. L'intégration informationnelle au sein d'une entreprise suppose que tous les processus utilisant par exemple les données relatives à un client X aient exactement les mêmes informations relatives à ce client. Si au cours d'un processus (ex. prise de commande d'un client) un acteur est amené à modifier des données (ex. nouvelle adresse du client), cette modification sera ainsi connue des acteurs des autres processus connexes (ex. livraison de la commande à la nouvelle adresse).

I.4.4 Intégration d'applications d'entreprise

L'intégration d'applications d'entreprise (EAI: Entreprise Application Integration) est un genre de Velcro technologique, permettant aux systèmes informatiques de s'adapter à un tel changement. EAI permet aux divers systèmes de se connecter entre eux rapidement pour partager des données, des communications, et des processus, allégeant les silos de l'information qu'infestent beaucoup d'entreprises. Les avantages d'assimiler de nouveaux systèmes sans efforts de programmation prolongés sont évidents après une activité de fusion et d'acquisition. Les solutions d'EAI fournissent une manière de connecter des systèmes de collaborateurs, de partenaires, et d'autres, autant qu'il est nécessaire, et permettent de découpler ces systèmes quand la relation s'est terminée. EAI est, essentiellement, la colle soluble pour la corporation modulaire.

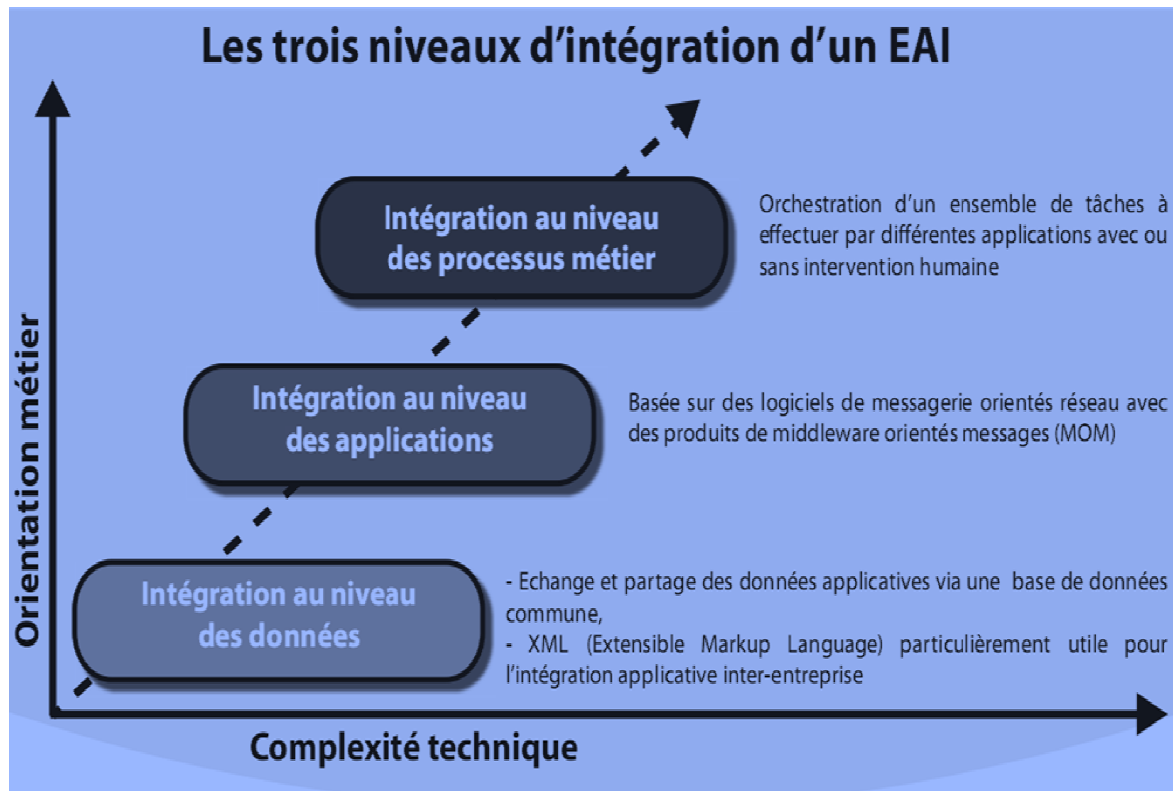


Figure I. 2: Les trois niveaux d'intégration d'un EAI

I.4.5 Avantages de l'intégration

Il y'a plusieurs avantages potentiels liés à l'intégration de systèmes. Les plus importants sont :

- Le calcul des indicateurs : de nombreux indicateurs sont basés sur des numérateurs et des dénominateurs provenant de différentes sources de données. Nous pouvons citer par exemple les taux de mortalité comprenant des données sur la mortalité au numérateur et sur la population au dénominateur, la couverture du personnel et le taux de charge de travail du personnel (données des ressources humaines, et de la population), les taux de vaccination, etc. Pour que ceux-ci soient calculés, vous devez avoir à la fois les données du numérateur et du dénominateur, et celles-ci devraient donc être intégrées dans un seul entrepôt de données. Plus il y a de sources de données qui sont intégrées, plus il peut y avoir d'indicateurs générés à partir du dépôt central.
- Réduire le traitement manuel et la saisie des données : avec des données différentes au même endroit, il n'est plus nécessaire d'extraire manuellement des indicateurs et de les traiter, ou de saisir de nouveau les données dans l'entrepôt de données. L'interopérabilité entre les systèmes de différents types de données (comme les

registres de patients et les entrepôts de données agrégées) permet aux logiciels des sous-systèmes à la fois de calculer et de partager des données par voie électronique. Ceci réduit la quantité d'étapes manuelles impliquées dans la transformation de données, ce qui augmente aussi la qualité des données.

- Il y'a des raisons d'organisation qui justifient l'intégration. Si toutes les données peuvent être traitées par une unité du ministère de la santé, au lieu de l'être à travers divers sous-systèmes gérés par différents programmes de santé, cette unité peut être professionnalisée. Avec un personnel qui a pour seule responsabilité la gestion, le traitement et l'analyse de données, des compétences peuvent être développées et le traitement de l'information peut devenir plus rationnel.

I.4.6 Les type d'intégrations

I.4.6.1 L'intégration des applications

L'intégration consiste à faire fonctionner ensemble des applications différentes.

Principes de l'intégration des applications :

- Communication entre les personnes par le biais des technologies d'accessibilité, des technologies de pontage et d'une large prise en charge des normes.
- Intégration de systèmes variés au moyen d'une gamme étendue de méthodes comme les protocoles de messages, les services Web et les outils de développement.
- Amélioration de l'efficacité des entreprises grâce aux produits tels que Microsoft® BizTalk® Server, Microsoft

Dynamics™, Microsoft Office, SharePoint® Server, Visual

Studio® et Windows Vista™.

- Opportunités pour les développeurs de créer des produits et des technologies innovantes pour Windows® grâce aux initiatives telles que MSDN®, TechNet, CodePlex et SSI

(Shared Source Initiative).

I.4.6.2 L'intégration des données

L'intégration des données concerne la création, le partage et l'accès aux données entre différents environnements informatiques. Apports de l'intégration des données :

- Communication entre les personnes facilitée par le biais des technologies d'accessibilité, les formats de fichiers ouverts, les fonctionnalités avancées et des applications simples d'emploi.

- Plus grand contrôle des données grâce aux technologies de nouvelle génération et notamment aux formats de fichiers Open XML d'Office pour les traitements de texte, les tableurs et les logiciels de présentation de documents.
- Entreprises plus efficaces grâce à la prise en charge des standards des industries verticales pour les secteurs de l'enseignement, de la santé, des finances, de l'automobile et des télécommunications.
- Pontage entre des technologies disparates grâce aux techniques de connexion avec les bases de données, la traduction des données et les services de transformation, aux formats de fichiers standards et à la prise en charge de XML en natif.

I.4.6.3 L'intégration des anciens systèmes

Cette intégration consiste à exploiter au maximum les ressources existantes tout en tirant partie des technologies de nouvelle génération. Les points essentiels :

- Prise en charge des systèmes existants y compris IBM DB2 les mainframes IBM zSeries et les mini-ordinateurs iSeries, les systèmes Novell NetWare et GroupWise, Oracle Fusion, SAP mySAP et Sun ONE.
- Des solutions pour étendre les fonctionnalités disponibles pour les données anciennes, la gestion des annuaires, l'authentification des identités, les files de messages et les implémentations de services Web.
- Amélioration de l'efficacité des entreprises grâce aux produits tels que Microsoft® Host Integration Server, BizTalk® Server, Visual Studio®, Windows Vista™ et Windows Server®.
- Outils pour les développeurs : Fournisseurs de données managées Microsoft ADO.NET pour DB2, prise en charge de COBOL et RPG dans Visual Studio et intégration des services Web.

I.4.6.4 L'intégration de la sécurité et des identités

Cette intégration consiste à garantir des environnements informatique plus sécurisés, plus fiables et respectant mieux la vie privée. Points essentiels :

- Technologies d'authentification d'identités qui permettent la prise en charge de l'authentification Kerberos, des infrastructures à clé publique, des certificats X.509, des jetons SAML 1.1 et des standards des services Web.
- Protection des utilisateurs contre les logiciels malveillants tels que les virus, le phishing et plus encore. Une protection rendue possible par des fonctions de sécurité avancées, la collaboration avec les différents acteurs du secteur informatique la prise en charge des standards de l'industrie.
- Scénarios d'ouverture de session unique entre plateformes et indépendants des langages utilisés. Des scénarios possibles grâce aux technologies fondées sur les protocoles et l'implémentation standardisée de services Web.
- Systèmes offrant une plus grande sécurité et davantage de fiabilité associés à des produits tels que Microsoft® Forefront™ Identity Lifecycle Management, Windows Vista™ et Microsoft Windows Server®.

I.5 Conclusion

L'interopérabilité informatique est un des éléments essentiels pour les réseaux de télécommunication mondiaux comme le téléphone et internet. Ces réseaux hétérogènes utilisent/intègrent des matériels, logiciels et applications variés pour assurer le stockage, le traitement et le transport des données et de l'information. Les travaux issus de nos communautés couvrent une partie des problèmes liés à l'interopérabilité des systèmes. Des avancées importantes ont ainsi été réalisées ces dernières années pour, à la fois, encourager une montée en abstraction « productive » et faciliter cette approche par une analyse des codes eux-mêmes embarqués comme des modèles. La très large plate-forme de travaux introduits ici montre cette richesse en portant un regard commun sur ces approches dans un contexte Système d'Information. Nous ne prétendons pas à l'exhaustivité mais à une vision croisée des travaux menés et des pistes à creuser. L'interopérabilité Les principales étapes de notre premier chapitre sont : (1) définition de système, (2) l'interopérabilité, (3) l'intégration de système. Pour le chapitre suivant on va parler sur la solution qu'on a opté qui est basée sur la méta-modèle.



Chapitre II:
Méta-modèle.

II. 1 Introduction

Le recours à la modélisation est depuis longtemps une pratique indispensable au développement logiciel, car un modèle est prévu pour arriver à anticiper les résultats du codage. Un modèle est en effet une représentation abstraite d'un système destiné à en faciliter l'étude et à le documenter. C'est un outil majeur de communication entre les différents intervenants au sein d'un projet. Chaque membre de l'équipe, depuis l'utilisateur jusqu'au développeur, utilise et enrichit le modèle différemment. En outre, les systèmes devenant de plus en plus complexes, leur compréhension et leur maîtrise globale dépassent les capacités d'un seul individu. La construction d'un modèle abstrait aide à y remédier. Le modèle présente notamment l'atout de faciliter la traçabilité du système, à savoir la possibilité de partir d'un de ses éléments et de suivre ses interactions et liens avec d'autres parties du modèle.

La méta-modélisation, qui est l'activité de construire des méta-modèles, est très utilisée dans le domaine de l'ingénierie des systèmes d'information et particulièrement dans l'ingénierie des modèles et des méthodes.

Généralement, la pratique de la méta-modélisation consiste à spécifier des méta-modèles contemplatifs qui reflètent la structure statique des modèles, c.à.d. les concepts et les liens entre ces concepts. Selon le formalisme utilisé, un tel méta-modèle peut être complété par la spécification de certaines contraintes. En utilisant UML par exemple pour construire un (méta) diagramme de classe, il est possible de compléter cette spécification avec des contraintes exprimées avec le langage OCL. On constate qu'il y a trois perspectives de la modélisation dont illustre la figure suivante :

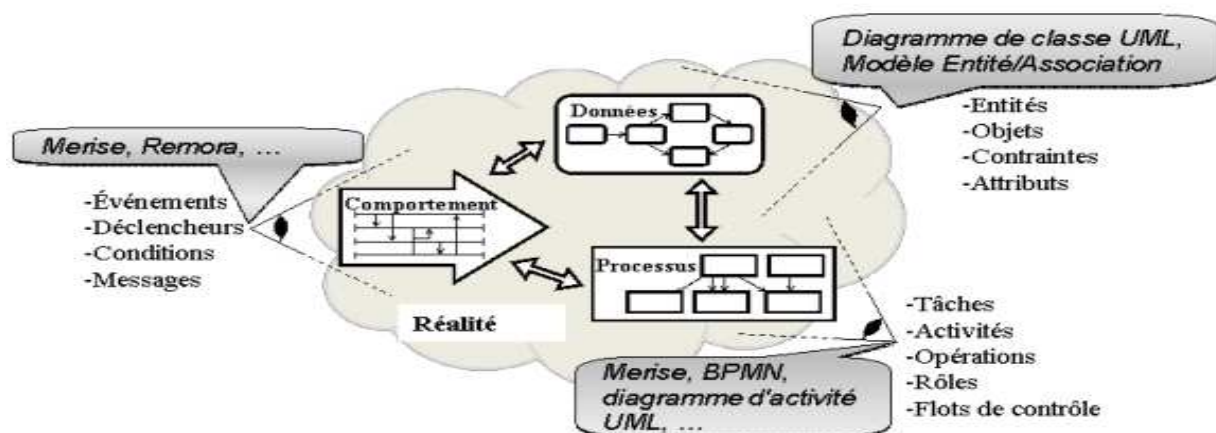


Figure II. 1 Les trois perspectives de la modélisation

II.2 Le modèle

II.2.1 définition

Un modèle est une abstraction de la réalité. Un modèle définit une frontière entre la réalité et la perspective de l'observateur. Ce n'est pas "la réalité", mais une vue très subjective de la réalité. Bien qu'un modèle ne représente pas une réalité absolue, un modèle reflète des aspects importants de la réalité, il en donne donc une vue juste et pertinente. Il résulte d'un processus de modélisation. Le caractère abstrait d'un modèle doit notamment permettre de faciliter la compréhension du système étudié (un modèle réduit la complexité du système étudié), de simuler le système étudié (un modèle représente le système étudié et reproduit ses comportements) Un modèle réduit (décompose) la réalité,

II.2.2 Notion de l'abstraction

Un des piliers de l'approche objet. Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité, en vue d'une utilisation précise. L'abstraction désigne aussi le résultat de ce processus, c'est-à-dire l'ensemble des caractéristiques essentielles d'une entité, retenues par un observateur.

II.2.3 Avantages d'un modèle

Abstrait : Il fait ressortir les points importants tout en enlevant les détails non nécessaires

Compréhensible : Il permet d'exprimer une chose complexe dans une forme plus facilement compréhensible par l'observateur

Précis : Il représente fidèlement le système modélisé

Prédictif : Il permet de faire des prévisions correcte sur le système modélisé

Peu coûteux : Il est bien moins coûteux à construire et étudier que le système lui même

II.2.4 La modélisation

La modélisation est un principe ou technique qui permet d'établir un modèle explicatif d'un phénomène ou comportement en recensant les variables ou facteurs explicatifs et l'importance relative de chacune de ces variables. C'est une conception d'un modèle pour

désigner une étape de construction d'un système d'information. La modélisation Est une étape d'abstraction /conceptualisation en amont de la programmation.

D'une manière générale, La modélisation est la conception d'un modèle. Selon son objectif et les moyens utilisés, la modélisation est dite mathématique, géométrique, 3D, mécaniste (ex : modélisation de réseau trophique dans un écosystème, cinématique...) Elle nécessite généralement d'être calée par des vérifications lesquelles passent par la paramétrisation et le calibrage des modèles utilisés.

Alors l'OMG a introduit l'architecture à quatre Niveaux illustrée dans la Figure comme suit :

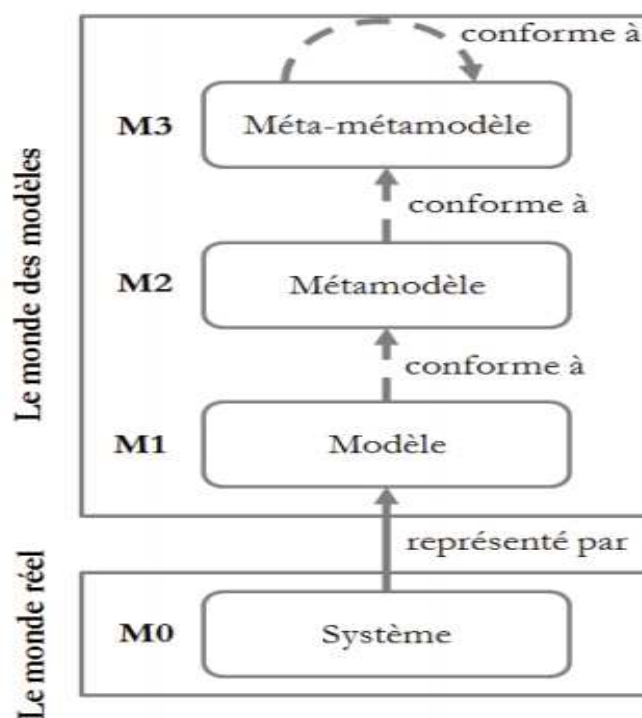


Figure II. 2: l'Architecture à quatre Niveaux modélisation par l'OMG

- ❖ M0 : c'est Le monde réel ou c'est la première couche correspondant à un système en exécution (running). contient des instances d'objets en cours de traitement par le logiciel.
- ❖ M1 : Le modèle c'est abstraction de la réalité. c'est la couche du modèle exécutable renfermant la structure et le comportement du système. C'est à ce niveau que se situent généralement les modèles que nous manipulons quotidiennement dans nos activités de développement de logiciel (ex. : modèle UML d'un service de gestion de

stock, modèle ER d'une bibliothèque, etc.).

- ❖ M2 :c'est la troisième couche de modèles appelés méta-modèles, et dont les instances sont des modèles de la couche M1. Le méta-modèle du langage UML en est un exemple (Remarquons qu'une classe X d'un modèle de niveau M1 écrit en UML n'est autre qu'une instance du type Class du modèle UML).
- ❖ M3 c'est un niveau d'abstraction encore plus élevé, où l'instance d'un modèle de cette couche donne un modèle de la couche M2. Un modèle de M3 donnera une syntaxe d'écriture de méta-modèles. méta -méta-modèle est un langage pour décrire des langages ; et ce décrit lui-même !

La figure suivante illustre les 4 niveaux de modélisation dans un contexte de bases de données relationnelles.

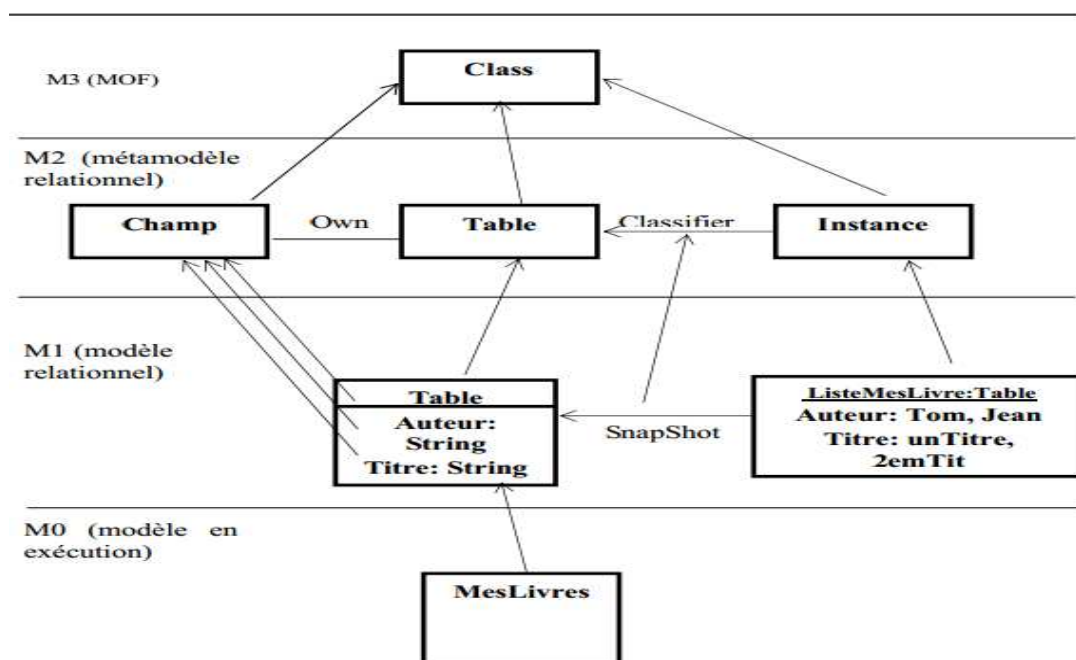


Figure II. 3: 4niveaux de modélisation dans un contexte de bases de données relationnelles

II.3 La méta-modèle

II.3.1 Définition

La notion de méta-modèle est utilisée dans trois domaines de l'informatique : en intelligence artificielle, en programmation objet et en génie logiciel [1]. En intelligence artificielle, et plus précisément en représentation des connaissances, un méta-modèle décrit la

structure des modèles et permet de raisonner sur les modèles comme sur les connaissances de premier niveau. Certains langages à objets, comme Java, décrivent de la même manière les classes, les méta-classes et les objets. En génie logiciel, c'est avec UML que la notion de méta-modèle a été développée. Le méta-modèle décrit les modèles : classes, objets, attributs, relations... et se décrit lui-même. Dans l'ingénierie des méthodes, c'est un outil conceptuel indispensable pour définir et raisonner sur de nouveaux modèles. Dans l'ingénierie des outils, les méta-modèles sont une définition formelle nécessaire pour concevoir et construire les outils pour éditer vérifier, transformer et éventuellement exécuter des instances de ces modèles.

Autrement dit un méta-modèle est une définition formelle d'un modèle qui aide à le comprendre et qui facilite le raisonnement sur sa structure, sa sémantique et son usage. Méta-modèle signifie littéralement modèle du modèle. Il peut être défini comme la représentation d'un point de vue particulier sur des modèles. Un méta-modèle qui se décrit lui-même est dit réflexif. Cette propriété permet d'avoir une représentation complète des modèles sans régression infinie : modèle, méta-modèle, méta-méta-modèle...

Les méthodes de conception d'applications et d'architecture informatiques (telles que le cycle en V), leurs représentations formelles et les logiciels associés relèvent également de la méta-modélisation. Le terme méta-modèle est un stéréotype¹ attaché à un modèle qui indique que le modèle est une abstraction d'un autre modèle (un modèle d'un modèle). L'UML lui-même sert de méta-modèle pour les modèles qui représentent des systèmes réels construits.

II.3.2 Ingénierie Dirigée par les Modèles

II.3.2.1 Définition

L'IDM (Ingénierie Dirigée par les Modèles) se réfère à l'utilisation systématique des modèles comme des éléments centraux tout au long du cycle de vie du logiciel. L'objectif derrière l'utilisation d'un tel paradigme est d'augmenter la productivité et réduire le temps de

¹ Stéréotype: est un mécanisme d'extensibilité du méta-modèle d'UML qui permet de créer de nouveaux types d'éléments du modèle qui dérivent des existants. Ces nouveaux éléments ont leurs propres propriétés spéciales (exprimées comme des valeurs étiquetées), sémantique et notation. Un stéréotype s'applique à un ou plusieurs types d'éléments modélisés. Un stéréotype peut aussi avoir une icône visuelle associée, comme celui pour un objet frontière. Un élément modélisé peut avoir au maximum un stéréotype. Cependant, les stéréotypes peuvent avoir plusieurs enfants.

développement des systèmes complexes au moyen de modèles qui sont beaucoup moins liés à la technologie et qui sont beaucoup plus proches du domaine. Cette abstraction des problèmes complexes rend les systèmes plus faciles à spécifier et à maintenir.

Dans le domaine de l'IDM, la méta-modélisation joue un rôle très important. En effet, elle est considérée comme une technique courante pour définir la syntaxe abstraite des modèles et des interrelations entre les éléments du modèle. Si le modèle est une abstraction des éléments du monde réel, le méta-modèle représente encore une autre abstraction, définissant les propriétés du modèle lui-même. Un modèle est dit conforme à son méta-modèle.

II.3.2.2 Concepts de l'architecture MDA

Les concepts de l'architecture MDA (Model-Driven Architecture) sont issus, d'un côté, de certains concepts utilisés dans le domaine de développement d'une manière générale. D'un autre côté, d'autres concepts de MDA sont originaires des différents aspects de la modélisation et la transformation de modèles. Les concepts les plus utilisés dans MDA seront discutés dans les points suivants.

- ➔ **Système** : Le contexte de MDA est le système logiciel, soit préexistant ou en construction.
- ➔ **Modèle** : Un modèle est une spécification formelle de la fonction, de la structure et du comportement d'un système dans un contexte donné, et d'un point de vue spécifique (ou d'un point de référence). Un modèle est souvent représenté par une combinaison de schémas et de textes, typiquement en utilisant une notation telle qu'UML, enrichi avec des expressions de langage naturel. Une spécification est dite formelle quand elle est basée sur un langage qui a une signification sémantique bien définie liée à chacune de ses constructions, pour le distinguer d'un simple diagramme montrant des boîtes et des lignes.
C'est ce formalisme, qui permet au modèle d'être exprimé en format tel que XML, conformément à un schéma bien défini (XMI).
- ➔ **L'Orienté Modèle (Model Driven)** : Ce concept décrit une approche de développement de logiciels par lequel des modèles sont utilisés comme source primaire pour documenter, analyser, concevoir, construire, déployer et maintenir un système.
- ➔ **Architecture** : L'architecture d'un système est des spécifications des parties et des connecteurs du système et des règles pour les interactions des parties en utilisant

les connecteurs. Dans le contexte de MDA ces parties, ces connecteurs et ces règles sont exprimés par l'intermédiaire d'un ensemble de modèles en corrélation.

➔ Point de vue (Viewpoint) : Un point de vue est une technique d'abstraction pour se concentrer sur un ensemble particulier de soucis dans un système tout en supprimant tout le détail non pertinent. Un point de vue peut être représenté par l'intermédiaire d'un ou plusieurs modèles.

➔ Point de vue MDA (MDA viewpoints) : MDA spécifie trois points de vue par défaut sur un système : indépendant de l'informatique, indépendant des plates-formes et spécifique à une plate-forme.

Le point de vue indépendant de l'informatique (computation independent viewpoint) se concentre sur le contexte et les conditions du système sans considération pour sa structure ou traitement.

Le point de vue indépendant de plate-forme (platform independent viewpoint) se concentre sur les possibilités opérationnelles d'un système en dehors du contexte d'une plate-forme spécifique (ou de l'ensemble de plates-formes), en montrant seulement les parties de ces spécifications complètes qui peuvent être soustraites hors de cette plateforme.

Un point de vue spécifique à une plate-forme (platform specific viewpoint) renforce un point de vue indépendant de plate-forme avec des détails concernant l'utilisation d'une plate-forme spécifique.

➔ Plate-forme : Une plate-forme est un ensemble de sous-systèmes et des technologies qui fournissent un ensemble logique de fonctionnalités par des interfaces et des modèles d'utilisation. Les clients d'une plate-forme l'utilisent sans souci pour ses détails d'implémentation. Les exemples des plates-formes incluent les systèmes d'exploitation, les langages de programmation, les bases de données, les interfaces utilisateurs, les solutions middleware etc.

➔ Indépendance de plate-forme (Platform Independence) : L'indépendance d'une plate-forme est une qualité qu'un modèle peut montrer quand il est exprimé indépendamment des dispositifs d'une autre plate-forme. L'indépendance est un indicateur relatif en termes de mesurer le degré d'abstraction, qui sépare une plate-forme des autres.

➔ Modèle de plate-forme : Un modèle de plate-forme décrit un ensemble de concepts techniques représentant ses éléments constitutifs et les services qu'elle fournit. Il

spécifie également des contraintes sur l'utilisation de ces éléments et de ces services par d'autres parties du système.

- ➔ Transformation de modèle : La transformation de modèle est le processus de conversion d'un modèle à un autre dans le même système. La transformation combine le modèle indépendant de plate-forme avec des informations supplémentaires pour produire un modèle spécifique à la plate-forme.
- ➔ Implémentation : Une implémentation est des spécifications qui fournissent toutes les informations exigées pour construire un système et pour le mettre en service. Elle doit fournir toutes les informations requises pour créer un objet, et pour permettre à l'objet de participer à fournir un ensemble approprié de services en tant qu'élément du système.

II.3.2.3 MDA (Model Driven Architecture)

MDA (Model-Driven Architecture) est un standard, lancé par l'OMG qui se base sur l'IDM, fournissant un ensemble de lignes directrices ainsi qu'une architecture pour la conception des systèmes logiciels. L'approche MDA donne la possibilité de comprendre les systèmes complexes et le monde réel à travers une abstraction de ceux-ci. Ce point de vue abstrait du système est élaboré dans un cadre conceptuel ainsi qu'un nombre des standards fournis par l'OMG permettant de définir les Modèles, leurs relations ainsi que leurs transformations

Le principe de base du MDA est l'élaboration de différents modèles, en partant d'un modèle métier indépendant de l'informatisation (*Computation Independent Model, CIM*), la transformation de celui-ci en modèle indépendant de la plate-forme (*Platform Independent Model, PIM*) et enfin la transformation de ce dernier en modèle spécifique à la plate-forme cible (*Platform Specific Model, PSM*) pour l'implémentation concrète du système. Les PSMs peuvent utiliser des langages spécifiques à un domaine ou des langages généralistes comme Java, C#, Python, etc. Les techniques employées dans le cadre de l'approche MDA sont donc principalement des techniques de modélisation et des techniques de transformation de modèles.

Pour Représenter visuellement l'approche MDA, l'OMG a mis en place un framework, structuré de plusieurs types de Modèles. La Figure représente le cycle de développement en Y, qui met en œuvre ces modèles ainsi que leurs relations:

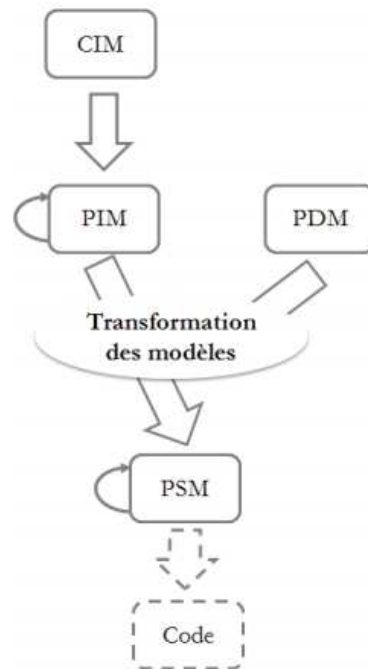


Figure II. 4: le cycle de développement en Y

⊕ CIM (Computation Independent Model):

Ces modèles décrivent le système à Concevoir d'un point de vue indépendant de l'informatisation. Le CIM permet une vision du système et de son environnement, tout en cachant les détails de structure et d'implémentation. Les Modèles du niveau CIM permettent de réduire l'écart entre les experts du domaine et entre les concepteurs. De ce fait, un modèle du CIM est parfois appelé un modèle de domaine.

⊕ PIM (Platform Independent Model) :

Les modèles du niveau PIM représentent une vision d'analyse et de conception du système, indépendamment de tout détail technologique concernant la plateforme (système d'exploitation, langage de programmation, matériel, performances du réseau, etc).

⊕ PSM (Platform Specific Model) :

Le niveau PSM présente une projection des Modèles du niveau PIM vers une plateforme spécifique. Ces Modèles combinent les Spécifications du PIM avec les détails propres à la plateforme.

⊕ PDM (Platform Description Model) :

Ces modèles décrivent la plateforme sur laquelle le système va être exécuté, en fournissant un ensemble de données techniques concernant les fonctionnalités et l'utilisation de la plateforme.

II.3.2.4 IDM vs MDA (Model Driven Architecture)

Ce que propose l'approche de l'ingénierie des modèles (IDM, ou MDE en anglais pour Model Driven Engineering) est simplement de mécaniser le processus que les ingénieurs expérimentés suivent à la main. L'intérêt pour l'IDM a été fortement amplifié à la fin du XX^e siècle lorsque l'organisme de standardisation OMG (Object Management Group) a rendu publique son initiative MDA (Model Driven Architecture), qui peut être vue comme une restriction de l'IDM à la gestion de l'aspect particulier de dépendance d'un logiciel à une plateforme d'exécution. De manière plus générale, et comme de nombreuses autres approches de génie logiciel telles que la programmation générative, la programmation par aspects, les usines à logiciel ou encore le MIC (Model Integrated Computing), l'IDM est une forme d'ingénierie générative, qui se singularise par une démarche par laquelle tout ou partie d'une application informatique est générée à partir de modèles.

Pour cela il faut bien sûr que les modèles comme les processus de tissage de la conception soient rendus explicites, et soient suffisamment précis pour être interprétés ou transformés par des machines. Le processus de conception peut alors être vu comme un ensemble de transformations de modèles partiellement ordonné, chaque transformation prenant des modèles en entrée et produisant des modèles en sortie, jusqu'à obtention d'artéfacts exécutables. Ainsi, quand on doit dériver un nouveau produit, qu'il soit simple évolution d'un produit existant ou nouvelle variante, on peut se contenter de « rejouer » automatiquement la plus grande partie du processus de conception, en changeant simplement quelques détails ici et là.

II.3.2.5 Standards d'OMG pour MDA

Pour obtenir une telle efficacité, plusieurs outils conceptuels sont mis à la disposition des équipes de développement. Parmi ces outils, on peut situer : UML, XMI, MOF et CWM. Ces outils sont proposés par OMG:

UML : Largement utilisé par ailleurs, qui permet une mise en oeuvre aisée de MDA en offrant un support connu comme étant un langage de modélisation qui fournit plusieurs constructions permettant de spécifier la structure et le comportement d'un système.

XMI : XML Metadata Interchange, propose un formalisme de structuration des documents XML de telle sorte qu'ils permettent de représenter des méta données d'application de manière compatible.

MOF : Meta Object Facility, spécification qui permet le stockage, l'accès, la manipulation et la modification de méta données.

CWM : Common Warehouse Metamodel, pour définir une base de données pour des métadonnées. L'OMG n'a pas jugé utile de standardiser un processus associé à ces outils. Leur rôle est de répondre aux besoins des utilisateurs de manière générique, et non de proposer de solutions définitives pour certains types d'applications précises. Un processus de génie logiciel exploitant les possibilités de MDA a cependant été proposé : le 'Model-Driven Software Development'

Conclusion :

En informatique le besoin de la modélisation est devenue de plus en plus un point important dans le développement des applications et au courant de ce chapitre on a présenté des notions concernant la méta-modélisation, dont on a défini le modèle et la méta-modèle, on a présenté IDM qui définit par l'OMG ; ou IDM est une approche utilisée dans contextes des logiciels préexistants, ou en cours de construction ; puisque notre travail consiste à réaliser une solution basée sur la méta-modèle pour l'intégration de système . Pour le chapitre qui suit on va entamer la conception où on va présenter notre de travail et le résultat finale qu'on veut atteindre, et pour la conception on va utiliser le langage de modélisation UML.



Chapitre III
Analyse et Conception

Cas d'études

E_Learning

III.1 Introduction

Les problèmes liés aux développements et à l'usage des composant ne sont pas encore complètement résolus, malgré une réelle précédée. Le principal but est que tout produit qui proviens d'un processus de développement, qu'il soit conceptuel (spécification) ou plus opérationnel (langage abstrait, code), est évidemment faiblement utilisable. Pour remettre un produit de développement réutilisable, ça nécessite d'avoir des langages conceptuels et opérationnels permettant l'expression de solution génériques, adaptables, intégrables, composables, etc.

L'intégration des applications et des processus métiers dans les entreprises est une question critique du point de vue de l'augmentation de l'efficacité et de la réduction des coûts. Les objectifs sont alors entre autres d'améliorer la réutilisation des services, des temps de développement réduits et une meilleure maîtrise des coûts de maintenance.

Ce chapitre est organisé en deux parties, nous allons d'abord parler de l'enseignement programmé et bien définir les sigles EAO et EIAO, ensuite voir les différents systèmes d'enseignement assistés par ordinateur, ainsi que les concepts de la formation à distance (FAD) en suite nous présentons les solutions et l'architecture matérielle d'une plate-forme de FAD. La dernière sous partie porte sur l'apprentissage collectif et ses pratiques en Formation à Distance. Puis nous allons passer à l'analyse et conception pour une solution d'integration en utilisant un méta-modèle

III.2 Les différents systèmes d'enseignement assisté par ordinateur

L'application des environnements informatiques dans le domaine de l'Éducation n'est pas une nouveauté. Du point de vue historique, il apparaît clairement que les concepts et les modèles pédagogiques de chaque époque ont des influences directes sur l'usage de la technologie informatique.

III.2.1 Enseignement programmé

Depuis l'avènement de l'ordinateur il y a plus d'une décennie, chaque innovation technique semble apporter avec elle ses problématiques, ses questions, ses recherches et ses réponses.

Les années soixante ont vu apparaître les premiers systèmes d'enseignement informatique. Cela a poussé certains à rêver de voir des professeurs assistés dans leur noble tâche par des programmes pédagogiques. Cet enseignement consistait à présenter à

l'apprenant une notion particulière, ensuite à l'interroger en lui posant différentes questions. Ces questions sont soit des QCM (Questions à Choix Multiples), soit dans l'attente d'un résultat déterminé. On nommait ces systèmes, les systèmes à règle de production ou système expert « si....Alors...sinon ». Pourquoi cette nomination, pour la simple raison que le système expert était basé sur le principe : « si l'apprenant répondait bien, le système faisait ceci, si l'apprenant répondait mal le système faisait cela ».

Ces systèmes eurent pour principal intérêt de supprimer le délai entre le travail effectué par l'apprenant et sa correction. Ainsi, ils eurent la possibilité de s'adresser à un grand nombre d'apprenants.

En parallèle, ces systèmes se révélèrent très contraignantes concernant la complexité et la durée de rechargement d'une part, et la limitation de leurs mémoires d'autre part.

Ce type d'enseignement peut être adapté pour un enseignement individuel des connaissances simples mais inadéquat pour apprendre à résoudre de véritables problèmes du domaine.

III.2.2 Enseignement Assisté par Ordinateur (EAO)

L'enseignement Assisté par Ordinateur (EAO) peut-être défini comme étant un ensemble des techniques et des méthodes d'utilisation de systèmes informatiques comme outils pédagogiques intégrés au contexte éducatif.

Les premières tentatives de l'Enseignement Assisté par Ordinateur (EAO) s'appuyaient sur le modèle théorique du béhaviorisme et sur les principes de l'enseignement programmé « *Le béhaviorisme peut être décrit comme une approche qui se concentre sur l'étude du comportement observable et du rôle de l'environnement en tant que déterminant du comportement. Le terme « béhaviorisme » vient de l'anglais **behaviour** (orthographe américain : **behavior**) qui signifie « comportement ».*

Dans ce type d'enseignement, les actions pédagogiques de l'apprenant relèvent d'une planification statique : la programmation du cours est faite par un acteur humain et l'ordinateur se contente d'exécuter le programme. Les connaissances sont compilées et un code est généré.

➤ Les logiciels de l'EAO (didacticiels)

Un didacticiel est un logiciel à but éducatif, conçu pour l'enseignement assisté par ordinateur, spécialisé dans une ou plusieurs matières mettant un apprenant en interaction avec la matière enseignée dans le but de lui faire acquérir des connaissances, on distingue :

- Les didacticiels de test ou questionnaires
- Les didacticiels informatifs
- Les didacticiels d'entraînements
- Les didacticiels de simulation
- Les didacticiels complets ou tutoriels
- Les didacticiels intelligences ou pédagogiques

Le test : logiciel à but purement évaluatif ou diagnostique. L'apprenant répond à un questionnaire séquentiel «ouvert » ou à choix multiple. Après chaque question, le didacticiel valide ou infirme la réponse donnée avant de passer à la question suivante. A la fin du test, l'apprenant recevra une évaluation chiffrée de sa performance globale et en tirera les conséquences qu'il jugera utile. Utilisés seuls, ils ne méritent pas l'appellation de didacticiel mais plutôt celle d'outil didactique diagnostique.

Le didacticiel informatif : il s'agit de mini-encyclopédies électroniques, apparentées aux banques de données. Ce type de logiciel permet de présenter des connaissances, sans chercher à vérifier leur acquisition par l'apprenant. Cette organisation est réduite à une présentation de pages écrans (textuelles et/ou graphiques, fixes ou animés), éventuellement agrémentées d'un zonage plus ou moins complexe.

Le didacticiel d'entraînement ou drill : c'est une combinaison simple des deux types précédents de logiciels. L'organisation d'un drill consiste essentiellement en une succession, plus ou moins longue, de modules-question.

Le didacticiel de simulation : c'est un logiciel réalisant une modélisation d'un processus quelconque, paramétrables par l'apprenant. La structure d'une simulation comprend : un module d'exposition du processus exploré, un module d'exposition du modèle utilisé, un module de paramétrage et un module de représentation.

Les deux premiers modules sont considérés comme des didacticiels informatifs, le troisième module consiste en une séquence dialoguée, qui permet à l'apprenant de donner des valeurs aux paramètres du module.

Le quatrième module est invoqué par l'apprenant pour obtenir une représentation du processus associé au jeu de paramètres choisis : Cette représentation peut être :

- ✓ Un texte, dans le cas d'une simulation grammaticale ou linguistique ;
- ✓ Un graphique, une animation dans le cas d'un phénomène physique, chimique... ;

- ✓ Un tableau ou un histogramme, et s'il s'agit d'économie, de sociologie, etc. ;
- ✓ Un message sonore, s'il s'agit de musique ou d'acoustique...

Ce module de représentation boucle sur celui de paramétrage. Le principe pédagogique de la simulation consiste à familiariser l'apprenant avec un modèle, en lui faisant « toucher du doigt » l'effet des diverses variables modifiables.

Le tutoriel : didacticiel complet, le tutoriel peut utiliser toutes les formes décrites ci-dessus, en les organisant dans une structure complexe. L'arborescence résultante a plusieurs fonctionnalités, il s'agit de prendre en compte :

- ✓ Les divers sous objectifs du but pédagogiques, qui ne doivent pas toujours être traités séquentiellement ;
- ✓ Pour chaque objectif, les divers stratégies utilisées : modules informatifs, tests locaux, drills, simulations ;
- ✓ Pour chaque question posée, l'analyse correspondante et l'aiguillage en fonction de la réponse de l'apprenant ;
- ✓ Les structures d'aide à mettre à disposition, à chaque étape du didacticiel ;
- ✓ La structure de suivi de l'apprenant, qui doit permettre un diagnostic final ou partiel ;
- ✓ Eventuellement une structure d'orientation, destinée à aiguiller l'apprenant selon son niveau initial, déclaré ou mesuré.

Le pédagogique: Toutes les formes générales décrites plus haut peuvent être utilisées. Ce néologisme désigne un didacticiel possédant en plus les caractéristiques suivantes : Les questions sont réellement « ouvertes ». L'analyse des réponses en langue naturelle est traitée « intelligemment ».

III.2.3 Enseignement Intelligent Assisté par Ordinateur (EIAO)

Dans les années 80, le module EAO a été remis en question et détrôné par le modèle cognitiviste. Ce modèle ne se satisfait pas de l'observation externe des comportements des apprenants, il cherche à comprendre comment ces comportements sont produits, quelles sont les activités mentales mises en jeu en situation d'apprentissage ? Les cognitivistes ont développé leur modèle en se basant sur la modélisation informatique et en considérant l'intelligence comme un système de traitement de l'information symbolique dans lequel la connaissance est une information qui est traitée, transformée et intégrée en mémoire à long

terme. Avec l'utilisation des techniques de l'intelligence artificielle dans les systèmes d'enseignement, on parle alors de l'enseignement intelligemment assisté par ordinateur EIAO.

La conception de ces systèmes suppose quatre axes de recherches :

- ✓ les contenus d'enseignement (**modèle expert**) : Une des principales caractéristiques d'un tutoriel intelligent, comme pour un enseignant, est d'être compétent dans la matière qu'il enseigne. En outre, cette compétence permet au tutoriel d'exposer son raisonnement et de justifier ses décisions. Cette composante est appelée modèle du domaine ou modèle de l'expert. Il a pour tâches : la génération de problèmes, la résolution de problèmes, le test des réponses de l'élève (diagnostic des cours) et l'explication.
- ✓ les difficultés de l'élève (**modèle de l'apprenant**) : Ce portrait porte le nom de modèle de l'élève. Il sert à construire un diagnostic qui pourra servir au système pour prendre une décision de nature didactique (questionnement, explications, ...). Le modèle élève représente les connaissances et les aptitudes de l'apprenant qui s'enrichit au fur et à mesure de l'apprentissage. Il contient les connaissances liées au domaine (ce que l'apprenant sait, ne sait pas, ce qu'il sait faire et ne sait pas faire), les connaissances indépendantes du domaine enseigné (les mécanismes d'apprentissage préférés par l'apprenant) ainsi que les connaissances servant à l'identification de l'apprenant.
- ✓ les méthodes d'enseignement (**modèle pédagogique**) : Un module pédagogique est donc un ensemble de spécifications sur la manière dont le système doit construire ses interventions c'est le noyau du système. Ce module interagit avec l'élève plus ou moins directement en sélectionnant les problèmes qu'il doit résoudre, en le guidant vers la solution, en critiquant ses performances, en lui fournissant une aide appropriée lorsque l'élève demande, en montrant des exemples.
- ✓ **L'interface utilisateur** (composante non négligeable dans le processus d'apprentissage) : La qualité de cette interface est fondamentale, Il est important qu'elle soit ergonomique, conviviale et facile à utiliser afin que l'apprenant se sente à l'aise dans son apprentissage. Cette interface permet à l'apprenant de communiquer avec le pédagogue et le système.

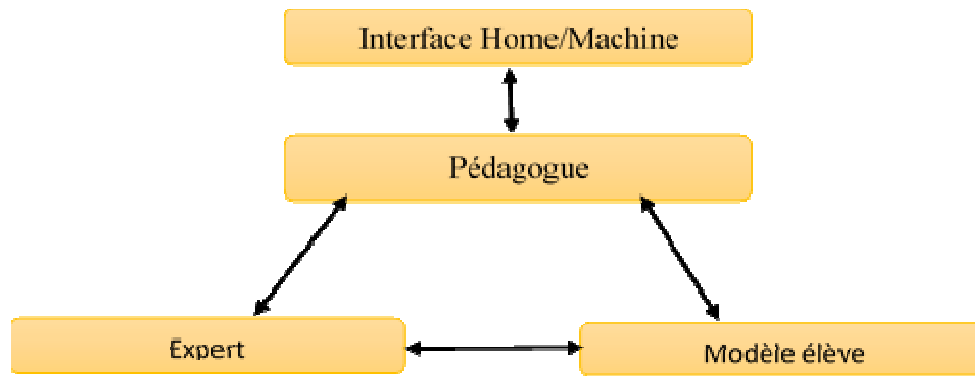


Figure III. 1: Les composants d'un système d'enseignement intelligent

III.2.4 Environnement Intelligent d'Apprentissage par ordinateur

L'**EIAO** pour initialement Enseignement Intelligemment Assisté par Ordinateur devient plus tard pour Environnement Interactif d'Apprentissage avec Ordinateur ou encore EGO (Enseignement Géré par Ordinateur) et EBO (Enseignement Basé sur Ordinateur). Ce changement d'appellation vise à mettre l'accent sur l'apprentissage plutôt que sur l'enseignement automatisé. Les capacités de résolution et d'explication de la machine s'étant révélées insuffisantes pour l'apprentissage, les recherches se centrent davantage sur les besoins de l'apprenant. L'interactivité et l'interaction deviennent le cœur des problématiques visées par ce nouvel axe de recherche mais les contraintes de l'informatique demeurent très pesantes et les résultats ne se révèlent pas toujours satisfaisants. Il était nécessaire de repenser l'outil informatique et de le rendre non-dirigiste pour favoriser l'apprentissage.

1. Les systèmes Tuteurs Intelligents

Les STI (Intelligent Tutoring Systems - ITS) sont nés dans les années 90 à partir d'une idée simple : on dispose d'un système expert pour la résolution de problèmes, d'une base de connaissances pour un domaine, on ajoute un module pour assurer le transfert de connaissances du système vers l'utilisateur et on obtient un système d'apprentissage individualisé qui va servir à former les étudiants en complément à un enseignement classique dispensé par exemple sous forme de cours magistraux. Dans un STI, l'apprenant peut être acteur ou observateur dans la résolution d'un problème (problème proposé par le système ou par l'apprenant). Il est acteur s'il résout lui-même le problème, le système assurant un guidage et un contrôle ; il est observateur et reçoit les explications que lui fournit le système si ce dernier résout le problème.

Généralement, l'architecture des STI comporte quatre composants indépendants : le module « représentation du domaine », le module « modèle de l'élève », le module « tuteur » et le module « interface »

2. Le micro-monde

Le concept de Micro-Monde se situe franchement sur le terrain de la simulation en s'appuyant sur des techniques hypermédias. Dans les simulations, les connaissances existent en tant que modélisation; les lois du domaine gouvernent les réactions du logiciel. Les objets manipulés dans un micro-monde ont des propriétés communes à la fois avec des objets formels du domaine et avec les objets concrets du monde réel. La tâche proposée à l'élève est de construire et de manipuler ces objets.

Donc les EIAO sont des systèmes d'enseignement dont l'objectif est de prendre ce qu'il y a de meilleur dans les STI, c'est-à-dire les capacités du système à pouvoir raisonner sur un domaine spécifique, et de prendre aussi ce qu'il y a de meilleur dans les micro-mondes, c'est-à-dire l'apprentissage par l'interaction.

III.2.5 Environnement Informatique pour l'Apprentissage Humain

Pour prendre en compte les différents aspects humains et communicationnels qui interviennent dans les processus d'apprentissage, l'appellation EIAO se transforme en 1998, en EIAH pour *Environnement Interactif pour l'Apprentissage Humain* (EIAH). Cette dénomination est un nouvel axe de recherche qui met en valeur la communication Homme-Homme à l'intérieur du domaine de communication Homme-Machine.

Aujourd'hui, le concept intégrateur d'EIAH est utilisé soulignant à la fois le côté informatique et le côté humain de l'apprentissage. Il est maintenant admis par tous les acteurs participant à la conception d'environnements d'apprentissage avec ordinateur qu'il est nécessaire d'intégrer l'enseignant humain (le tuteur) dans le dispositif de formation.

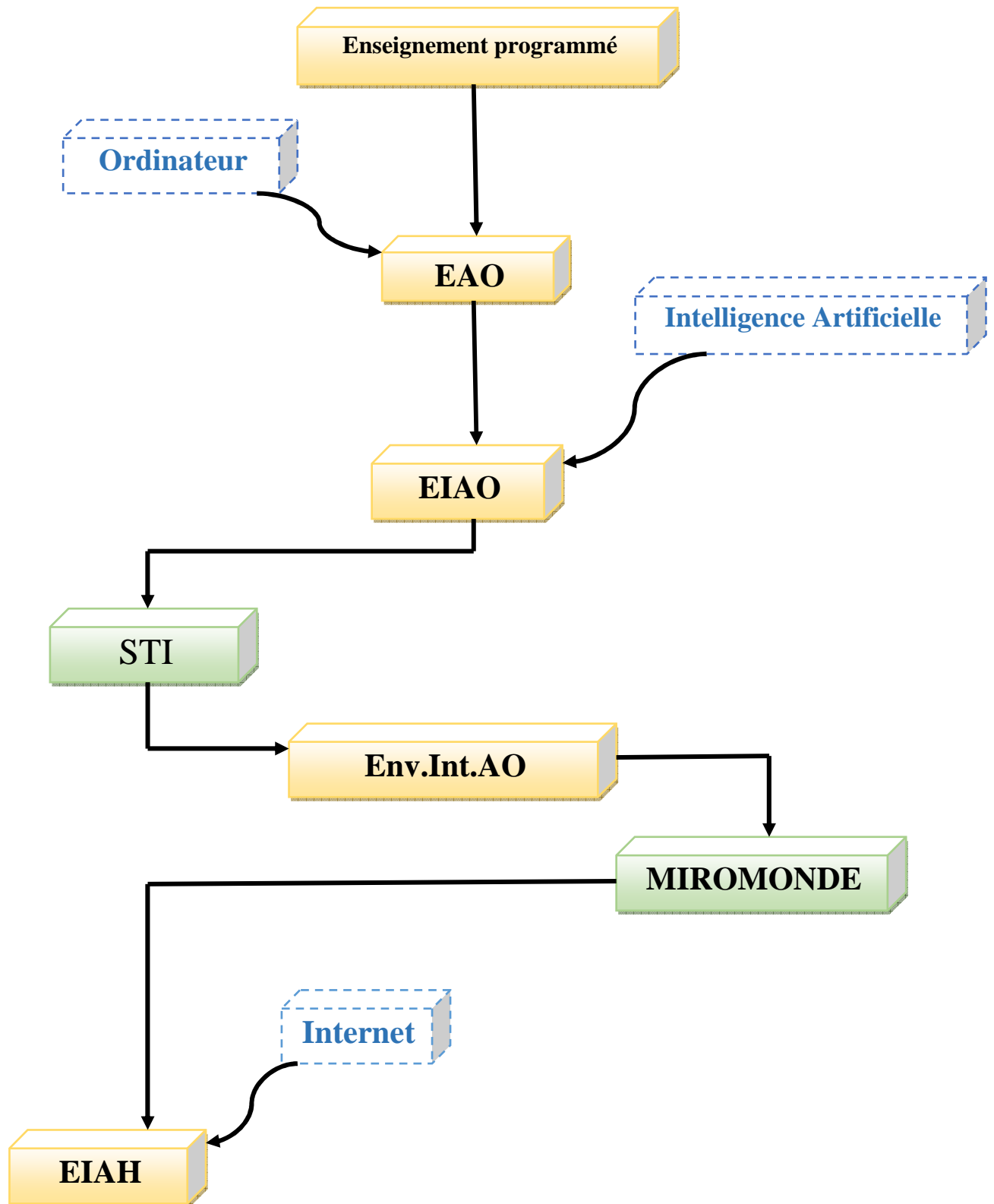


Figure III. 2: Schéma récapitulatif des systèmes d'enseigne

III.2.6 La formation et l'enseignement en ligne

III.2.6.1 Présentation

L'enseignement en ligne est un aspect relativement récent de l'e-learning il fait partie de la 3eme génération dans le processus d'évolution de ce nouveau mode de formation. En quoi consiste-t-il ?

On trouve, au premier échelon, l'enseignement qui met en ligne ses travaux dirigés et permet ainsi aux apprenants de se préparer à la séance qui va suivre. Ce support de cours en ligne encourage donc un aller-retour entre le temps de formation en présentiel et le temps d'apprentissage à distance. Au second échelon, l'informatique prend toute sa dimension révolutionnaire : avec Internet, chacun peut mettre en ligne ses pensées, ses œuvres, s'exprimer sans frontière, sans autorisation, sans frais d'édition, en dématérialisant son existence physique au profit de la circulation de son esprit ... Ainsi, au-delà de la nouvelle lecture offerte par Internet il existe nécessairement une nouvelle écriture, libérée des contingences du monde de l'édition et des territoires gardés de l'élite intellectuelle. Cette massification de l'usage informatique de la pensée est en cours, d'ores et déjà.

III.2.6.2 Les différents aspects de la formation en ligne

La formation en ligne se présente sous une multitude de formes et emprunte une vaste panoplie de techniques. Cette particularité facilite son intégration aux stratégies de formation continue au sein des entreprises. Les produits de l'e-formation vont du simple transfert en ligne de présentation actuelles sous format Powerpoint par exemple, jusqu'aux systèmes développés sur mesure pour l'entreprise, en passant par un assortiment étendu de logiciels de formation standardisés se prêts à l'emploi. L'apprentissage à distance peut être également combine à diverses formes d'apprentissage plus traditionnelles afin de répondre de manière plus spécifique aux besoins de l'entreprise. En se familiarisant avec le thème, on voit plus clairement les méthodes ou les combinaisons les plus utiles.

- ✓ Formation exclusivement en ligne (sans tutorat)

Ce type de formation est proche du service fourni par un cédérom ou d'une vidéo de formation. On la trouve souvent sur les « portails de formation » aux Etats-Unis. Elle tend à devenir gratuite, certains portails faisant payer uniquement le tutorat en ligne.

- ✓ Formation exclusivement en ligne (avec tutorat)

C'est le modèle de référence des « portails de formation », plutôt destiné au grand public. L'acheteur paie en ligne sa formation. Un tuteur lui est attribué qui lui propose un programme de travail. Dans ce cas, la formation peut être asynchrone ou synchrone.

- ✓ Formation mixte (avec tutorat en ligne)

C'est plutôt le modèle des universités, écoles et organismes de formation. En mettant en ligne le contenu de la formation, des tests, des évaluations et un tutorat, cela permet de réduire le temps en présentiel et d'individualiser la formation. Le présentiel permet de faire le point, de répondre à des interrogations, voire d'approfondir le sujet traité.

Ce modèle de formation combine donc les éléments de l'apprentissage en ligne et de l'apprentissage traditionnel en classe.

III.2.6.3 Quelques exemples de formules d'apprentissage mixte

- ✓ Avant de bénéficier d'un enseignement traditionnel en classe, les apprenants reçoivent du matériel ayant recours aux technologies de l'information (Internet, cédéroms). Le temps de rencontre est diminué par cette préparation et la formation en classe se concerne sur les échanges approfondis.
- ✓ Entre deux séances de formation en classe, les participants poursuivent leur formation au moyen d'échanges en ligne structurés et d'outils de collaboration.
- ✓ Une classe se réunit en temps réel (mode synchrone : vidéoconférence, clavardage) et se poursuit en différé (forum de discussion, échange de courriels, didacticiels).
- ✓ Un support (coaching) est apporté aux apprenants, sur les lieux de travail, afin de faciliter l'intégration des connaissances acquises à travers une formation en ligne.

III.2.6.4 Les domaines de la formation en ligne

L'e-formation se développe dans les 3 domaines suivants :

- La formation « à la carte » : Ici, la cible est le particulier qui cherche une formation plutôt dans le domaine des loisirs ou de la culture. Cela passe par des portails de formation comme Smartplanet.com aux États-Unis.
- La formation initiale : La cible étant les écoles, les universités. Les organismes de formation mettent en œuvre des programmes de formation ou l'étudiant s'inscrit en

ligne à des cours, s'auto évalue, fait appel à un enseignement par e-mail, etc. l'aspect logiciel repose surtout sur des « plates-formes de téléformation » comme WebCt aux États-Unis.

- La formation continue professionnelle : les cibles sont les salariés d'entreprises, les indépendants, les professions libérales, les adultes en reprise d'études, etc. ce domaine est actuellement le plus concerné par l'offre de e-formation.
- Le rôle du tuteur dans la formation à distance : le tuteur dispose d'une base de données en ligne qui lui permet de connaître le nombre de connexions, leurs origines, les parcours pédagogiques des apprenants, le résultat des tests effectués, etc. on distingue deux types de tutorat : synchrone et asynchrone.
- Le tutorat synchrone : un tutorat synchrone peut s'exécuter par visioconférence, par réseau de messagerie, intégré ou non à la plate-forme de formation, ou encore par une « prise en main directe » sur le poste de l'apprenant.
- Le tutorat asynchrone : avant tout, il faut savoir qu'un apprenant ne maîtrise pas forcément les possibilités offertes par une plate-forme de formation en ligne. Il faut donc un tuteur pour l'aide à l'utilisation technologique, indépendamment du tuteur en charge de la formation en ligne. Au cours du processus de formation on identifie différents niveaux d'intervention du tuteur. Il répond (via la messagerie) aux questions des apprenants, les aides à surmonter les difficultés rencontrés. Mais il peut également solliciter l'apprenant, animer la formation en ligne, fournir des compléments spécifiques, orienter l'apprentissage vers un niveau supérieur, etc.

III.3 Notion de la FAD (Formation à Distance) ou EAD (Enseignement à Distance)

III.3.1 Origine de la FAD

En effet, les premiers cours à distance ont été proposés dès 1728, par Caleb Phillips. Il proposait des cours de correspondance privée à travers des petites annonces publiées dans la Boston Gazette (journal 1719-1798, Boston).

En 2840, Sir Isaac Pitman invente une méthode de sténographie. Pour la populariser, il propose des cours par correspondance. Il est bientôt suivi sur le continent par deux associés de Berlin, Charles Toussaint et Gustav Langenscheidt qui lancent leur propre école de correspondance et popularisent également la sténographie. Leur entreprise survivra jusqu'à la seconde guerre mondiale.

En 1858, l'Université de Londres sera la première à dispenser des diplômes reconnaissant les formations dispensées par correspondance.

En 1873, Anna Ticknor fonde la Société d'Encouragement pour l'Etude à domicile : une école également par correspondance et qui vise spécialement un public féminin.

III.3.2 Evolution vers les plates formes informatiques pour la formation à distance

L'évolution technologique s'accélère : Internet, Intranet, réseaux à hauts débits composent les nouvelles technologies de l'information et de la communication. De tels outils apportent une plus-value organisationnelle et pédagogique à la formation à distance en permettant un accès à des contenus distants et en permettant également les échanges entre les différents acteurs (apprenants/professeurs, apprenants entre eux, professeurs entre eux). Nous appelons Plate-forme Informatique pour la Formation À Distance (PIFAD) les dispositifs de formation à distance utilisant les réseaux informatiques comme support. Ils ont pour finalité la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le tutorat à distance. Ils intègrent des outils pour les différents acteurs de la FAD, pour faciliter les rôles et fonctions tenus par chacun de ces acteurs : professeur est chargé de concevoir le matériel pédagogique des cours, le concepteur informatique, ou parfois le professeur lui-même, réalise alors les médias (texte, image, vidéo, etc.). Le professeur crée par ailleurs des parcours pédagogiques. Le tuteur effectue un suivi du travail des apprenants ainsi qu'une assistance dans l'apprentissage de ces derniers. L'apprenant consulte en ligne ou télécharge les contenus pédagogiques qui lui sont recommandés, organise son travail, effectue des exercices, s'auto évalue et transmet des travaux au tuteur qui les évalue. L'administrateur installe et assure la maintenance du système, s'occupe de l'inscription administrative des étudiants, gère les accès et les droits aux ressources pédagogiques. On entend donc par administrateur un rôle spécifique à la plate-forme et non un rôle administratif habituel. La Figure 3 décrit ce modèle des PIFAD avec cinq acteurs.

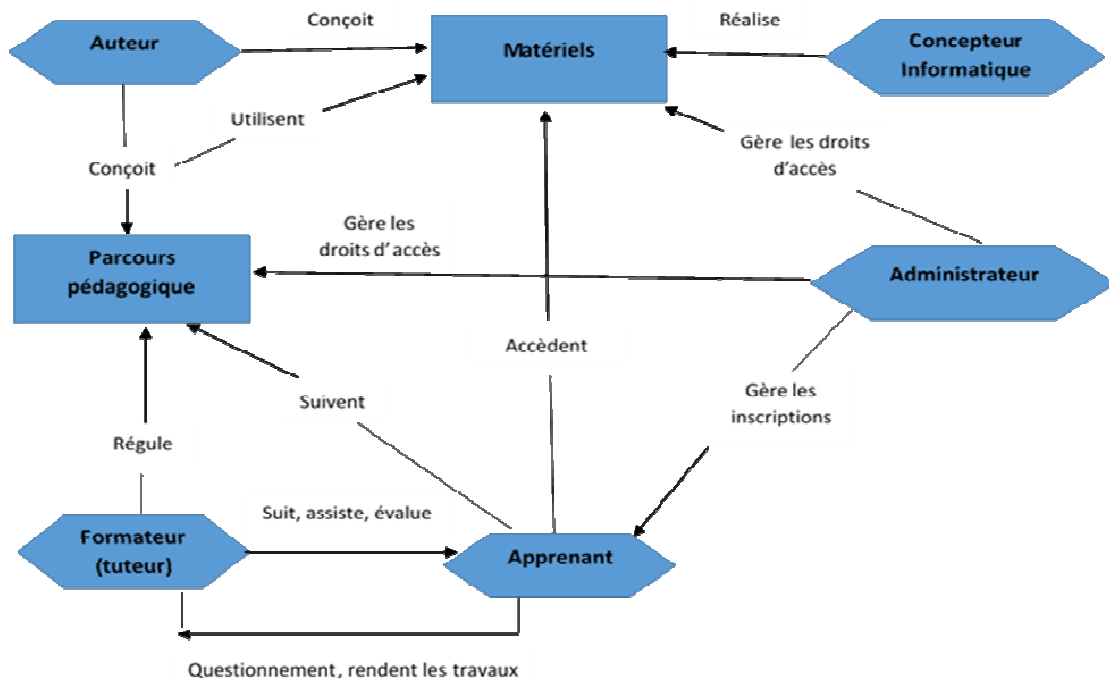


Figure III. 3: Schéma général des plates-formes pour la formation à distance

III.3.3 Modalités de la FAD

III.3.3.1 Formation à distance asynchrone

Est une méthode d'apprentissage s'adaptant aux disponibilités de l'apprenant. Il s'agit d'un apprentissage délivré à la demande. Dans ce mode, l'échange avec les acteurs (apprenant, enseignement) s'effectue via des modes de communication qui ne demandent pas une connexion simultanée des acteurs. Il peut s'agir des forums de discussion ou bien encore d'échanges de courrier électronique. Les principaux avantages de la formation asynchrone sont :

- Les apprenants évoluent à leur propre rythme et adaptent l'ordre dans lequel ils appréhendent les éléments du cours (personnalisation) ;
- Ils peuvent revoir et approfondir certains aspects du cours à leur guise ;
- Flexibilité de la formation (se libérer des contraintes de temps et de lieu)
- Accès illimité aux supports d'apprentissage
- La connaissance stockée dans le serveur web peut être maintenue et mise à jour d'une manière régulière et efficace.

III.3.3.2 Formation à distance synchrone

Il implique la connexion simultanée des participants à une session de formation. Il se rapporte à n'importe quel événement d'apprentissage délivré en temps réel à des apprenants distants et inclut la communication interactive et immédiate entre participants

Elle peut intégrer plusieurs modes de communication : visuel, audio, audiovisuel, textuelle... Les principaux avantages de la formation synchrone sont -Les apprenants interagissent intensivement à l'écran avec les formateurs, les langages oral et visuel sont utilisés (cas de l'audio vidéoconférence) ;

-la possibilité d'adapter l'apprentissage à l'apprenant en fournissant le feedback approprié ; - le modèle de la classe est familier ;

-la dynamique de groupes s'installe plus rapidement (motivation, coopération) ;

-le sentiment d'isolement est écarté.

III.3.4 Architecture matérielle d'une plate-forme de FAD

Les plates formes sont le plus part du temps constituées d'un logiciel serveur pouvant communiquer, par l'intermédiaire d'un serveur Web, avec un logiciel dit « Navigateur » comme le montre la figure :

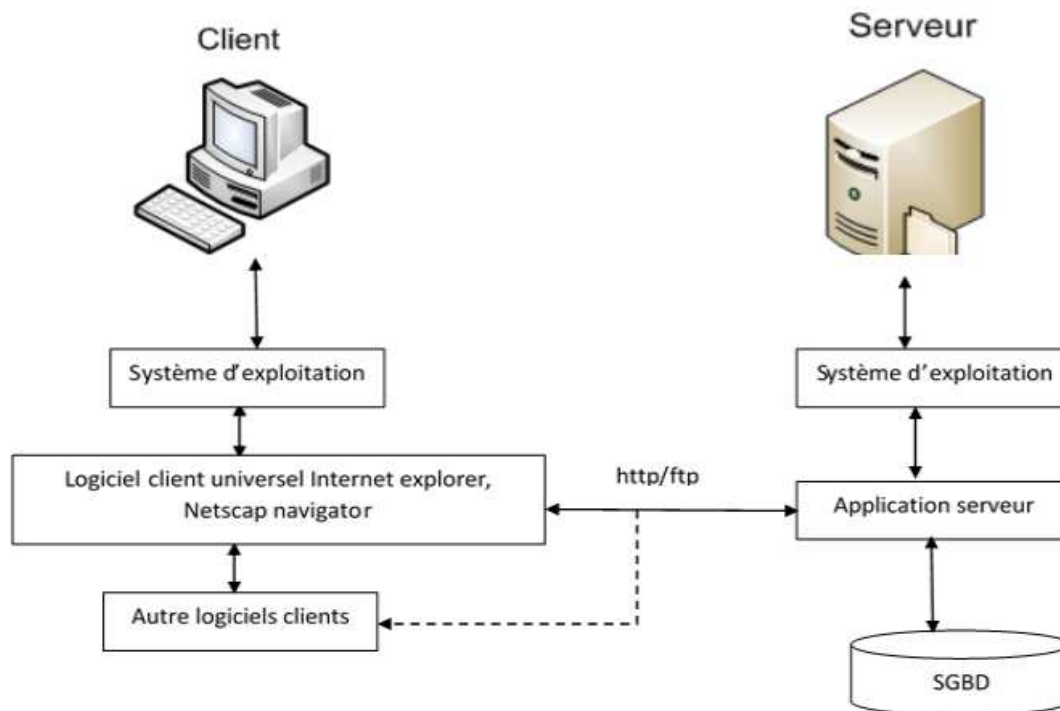


Figure III. 4: Architecture matérielle d'une plate-forme de FAD

III.4 Vers des Environnements Interactifs d'Apprentissage à Distance (EIAD)

L'évolution des technologies de la communication et les besoins en formation à distance renouvellent les problématiques de l'EIAO.

III.4.1 Définition des EIAD

L'EIAD se situe dans le prolongement des recherches en EIAO. L'idéification des EIAD générale des EIAD, comme des EIAO, est de considérer l'ordinateur comme pouvant favoriser la construction des connaissances chez l'utilisateur en créant des situations d'interactions. La différence majeure par rapport aux EIAO est que le système utilisé par l'apprenant est distribué, lui permettant ainsi d'accéder à des ressources distantes et de communiquer avec des personnes se trouvant à distance. Il est difficile de donner une définition précise pour les EIAD car les approches peuvent être différentes, à l'image de la variété d'approches que l'on trouve en EIAO.

III.4.2 Les situations d'interactions dans les EIAD

Dans un contexte d'EIAD on peut trouver principalement quatre principaux d'interactions : les interactions de l'apprenant avec le système distribué, les interactions de

l'enseignant avec le système distribué, et les interactions médiatisées apprenant-enseignant et apprenant-apprenants. Les interactions de l'enseignant avec le système distribué ne sont pas directement liées à l'apprenant mais font partie du contexte global en influençant la situation d'apprentissage. La Figure représente ces interactions.

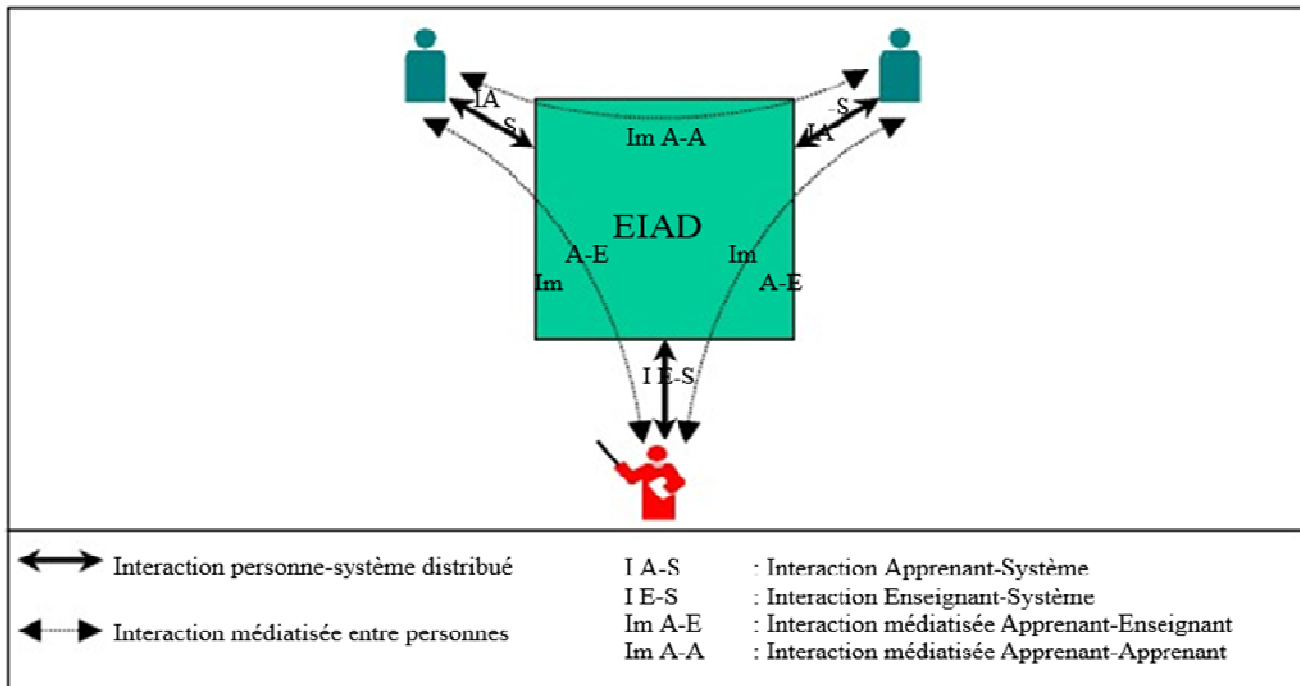


Figure III. 5: Les situations d'interactions dans les EIAD

III.4.3 L'apprentissage collectif

III.4.3.1 Coopération et collaboration

Les termes de coopération et de collaboration ne sont pas utilisés de façon univoque. Nous trouvons une distinction franche entre collaboration et coopération dans les domaines de recherche analysant les groupes de travail humain. La collaboration autant que la coopération implique un partage d'un objectif commun par plusieurs personnes mais la distinction se fait au niveau des sous-buts ou buts immédiats. Les personnes peuvent poursuivre un but général commun sans que les buts immédiats de leurs tâches soient identiques. Certains auteurs utilisent alors le terme de coopération ou de coopération distribuée.

III.4.3.2 Apprentissage coopératif et apprentissage collaboratif

Nous avons vu qu'il existait dans la littérature une certaine ambiguïté entre les termes « coopération » et « collaboration ». Il en est de même entre les expressions « apprentissage coopératif » et « apprentissage collaboratif ». Ainsi, Pléty souligne que l'orientation «

coopération » semble avoir été privilégiée dans le vocabulaire alors que, dans la réalité, l'apprentissage coopératif recouvre le plus souvent des pratiques de « collaboration » [Pléty 1998]. En effet, ces expressions véhiculent des connotations et des pratiques bien spécifiques. Nous ne pouvons pas dire que dans un cas nous sommes en présence d'apprenants qui collaborent et que dans l'autre cas les apprenants coopèrent. La distinction, quand elle est faite, se trouve ailleurs.

III.4.3.3 Apprentissage coopératif

L'origine de l'expression « apprentissage coopératif » est essentiellement américaine avec les écrits de Dewey mettant l'accent sur la nature sociale de l'apprentissage et avec le travail de Lewin sur la dynamique de groupe. Cette expression évoque un moyen pédagogique utilisé plus particulièrement avec des enfants dans des salles de classe. Roger et David Johnson caractérisent l'apprentissage coopératif par une interdépendance positive des objectifs des apprenants, avec des responsabilités individuelles [Johnson et Johnson 1988]. Le plus important pour eux est d'amener les apprenants à prendre conscience qu'ils « coulent ou nagent ensemble » (« sink or swim together »). Les autres points importants de l'apprentissage coopératif sont les interactions face-à-face et l'enseignement de compétences pour le travail en groupe.

III.4.3.4 Apprentissage collaboratif

L'expression « apprentissage collaboratif » semble avoir une origine anglaise, fondée sur le travail d'enseignants ayant exploré la façon dont les élèves pouvaient tenir un rôle plus actif dans leur propre apprentissage ainsi que des autres. Cette approche donne beaucoup de liberté à l'apprenant. Les activités ne sont pas très dirigées et les apprenants gèrent en grande partie leur travail de groupe. Par exemple, les rôles ne sont pas assignés par l'enseignant mais les apprenants négocient ces rôles entre eux. Dans ce qui suit on dresse un tableau qui synthétise les caractéristiques de l'apprentissage coopératif et de l'apprentissage collectif.

III.4.4 Pratiques avec les plates-formes informatiques pour la formation à distance

Un certain nombre de plates-formes informatiques pour la formation à distance existent actuellement. Ces plates-formes proposent des outils de communication entre apprenants. Les outils de forum de discussion et de courriers électroniques sont les plus répandus (communication asynchrone), actuellement ces plates-formes mettent davantage l'accent sur la diffusion des cours que sur la mise en place d'activités pédagogiques.

III.4.5 Apprentissage collectif soutenu par ordinateur : le domaine des CSCL

Dans cette partie, nous nous intéressons aux environnements connus sous la dénomination de CSCL (Computer-Supported Collaborative Learning). L'intérêt pour ces environnements est relativement récent. Cependant les recherches dans ce domaine sont de plus en plus nombreuses.

III.5 Notion d'E-Learning

III.5.1 Definition d'E-Learning

L'e-learning est un processus d'apprentissage à distance s'appuyant sur des ressources multimédias, qui permet à une ou plusieurs personnes de se former à partir de leur ordinateur. Les supports multimédias utilisés peuvent combiner du texte, des graphismes en 2 ou 3 dimensions, du son, de l'image, de l'animation et même de la vidéo.

Ces supports permettent de révolutionner l'approche pédagogique, d'employer des méthodes plus ludiques où l'interactivité joue un grand rôle, de diversifier les outils employés, de s'adapter davantage au processus d'apprentissage de l'apprenant, qui devient le pilote de sa formation. Ce dernier pourra se former à son rythme, en fonction de ses besoins et de ses disponibilités, ce qui est particulièrement important à une époque où la formation se décline tout au long d'une vie.

III.5.2 Evolution de l'e-learning

C'est au début du siècle, le développement du service postal comme le timbre-poste, la production et l'utilisation du papier ont été à l'origine des cours par correspondance en Angleterre. Aujourd'hui, grâce aux technologies de l'information et de la communication, sommairement appelés réseau, Internet et Web, les campus virtuels semblent s'imposer comme étant le dispositif technologique, pédagogique et communicationnel le plus utilisé.

Aujourd'hui, les vecteurs de communication connaissent une importante évolution, qu'il s'agisse des supports de stockage, des techniques et des outils de diffusion ou d'accès à l'information. La digitalisation de l'information et son stockage numérique, la télématique et les réseaux, enfin la restitution de l'information quelle qu'en soit la nature (texte, dessin, photo, cinéma, etc.), à partir d'un poste intégré, l'ordinateur multimédia, jouent le même rôle que jouaient respectivement la poste et le livre.

L'histoire de la formation à distance pourrait donc être comprise à partir de l'évolution des médias et des différents dispositifs qu'elle a utilisés. Chronologiquement tout a commencé par l'imprimé qui a marqué le début de la formation à distance et constitué la base des cours par correspondance. L'imprimé donc, est le principal élément d'enseignement et d'encadrement.

Dès les années 60, s'ouvre l'ère du multimédia caractérisée par un usage de différents médias (imprimé, radio, télévision, vidéo, CD, DVD, CD-I...) complémentaires et coordonnés en vue d'un objectif pédagogique nouveau.

Dans les années 80, avec la naissance de la micro-informatique puis de la télématique, commence l'époque contemporaine, celle d'Internet, des plates formes et des campus virtuels.

III.5.3 Différents aspects de l'e-learning

- L'usage des cédéroms éducatifs: On peut lire, écouter, regarder une œuvre dans l'ordre voulu par l'auteur. Naviguer, c'est choisir son parcours, sélectionner ses centres d'intérêts. Cela suppose une organisation transparente de l'information. C'est possible sur une bande magnétique mais pesant du fait des délais mécaniques. Sur CD, les temps d'accès sont de l'ordre de la milliseconde : les possibilités de navigation ne sont limitées que par l'imagination et le travail des auteurs. Les liens hypertextes sont les ingrédients par lesquels un cédérom, comme le web, est facilement navigable : un simple clic nous transporte sur une autre page, d'un mot à sa définition, etc.

C'est le cas par exemple des encyclopedies électroniques comme Encarta de Microsoft ou Encyclopédie Hachette de Hachette de Multimédia ; ces logiciels sont une mine inépuisables d'informations. L'encyclopédie Encarta par exemple peut être mise à jour via Internet ce qui garantit qu'elle soit toujours fiable et d'actualité.

- La visioconférence : Caractérise un rassemblement de personnes qui ne se trouvent pas dans le même lieu lors de la conférence mais qui peuvent voir et entendre les participants grâce à la vidéo. Elle assure le transfert bidirectionnel, en temps réel.
- L'audioconférence permet : la communication audio entre postes de travail équipés d'entrées/sorties audio. Pour cela, le son est numérisé, comprimé et transmis vers les destinations. L'envoi de son suppose une interface d'acquisition sonore et un micro. Pour éviter les problèmes d'écho, il est conseillé d'utiliser un casque, plutôt que le

haut-parleur de la machine pour entendre ses correspondants et leur parler en même temps.

- La visioconférence permet : la transmission d'images vidéo (orateurs, transparents,...) entre postes de travail équipés d'entrées/sorties vidéo. Certains transmettent également l'audio. La réception vidéo ne nécessite généralement aucun matériel spécifique puisque tout est fait en logiciel et affiché dans une fenêtre de l'interface graphique de la machine(X11, Windows,..). Cependant certains logiciels savent exploiter les possibilités des cartes vidéo performantes (la décompression JPEG ou MPEJ par exemple). L'émission vidéo suppose une carte d'acquisition vidéo (intégrée dans quelques rares stations de travail et Macintosh) et une caméra.
- Tableaux blancs partagés : ces outils permettent de travailler à plusieurs sur des données textuelles, graphiques et/ou scientifiques en reprenant le concept du tableau sur lequel tout le monde voit ce qui est affiché et sur lequel tout le monde peut écrire.
- Mbone : (Multicast Backbone) est un réseau virtuel développé dans le cadre du projet MICE (Multimedia Integrated Conferencing in Europ). MBone implémente de nouvelles solutions pour la transmission multicast, notamment en termes d'adressage et de routage. Sa topologie à l'échelle mondiale (20 pays) est une combinaison de maillages et arbres dont les feuilles sont situées sur les réseaux locaux des utilisateurs en service de diffusion.

MBone est constitué d'ordinateurs exécutant le programme « mrouterd ». Celui-ci implémente une version modifiée de DVMP (Distant Vector Multicast Protocol). L'acheminement de paquets IP ordinateurs destinés aux routeurs multicast « m-routeurs » voisins, qui répercutent par encapsulation entre « m-routeurs » est appelé « tunneling ».

- Annuaire de sessions : les annuaires de sessions permettent d'annoncer les services accessibles sur le MBone : nom, type (audio, vidéo, tableau blanc, ...), l'adresse de groupe, la durée de vie, la description, et de lancer automatiquement les applications permettant d'y accéder (à condition qu'elles aient été installées). Ils sont aussi utilisés pour créer et annoncer des conférences sur le MBone.

II.5.4 Les avantages de l'apprentissage

Accessibilité : L'e-learning peut faciliter l'accès à la connaissance. Un ordinateur équipé d'une connexion à Internet suffit. Il n'est pas nécessaire de se déplacer.

Flexibilité : Selon la formule choisie, la formation peut être suivie à n'importe quel moment, à n'importe quel rythme et depuis n'importe quel endroit.

Performances : De nombreuses études mettent en avant de meilleurs résultats de la formation en e-learning que lors d'une formation présentielle traditionnelle.

III.5.5 Les inconvénients de l'apprentissage en ligne

Autodiscipline : Les apprenants doivent faire preuve de rigueur et de discipline, particulièrement s'ils sont isolés dans une formation à distance.

Isolement : Les contacts avec les autres apprenants de la formation et avec le formateur sont réduits, voire inexistant.

Maîtrise des outils : L'e-learning nécessite une maîtrise suffisante des outils informatiques et d'Internet pour pouvoir suivre la formation

III.6 Conception

Présentation

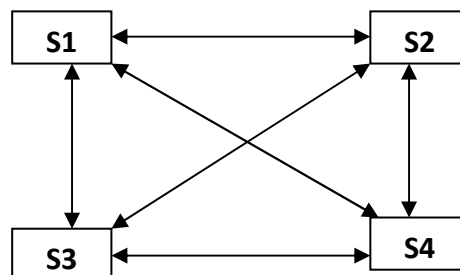
Comme nous avons montré dans le deuxième chapitre pour tout projet informatique la modélisation joue un rôle important dans sa réalisation; pour cela dans ce chapitre nous allons modéliser notre travail en utilisant UML comme langage de modélisation

Objectif du travail :

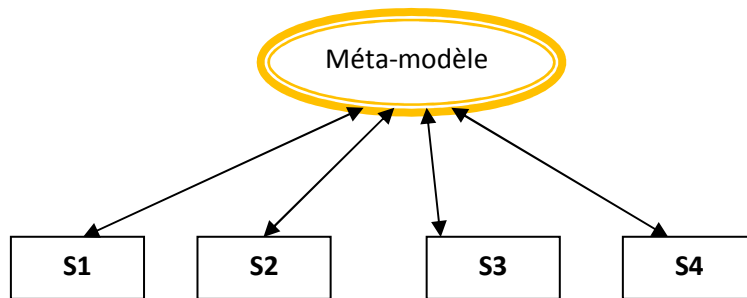
L'objectif est la réalisation d'un méta modèle pour l'intégration de système qui consiste à implémenter un moyen qui nous permettra de faire une fusion de plusieurs système nous intéressons à l'intégration de données. Supposant que nous devons faire une intégration pour n système, donc nous aurons besoin de réaliser $(n-1)!$ Application entre ces différents systèmes, ce qui va prendre un énorme de temps pour le faire, et une très grande complexité

Par exemple dans ce cas nous avons 4 systèmes, pour créer des liens nous aurons besoin de 6 liens $(4 - 1)! = 6$.

La solution basée sur des applications d'interfaçage :



La solution que nous allons proposer c'est de réaliser un méta-modèle de façon de minimiser le temps et la complexité.



III.6.1 Les diagrammes de classes représentant notre méta-modèle

Après avoir donné les concepts concernant le e-Learning nous allons amener a la conception de méta-modèle que nous voulons proposer pour la solution d'intégration de système. Pour ne pas compliquer la conception de notre méta-modèle nous allons le décomposer en petits scénarios qui correspondent aux parties importantes du système.

Comme nous avons montrer dans la première partie qui représente les applications de E_learning ; ces applications ont comme acteur principaux administrateur ; enseignant (formateur, auteur)

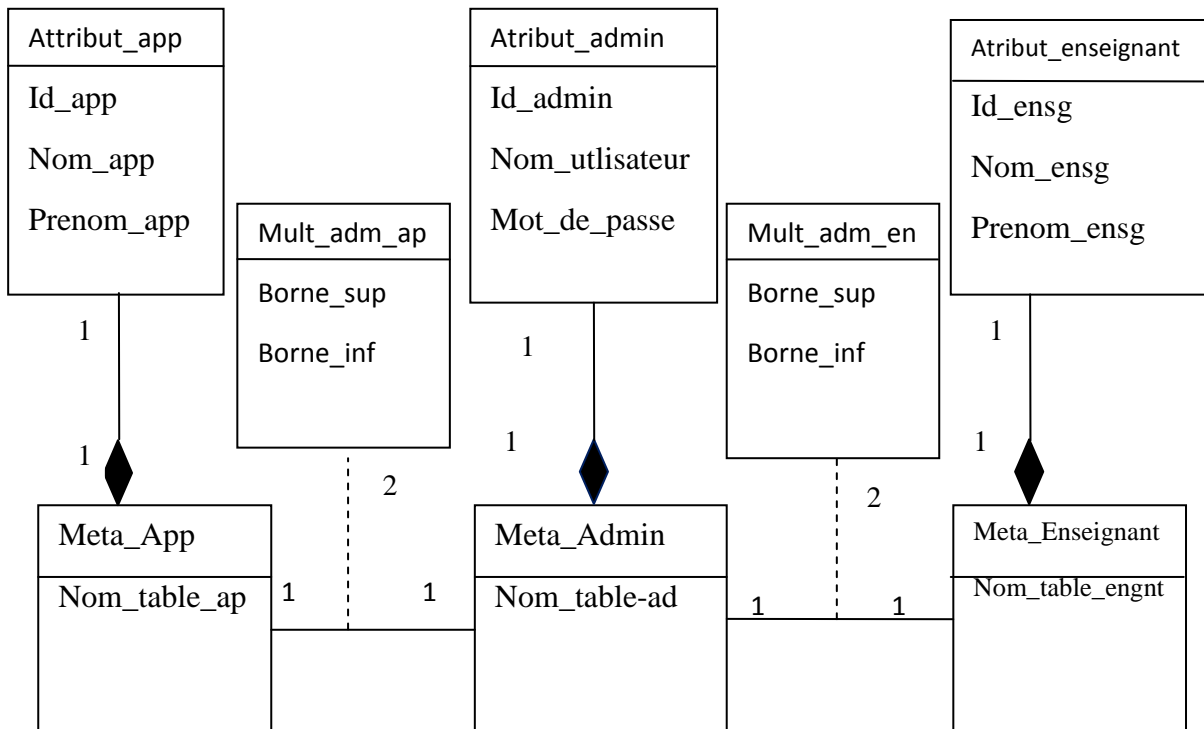


Figure III.6: Méta- modèle pour le rôle de l'administrateur

Les classes Meta_App, Meta_Admin, Meta_Ensgnt représente la structure des tables des modèle correspond à un système conforme à ce méta-modèle, dont chaque table elle a un nom, par ordre apprenant, administrateur, enseignant par exemple est composé de propriétés représenté (attribut), et entre les tables reliés nous aurons deux multiplicités.

Et le même principe que nous allons suivre pour les autres scénarios

III.6.1.2 Meta-modèle pour rôle d'enseignant

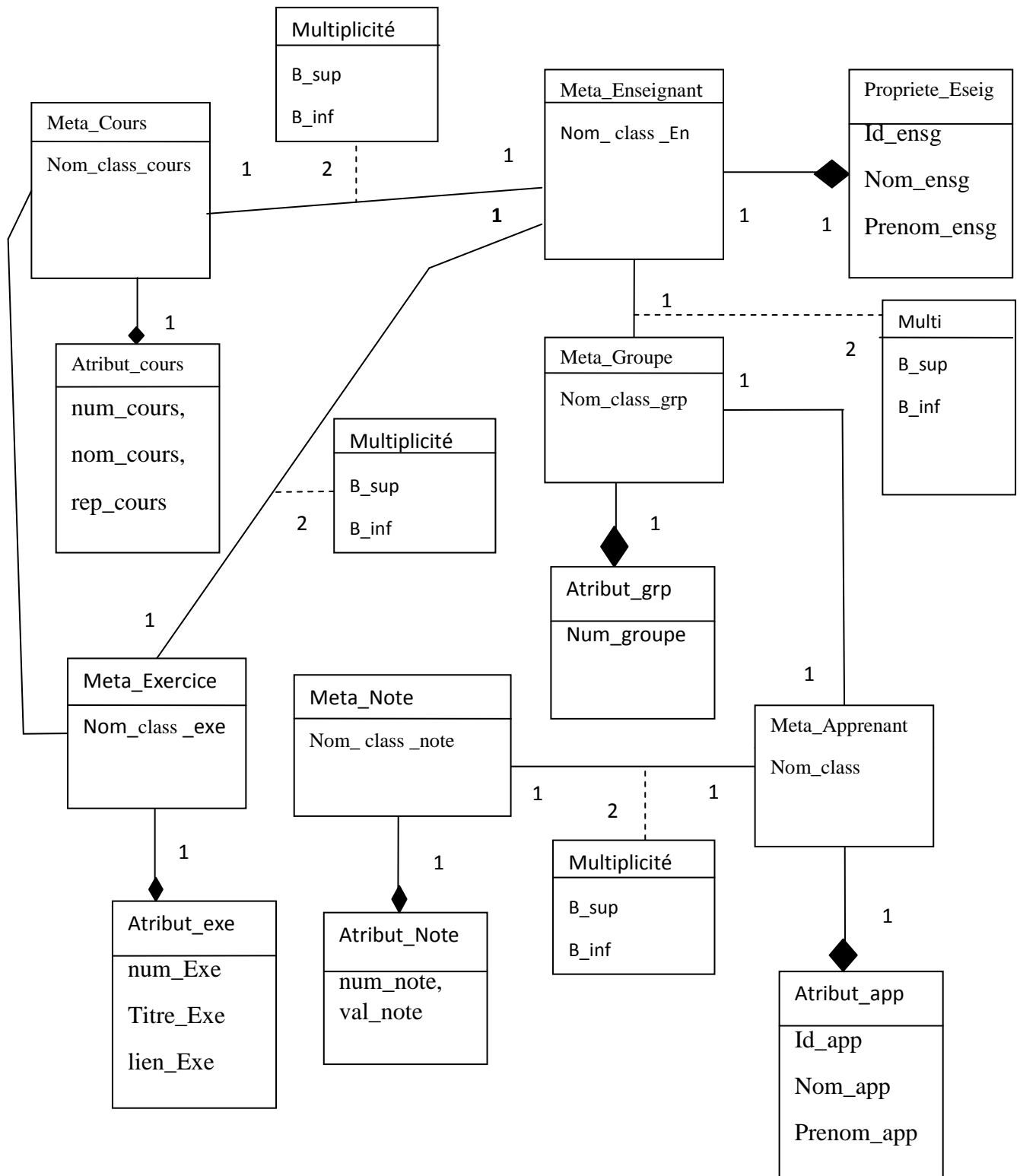


Figure III. 7: Meta-modèle pour rôle d'enseignant

III.6.1.3 Meta-modèle pour la gestion d'une matière

Pour la gestion de la matière nous trouvons les modules qui constitue de l'ensemble de cours et un cours peut avoir des exercices ou pas; pour chaque module aura un test

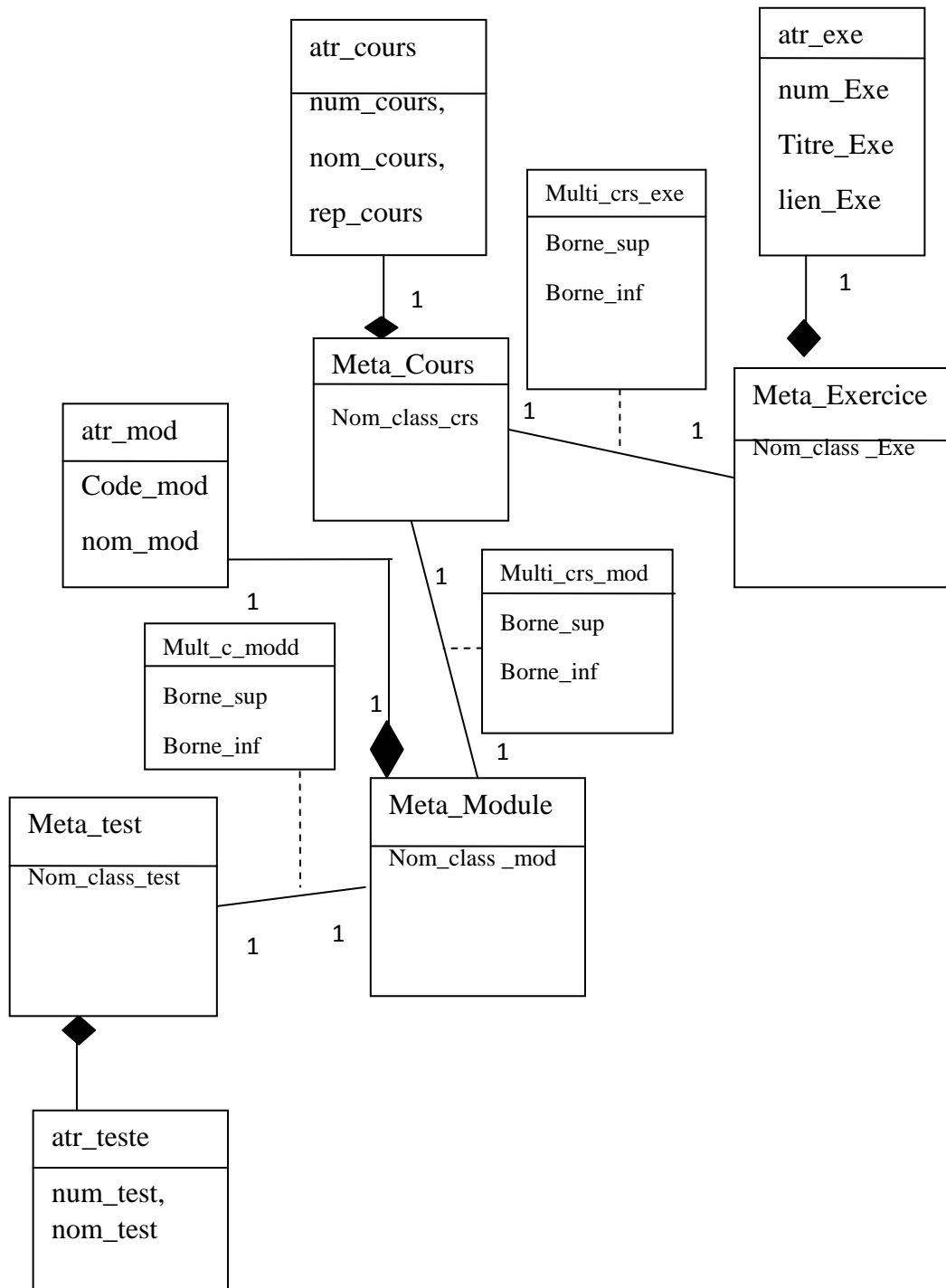


Figure III. 8: Meta-modèle pour la gestion d'une matière

III.6.1.4 Méta-modèle pour gestion de l'inscription d'un apprenant à un module

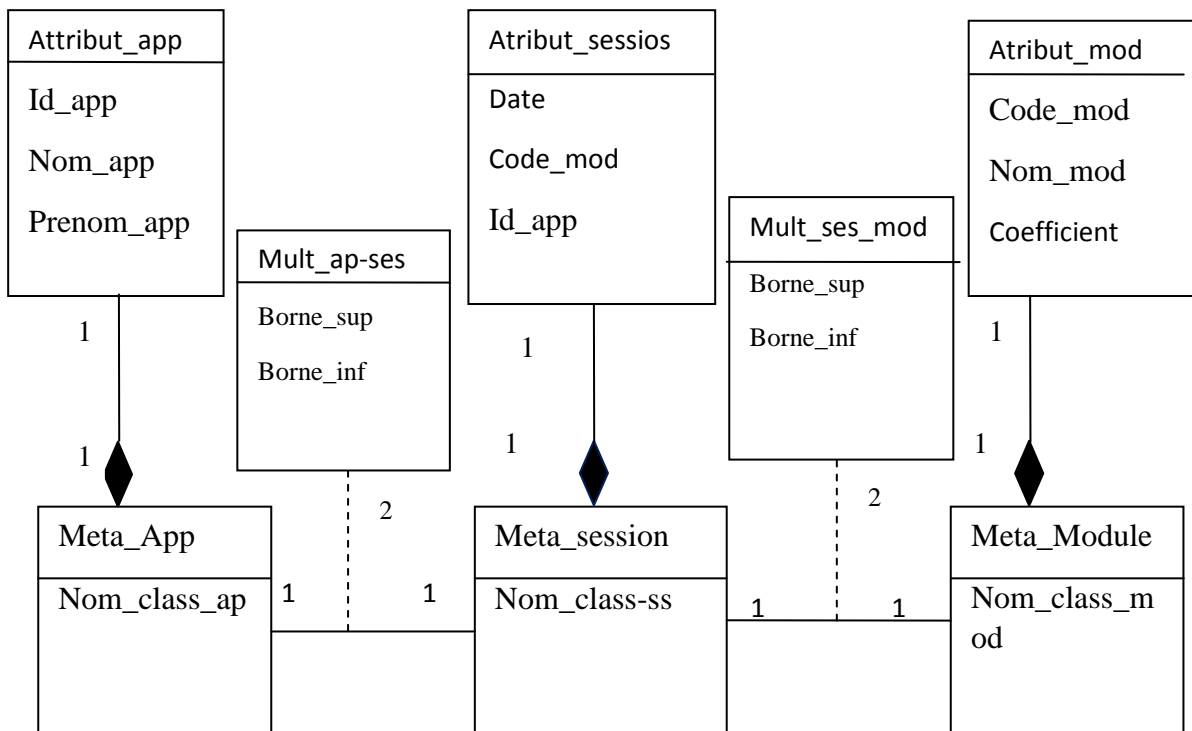


Figure III. 9: Méta-modèle pour gestion de l'inscription d'un apprenant à une matière

Nous allons prendre méta-modèle pour le rôle de l'administrateur pour montrer comment nous aurons le modèle conforme au méta-modèle décrit auparavant correspond a sit1 par exemple:

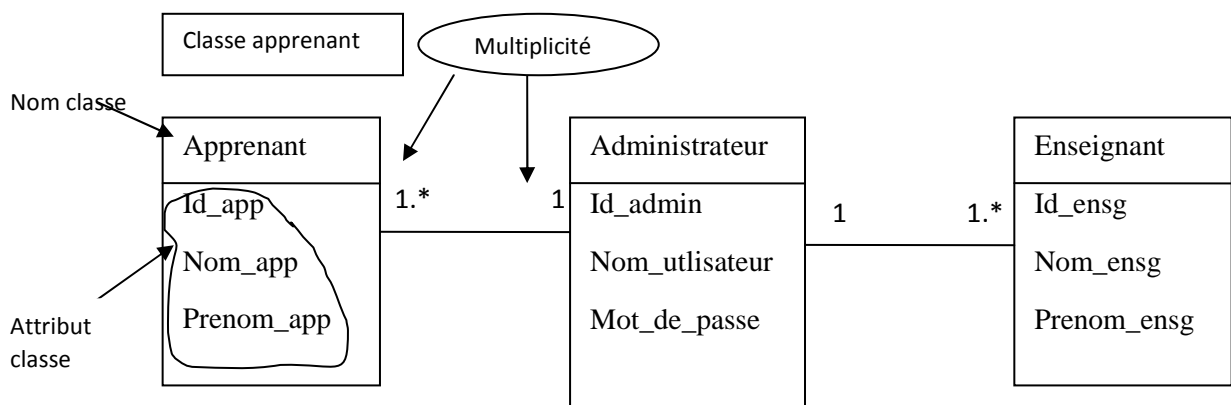


Figure III. 10: diagramme de classe rôle de l'administrateur

III.6.2 Transformation de diagramme de classe à un modèle relationnelle

Pour aller de diagramme de classe à modèle relationnelle on prend en considération les règles de transformations suivantes :

Règle1: présence de la cardinalité (?..1) d'un côté de l'association

- Chaque classe se transforme en une table
- Chaque attribut de classe se transforme en un champ de table
- L'identifiant de la classe qui est associée à la cardinalité (?..1) devient la clé étrangère de l'autre classe

Règle2: présence de (?..N) des deux côtés de l'association

- Chaque classe se transforme en une table
- Chaque attribut de classe se transforme en un champ de table
- L'association se transforme en une table. Cette table a comme champs l'identifiant de chacune des deux classes, plus d'éventuels autres attributs.

- Règle3: présence d'une généralisation

Méthode 1:

- Créer une table avec tous les attributs des classes
- Ajouter un attribut pour distinguer les types des objets

Méthode 2:

- Créer une table pour chaque sous type, chaque table se compose des attributs génériques et d'attributs spécifiques

Méthode 3:

- Créer une table par classe et des associations

III.6.3 Modèle relationnel conforme à notre Méta-modèle

Après qu'on a suivi les règles de transformation, nous donnerons ici le modèle relationnel du site1 ;

Cours (num_cours, nom_cours, repertoire_cours)

Module (Code_mod,nom_mod,coefficient)

Mod_cours(num_cours , Code_mod)

Test (num_ExTest, Titre_ExTest, lien_ExTest)

Exercice (num_Exe, Titre_Exe, lien_Exe)

Note (num_note, val_note,id_app*,Code_mod*)

Apprenant (id_app, nom_utilisateur, mot_passe, adr_per, num_g*,)

enseignant (id_ens, nom_ens,prenom_ens, nom_utilisateur, mot_passe, adr_per)

Administrateur (id_per, nom_utilisateur, mot_passe,)

Groupe (num_group, nom_group)

Session (date, id-app*, code_module*)

Conclusion

Au cours de réalisation de ce chapitre, nous avons pu construire une collecte d'information concernant le domaine de E_learning, dont nous avons montrer EAO, EIAO, FAD, en suite nous avons essayer d'élaborer notre méta-modèle pour les systèmes e-Learning avec quelques scénarios et nous avons terminer ce chapitre par un exemple de BDD conforme à notre méta-modèle



Chapitre VI
Implémentation et
Réalisation

VI.1 Introduction

Après avoir créé notre méta-modèle, nous atteindrons maintenant à présenter le dernier chapitre « Réalisation » qui décrit l'implémentation de la solution proposé durant la conception.

Ce chapitre réalisation sera présenté comme suit, nous commencerons par la description de l'environnement et les outils de développement, puis nous passerons à la présentation de l'architecture de notre application, ensuite les fonctionnalités de l'application, enfin nous terminerons par une conclusion.

VI.2 L'environnement de développement

VI.2.1 IDE (NetBeans 8.0)

C'est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et des pages web). NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS.

NetBeans est à l'origine un EDI Java. NetBeans fut développé à l'origine par une équipe d'étudiants à Prague, racheté ensuite par Sun Microsystems. Quelque part en 2002, Sun a décidé de rendre NetBeans open-source. Mais NetBeans n'est pas uniquement un EDI Java. C'est également une plateforme, nous permettant d'écrire nos propres applications Swing. Sa conception est complètement modulaire : Tout est module, même la plateforme. Ce qui fait de NetBeans une boîte à outils facilement améliorable ou modifiable. La licence de NetBeans permet de l'utiliser gratuitement à des fins commerciales ou non. Elle permet de développer tous types d'applications basées sur la plateforme NetBeans. Les modules que nous pourrions écrire peuvent être open- source comme ils peuvent être closed-source, Ils peuvent être gratuits, comme ils peuvent être payants.

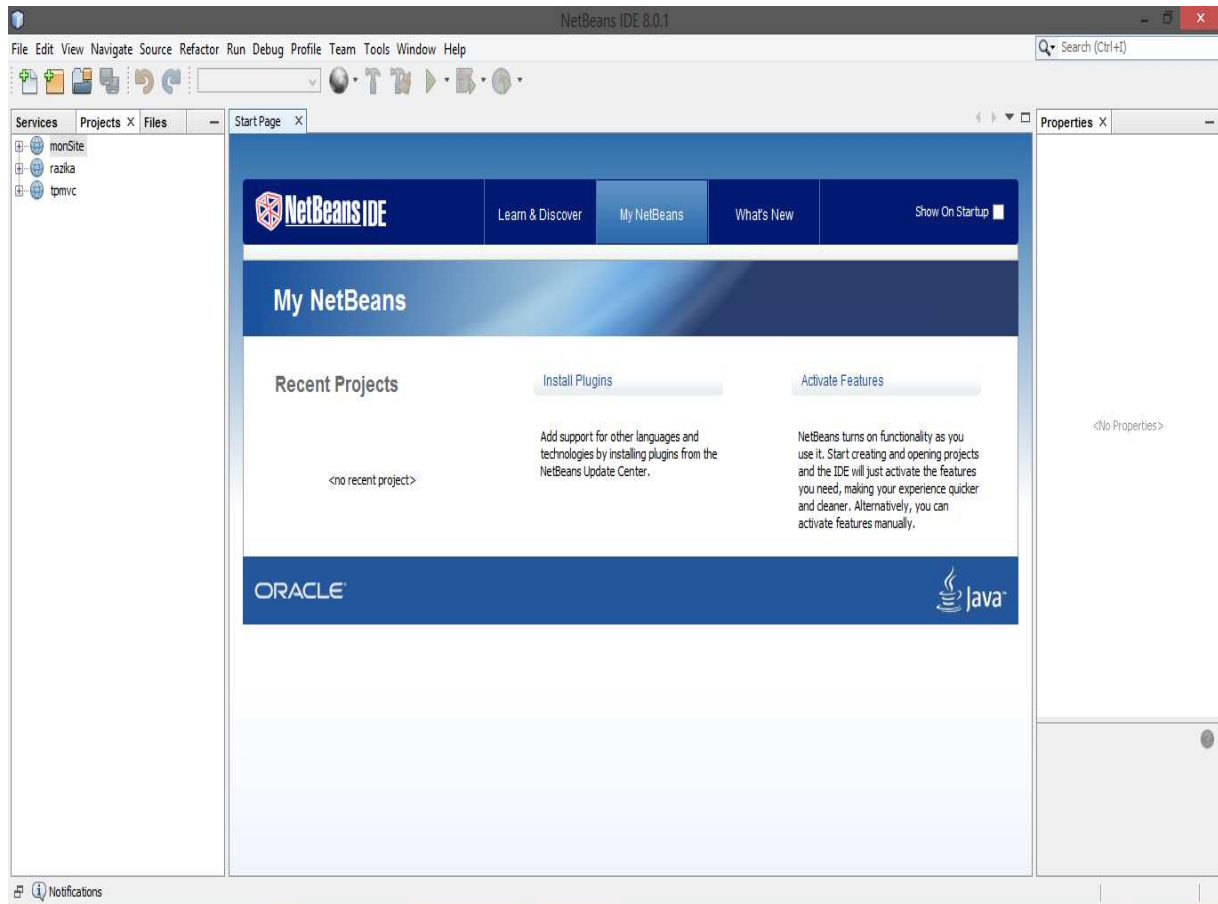


Figure VI. 1: Aperçu NetBeans (IDE)

Pourquoi l'IDE NetBeans ?

Signification et définition d'un IDE:

IDE (Integrated Development Environment) = (Environnement de développement Intégré)

Un IDE est un outil qui nous permettra d'augmenter notre productivité pour les développements d'entreprise.

Nous avons doté de choisir l'IDE « *NetBeans* », grâce à son adéquation aux développements d'entreprise, il nous permettra de créer une application sans écrire une ligne de code, en utilisant des composants prêts et en générant automatiquement le code nécessaire.

VI.2.2 Le langage de programmation java :

C'est un langage orienté objet. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java est apparu en fin d'années 1995 et début 1996 et développé par Sun Microsystems, inspiré des langages C et

C++ dont il hérite, le **polymorphisme** et l'**héritage**. Au départ, java est destiné à la programmation de centraux téléphoniques sous l'appellation de langage "oak". La société Sun a eu l'idée de le recentrer sur les applications de l'internet et des réseaux ; Java C'est un langage en évolution permanente Java 2 est la version stabilisée de java fondée sur la version initiale 1.2.2 du JDK (Java Development Kit de Sun : <http://java.sun.com>). Encore il s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur.

➤ Les objectifs de langage java :

Java a été développée dans le but d'augmenter la productivité des programmeurs. Pour cela, ses quatre raisons principales de développement ont été mises en place :

- C'est un langage orienté objet dérivé du C, mais plus simple à utiliser et plus « pur » que le C++. On entend par « pur » le fait qu'en Java, on ne peut faire que de la programmation orienté objet contrairement au C++ qui reste un langage hybride, c'est-à-dire autorisant plusieurs styles de programmation. C++ est hybride pour assurer une compatibilité avec le C.
- Il est doté, en standard, de bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphisme), la programmation multi-threads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau ... L'utilisation de ces bibliothèques facilitent grandement la tâche du programmeur lors de la construction d'applications complexes.
- Il est doté, en standard, d'un mécanisme de gestions des erreurs (les exceptions) très utile et très performant. Les exceptions représentent le mécanisme de gestion des erreurs intégré au langage Java. Il se compose d'objets représentant les erreurs et d'un ensemble de trois mots clés qui permettent de détecter et de traiter ces erreurs (try, catch et finally) et de les lever ou les propager (throw et throws).Ce mécanisme, inexistant en C, existe en C++ sous la forme d'une extension au langage beaucoup moins simple à utiliser qu'en Java. Ces mécanismes permettent de renforcer la sécurité du code Java
- Il est multi plates-formes : les programmes tournent sans modification sur tous les environnements où Java existe (Windows, Unix et Mac).c-à-dire c'est un langage de programmation qu'on puisse compiler et exécuter sur énormément de plate-forme.

➤ **Les caractéristiques de langage java :**

- ✚ Java est interprété : Le code source est compilé en pseudo code ou Byte code puis exécuté par un interpréteur Java qui est : la Java Virtual Machine (JVM). Ce concept est à la base du slogan de Sun pour Java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le Byte code, s'il ne contient pas de code spécifique à une plate-forme particulière peut être exécuté et obtenir les mêmes résultats sur toutes les machines disposant d'une JVM. Autrement dit le Byte code est normalisé, peut être exécuté sur plusieurs plates formes, et exploitable par tous les systèmes d'exploitations.
- ✚ Java est portable : il est indépendant de toute plate-forme : il n'y a pas de compilation spécifique pour chaque plate-forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du byte code.
- ✚ Java est orienté objet : comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).
- ✚ Java est simple : le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...
- ✚ Java est fortement typé : toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.
- ✚ Java assure la gestion de la mémoire : l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.
- ✚ Java est robuste et sûr :
 - Peu de pièges.
 - Pas de pointeurs, pas de fonctions d'arguments variables.
 - Compilateurs très stricts car toutes les valeurs doivent être initialisées.

- Le traitement des exceptions est obligatoire.
- Les erreurs à l'exécution sont vérifiées tout comme les limites des tableaux.

- ✚ Java est économe : le pseudo code a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.
- ✚ Java est multitâche : il permet l'utilisation de threads qui sont des unités d'exécution isolées. La JVM, elle-même, utilise plusieurs threads. Un thread est une unité d'exécution faisant partie d'un programme. Cette unité fonctionne de façon autonome et parallèlement à d'autres threads. En fait, sur une machine mono processeur, chaque unité se voit attribuer des intervalles de temps au cours desquels elles ont le droit d'utiliser le processeur pour accomplir leurs traitements.

Le principal avantage des threads est de pouvoir répartir différents traitements d'un même programme en plusieurs unités distinctes pour permettre leur exécution "simultanée".

La classe `java.lang.Thread` et l'interface `java.lang.Runnable` sont les bases pour le développement des threads en java. Par exemple, pour exécuter des applets dans un thread, il faut que celles-ci implémentent l'interface `Runnable`.

Pourquoi le langage java ?

Notre choix est basé sur le langage de programmation *java*, d'une part parce qu'il est un langage de programmation orientée objet qui n'est qu'un ensemble d'objets autonomes et responsables, qui s'entraînent pour résoudre un problème final en s'envoyant des messages, d'autre part les principes de la programmation orientée objet sont nettement plus élaborés dans le langage *java* ; Bien sûr ne sont pas spécifiques à *java*, mais de nombreux programmeurs les découvrent au même temps que le langage *java*. Comme nous pourrions justifier notre choix encore, par l'adéquation de *java* aux concepts du monde réel ; en *java* tout est objet ! *Java* est le rare langage qui est proche aux concepts de la vie réelle.

Exemple de code java

```
class test1 {  
    public static void main (String args[]) {  
        char code = 'D';  
        int index = code - 'A';  
        System.out.println("index = " + index);  
    }  
}
```

VI.2.3 J2EE :

J2EE est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

- un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'applications.
- un ensemble d'API qui peut être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

VI.2.4 Représentation de servlet :

Une servlet est un programme qui s'exécute côté serveur en tant qu'extension du serveur. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat. La liaison entre la servlet et le client peut être directe ou passer par un intermédiaire comme par exemple un serveur http.

Même si pour le moment la principale utilisation des servlets est la génération de pages html dynamiques utilisant le protocole http et donc un serveur web, n'importe quel protocole reposant sur le principe de requête/réponse peut faire usage d'une servlet.

Ecrit en java, une servlet en retire ses avantages : la portabilité, l'accès à toutes les API de java dont JDBC pour l'accès aux bases de données, ...

Une servlet peut être invoquée plusieurs fois en même temps pour répondre à plusieurs requêtes simultanées. La servlet se positionne dans une architecture Client/Serveur trois tiers dans le tiers du milieu entre le client léger chargé de l'affichage et la source de données.

VI.2.5 Présentation des JSP

Les JSP permettent d'introduire du code Java dans des tags prédéfinis à l'intérieur d'une page HTML. La technologie JSP mélange la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client.

Une JSP est habituellement constituée : de données et de tags HTML de tags JSP de scriptlets (code Java intégré à la JSP). Les fichiers JSP possèdent par convention l'extension jsp.

Concrètement, les JSP sont basées sur les servlets. Au premier appel de la page JSP, le moteur de JSP génère et compile automatiquement une servlet qui permet la génération de la page web. Le code HTML est repris intégralement dans la servlet. Le code Java est inséré dans la servlet.

La servlet générée est compilée et sauvegardée puis elle est exécutée. Les appels suivants de la JSP sont beaucoup plus rapides car la servlet, conservée par le serveur, est directement exécutée.

Il y a plusieurs manières de combiner les technologies JSP, les beans/EJB et les servlets en fonction des besoins pour développer des applications web.

Comme le code de la servlet est généré dynamiquement, les JSP sont relativement difficiles à déboguer.

VI.2.6 Représentation de MySQL

Définition MySQL

➤ Présentation de système de gestion de base de données MySQL :

MySQL, le plus populaire des serveurs de bases de données SQL Open Source, est développé, distribué et supporté par **MySQL AB**. **MySQL AB** est une société commerciale, fondée par les développeurs de MySQL, qui développent leur activité en fournissant des services autour de MySQL.

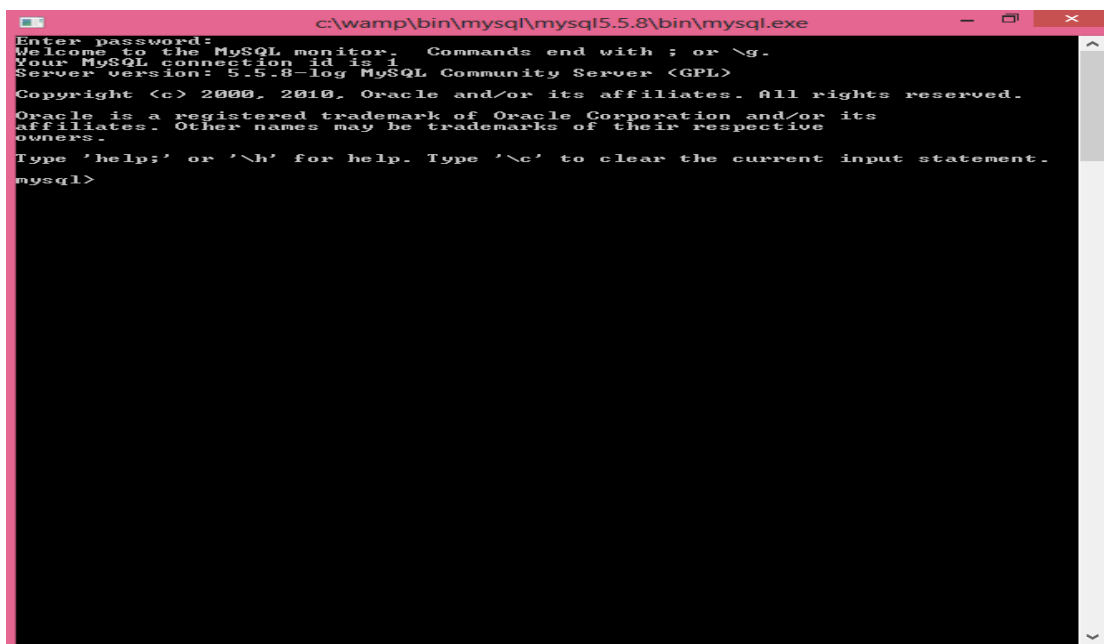
- **MySQL** est un système de gestion de base de données SQL :

Il se compose d'un langage de définition de données et de droits ainsi qu'un langage de manipulation des données. En effet, si (on a besoin d'un système de gestion de bases de données tel que le serveur MySQL pour ajouter, lire et traiter des données dans une base de données et comme les ordinateurs sont très bons à manipuler de grandes quantités de données) alors le SGBD MySQL joue un rôle central en informatique.

- **MySQL** est disponible sous Windows et sous Unix.
- **MySQL** est un serveur de bases de données relationnelles :

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données. SQL est le plus populaire langage de base de données dans le monde. SQL est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information. Par exemple, on peut utiliser le SQL pour récupérer des informations sur un produit ou stocker des informations client sur un site web.

- *MySQL* est un véritable serveur (ou gestionnaire) de base de données SQL multi-utilisateur et multi-threadé.
- *MySQL* est une configuration client/serveur.
- *MySQL* est suffisamment rapide et flexible pour gérer des historiques et des images.



```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.8-log MySQL Community Server <GPL>
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Figure VI. 2: Aperçu MySQL

VI.2.7 XML

XML est l'acronyme de «eXtensible Markup Language». Sous-ensemble simplifié du langage SGML (Standard Generalized Markup Language), permet de décrire des informations organisées sous une forme hiérarchique à l'aide de balises entre <> de façon similaire au langage HTML.

XML permet d'échanger des données entre applications hétérogènes car il permet de modéliser et de stocker des données de façon portable.

XML est extensible dans la mesure où il n'utilise pas de tags prédéfinis comme HTML et il permet de définir de nouvelles balises : c'est un métalangage.

Le format HTML est utilisé pour formater et afficher les données qu'il contient : il est destiné à structurer, formater et échanger des documents d'une façon la plus standard possible.

XML est utilisé pour modéliser et stocker des données. Il ne permet pas à lui seul d'afficher les données qu'il contient.

Pourtant, XML et HTML sont tous les deux des dérivés d'un langage nommé SGML (Standard Generalized Markup Language). La création d'XML est liée à la complexité de SGML. D'ailleurs, un fichier XML avec sa DTD correspondante peut être traité par un processeur SGML.

XML et Java ont en commun la portabilité réalisée grâce à une indépendance vis à vis du système et de leur environnement.

Au premier abord, XML est un langage assez déroutant car il nous laisse la liberté de décider du nom des balises à utiliser et n'impose que peu de règles de construction. Cette souplesse lui a permis d'être adopté très rapidement par la communauté informatique, qui voit XML comme un des meilleurs formats pour structurer lisiblement des données et les échanger dans le monde ouvert d'Internet.

Un document XML débute généralement par le prologue :

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

Ce prologue peut comporter aussi un attribut encoding qui spécifie le type d'encodage des caractères utilisé dans le document ; comme encoding est égal par défaut à UTF-8, la ligne précédente est donc équivalente à :

```
<? Xml version="1.0" encoding="UTF-8"?>
```

Suivent ensuite les données du document décrites sous une forme arborescente d'*éléments* inclus les uns dans les autres, un document XML ne devant contenir qu'un seul élément racine. Chaque *élément* commence par une *balise (tag)* de début notée entre les symboles < et > et se termine par une balise de fin notée entre les symboles </ et >. Chaque balise qui porte le nom de votre choix, est utilisée pour nommer un élément et décrire ses données avec :

- une suite d'*attributs* dans sa balise de début, chaque attribut s'écrivant avec un nom suivi du signe = et d'une valeur entre ' ' ou entre " " (par exemple <messages sujet="chat" >);
- et/ou des *données textuelles* et/ou d'autres éléments *enfants (child)* entre ses balises de début et de fin.

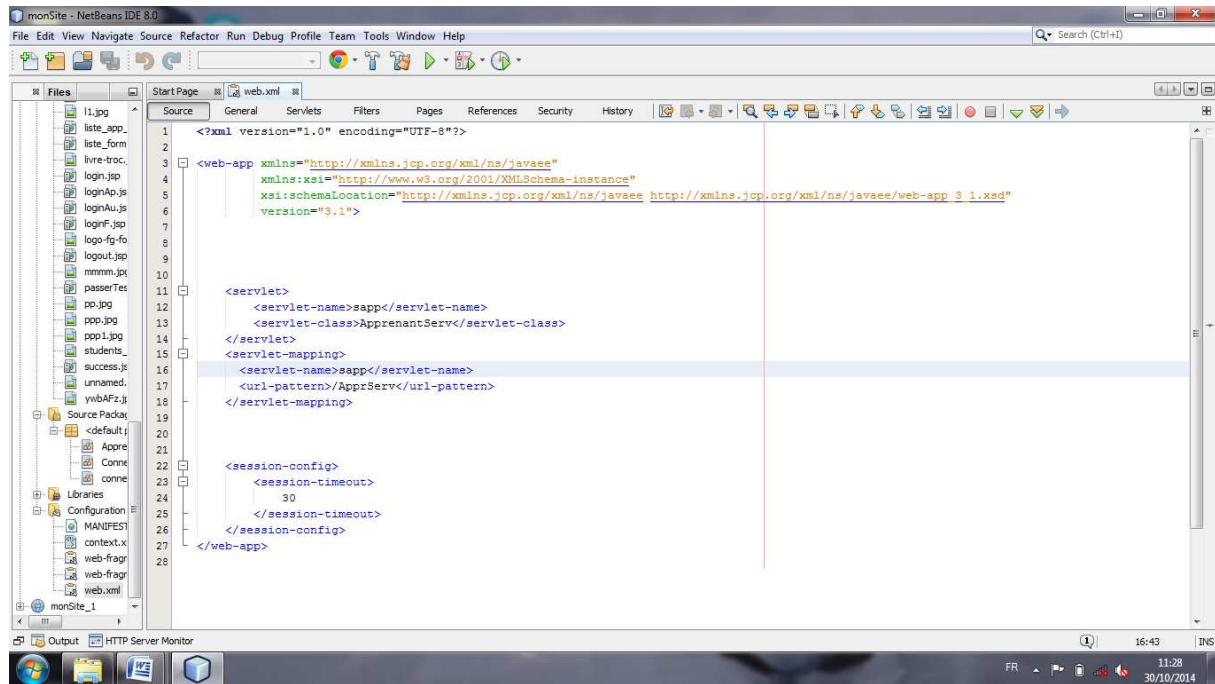


Figure VI. 3 : Aperçu fichier XML

VI.2.8 Apache tomcat

Tomcat, développé par Apache (www.apache.org), est une norme la mise en œuvre de référence pour les servlets Java et les JSP. Ca peut être utilisé de manière autonome comme un serveur Web ou être branché dans un site Web serveur comme Apache, Netscape Enterprise Server ou Microsoft

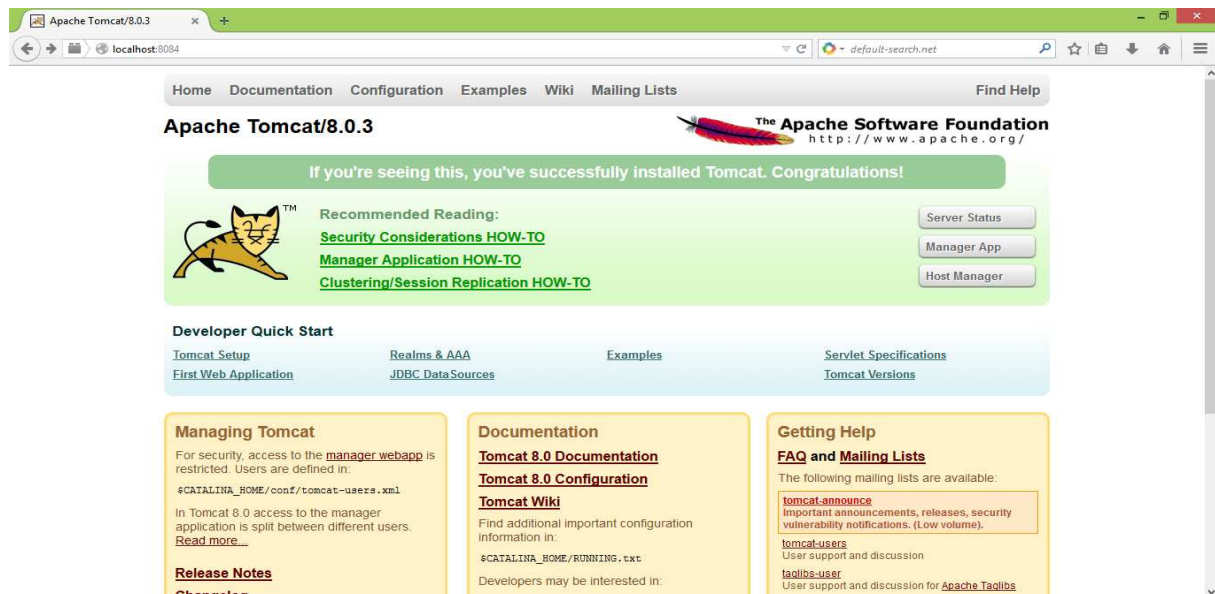


Figure VI. 4: Apache tomcat

VI.3 Quelques captures d'écran de notre application

Premier site web :

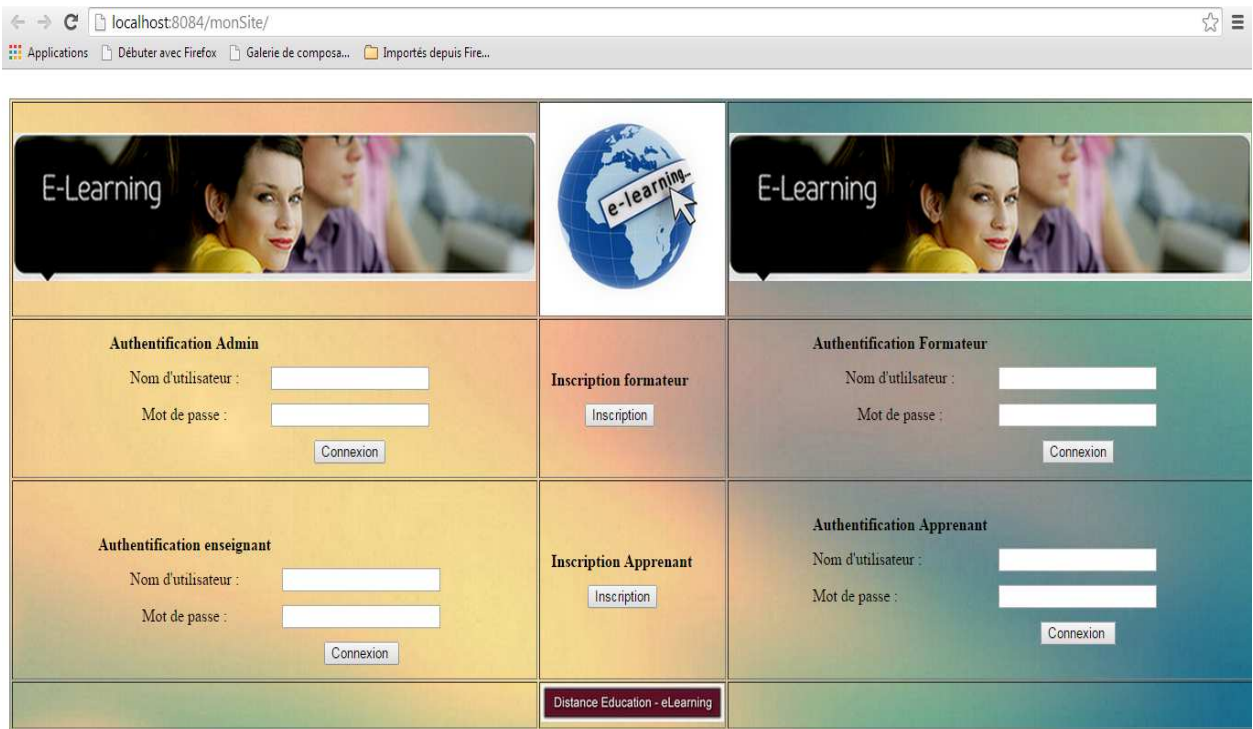


Figure VI.5: Page d'accueil du 1^{er} Site

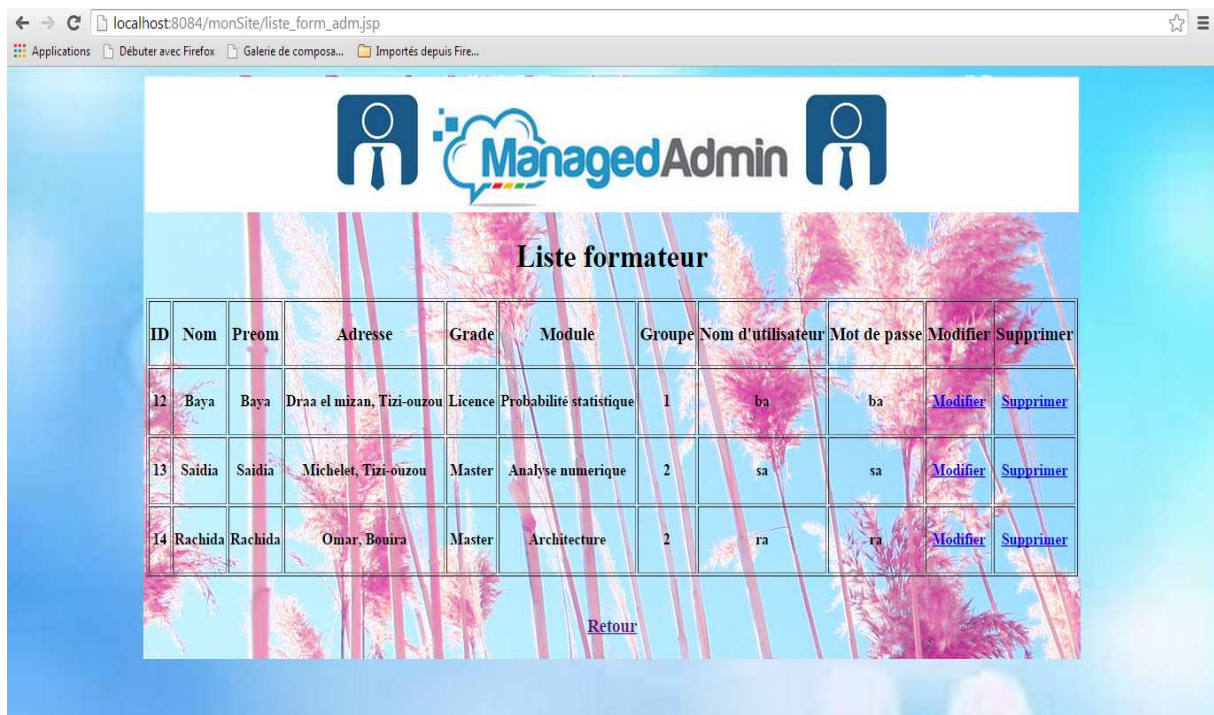


Figure VI.6: Liste des formateurs du 1^{er} Site.

Deuxième site web :

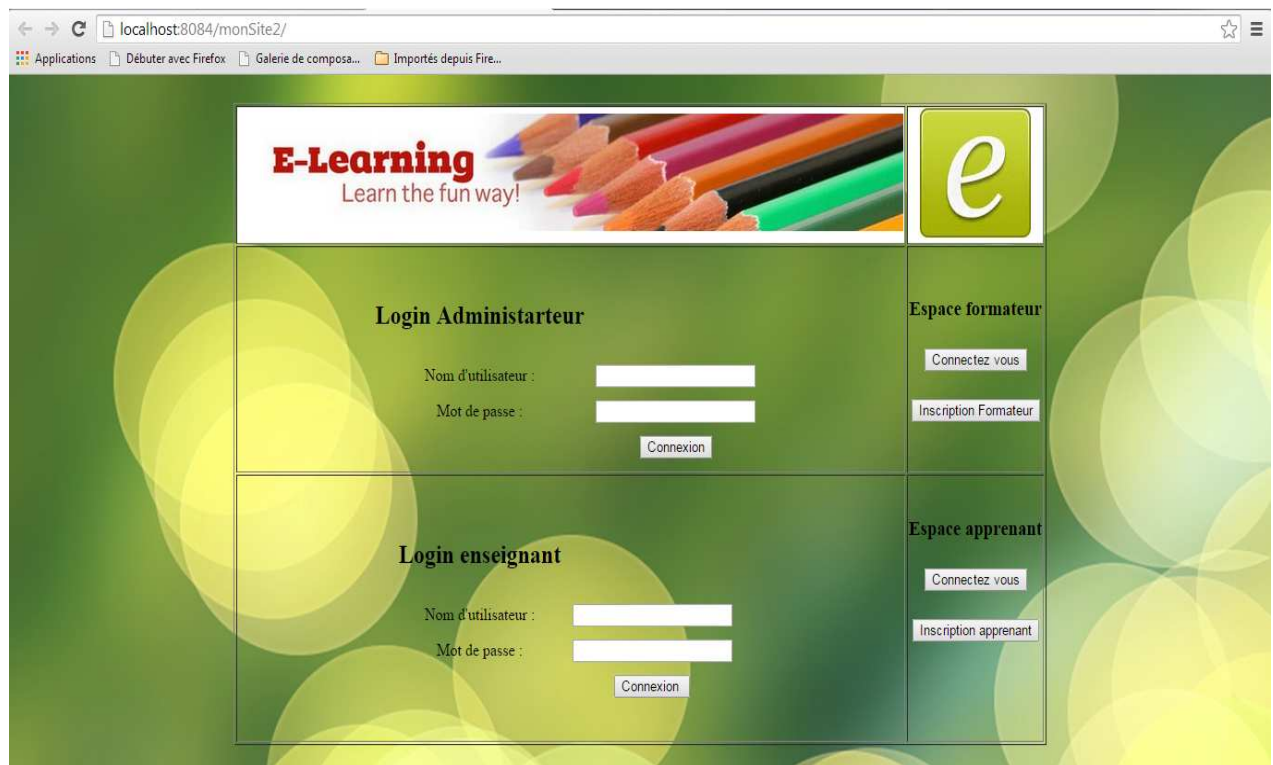


Figure VI.7: Page d'accueil du 2^{ème} Site.

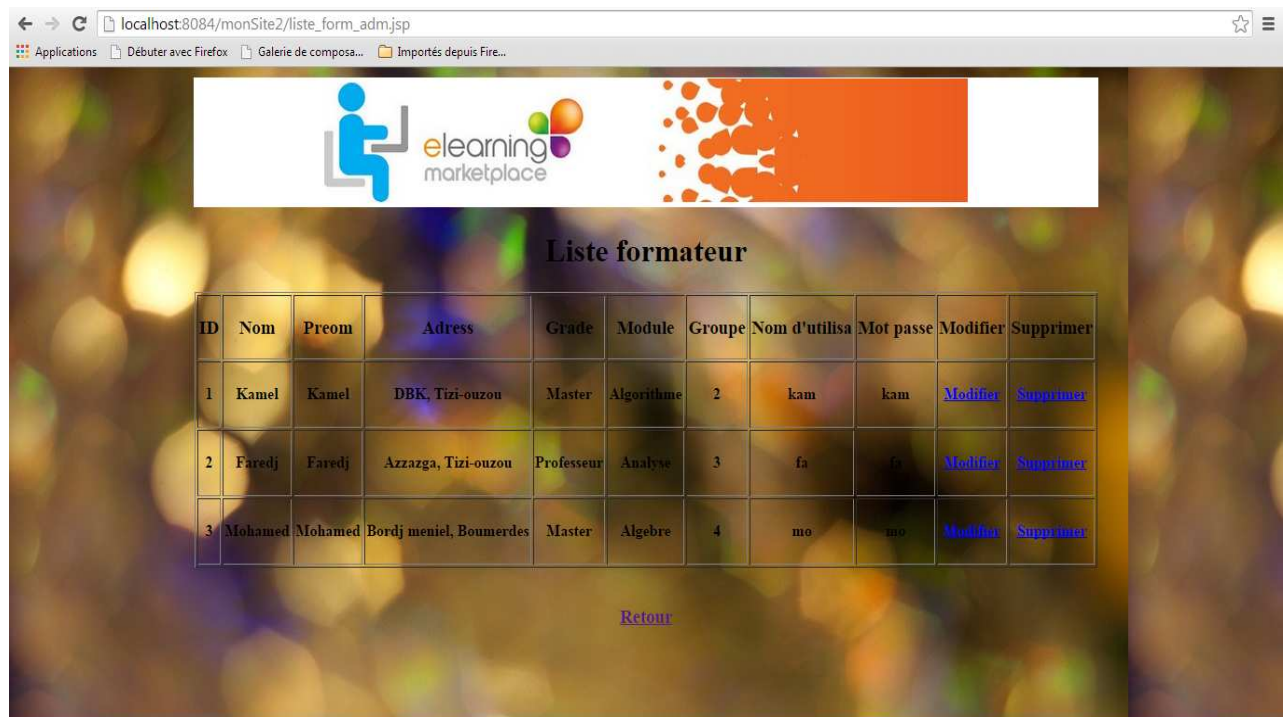


Figure VI.8: Liste des formateurs du 2^{ème} Site.

Application d'intégration :



Figure VI.9: Page d'accueil d'application d'intégration.

ID	Nom	Preom	Adresse	Grade	Module	Groupe	Nom d'utilisa	Mot de passe
12	Baya	Baya	Draa el mizan, Tizi-ouzou	Licence	Probabilité statistique	1	ba	ba
13	Saidia	Saidia	Michelet, Tizi-ouzou	Master	Analyse numerique	2	sa	sa
14	Rachida	Rachida	Omar, Bouira	Master	Architecture	2	ra	ra
1	Kamel	Kamel	DBK, Tizi-ouzou	Master	Algorithmie	2	kam	kam
2	Faredj	Faredj	Azzouza, Tizi-ouzou	Professeur	Analyse	3	fa	fa
3	Mohamed	Mohamed	Boujamel, Boumerdes	Master	Algebre	4	mo	mo

Figure VI.10: Liste des formateurs d'application d'intégration.

Conclusion :

Durant ce chapitre « Réalisation », on a présenté d'abord les outils de développement nécessaires à la réalisation de notre application.

Conclusion générale

L'intégration des systèmes d'information constitue aujourd'hui l'un des facteurs majeurs du succès d'une fusion-acquisition d'entreprises. Le génie logiciel est une science de l'ingénieur dont la finalité est la fabrication de systèmes informatisés. Ces derniers prennent en charge de nombreuses fonctions relatives au traitement de l'information. Au sein de cette discipline, l'intégration concerne une phase du cycle de vie d'un développement de systèmes informatiques. L'intégration y est perçue comme les interventions visant à mettre en relation les applications fonctionnelles et les bases de données où sont saisies, distribuées et actualisées les informations. Il s'agit là d'une opération dépassant la simple interconnexion des applications puisqu'elle induit une mise en commun de ressources.

Dans le but d'intégration des systèmes d'information ; l'interopérabilité est une caractéristique majeure dans des applications d'intégration, pour cela le développement sera indépendant de la plateforme ; donc nous utilisons le principe de transformation modèle MDA pour construire les applications d'intégration ; à partir de méta-modèle que nous avons démontré, nous avons donné un exemple de modèle sous forme de diagramme de classe conforme au méta-modèle puis nous avons transformé ce modèle en modèle relationnelle ;

Enfin, on espère que notre travail sera améliorée par d'autre personne intéressée par ce sujet parce que l'intégration des systèmes information on utilisant le méta-modèle est un sujet nouveau, vaste.

Bibliographie :

- [1] Guy Caplat ; modèles et méta-modèle ; Presse polytechnique et universitaire remandes ; 2008
- [2] Intégration des Systèmes Clients/Serveurs ; Andre Freyssinet PDF (www.i3s.unice.fr/~riveill/cours/car/00-integration-client-serveur.pdf)
- [3] D.Konstantas,J-P Bourriet ;Interopirabilité of Entreprise software and application ; british Library Cataloguinig in publication data ;2005.
- [4] Thèse: Matthieu Lemerre, Intégration de systemes heterogenes en termes de niveaux de securite ; Université de Paris Sud – Orsay promo : 2009. (tel.archives-ouvertes.fr/tel-00440329)
- [5] L'évolution des besoins et des solutions d'intégration, Philippe COURTY, CFPIM
- [6] www.csc.com/fr/ds/71138/713064_règles_d_or_pour_réussir_une_fusion_de_systèmes_d_information_bancaires.
- [7] Annie Authosserre-Cavarero, F. Bertrand, et al. ; Interopérabilité des systèmes d'information : approches diriées par les modèles ; (<https://hal.archives-ouvertes.fr/file/index/docid/707536/filename/Auth12a-interop-si-2012.pdf>)
- [8] http://www.fmpusmba.ac.ma/umvf/UMVFMiroir/c2i/c2i/Nice_STACCINI_Pascal_P02/STACCINI_Pascal_P02.pdf
- [9] <http://definition-interoperabilite.info/>
- [10] Karim Zarour; L'interopérabilité des systèmes d'information médicaux: une approche basée agent; Constantine le 22/02/2012 (www.umc.edu.dz/buc/theses/informatique/ZAR6117.pdf)
- [11] <http://www.microsoft.com/france/openness/open-source/interopabilite.aspx>.
- [12] [http://msdn.microsoft.com/fr-fr/library/office/bb610378\(v=office.14\).aspx](http://msdn.microsoft.com/fr-fr/library/office/bb610378(v=office.14).aspx).

- [13] <http://www-rp.lip6.fr/~ms/Francais/Interop.htm>
- [14] [http://msdn.microsoft.com/fr-fr/library/dd233112\(v=vs.110\).aspx](http://msdn.microsoft.com/fr-fr/library/dd233112(v=vs.110).aspx).
- [15] http://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9_en_informatique.
- [16] <http://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9>
- [17] <http://www.gralon.net/articles/internet-et-webmaster/logiciel/article-l-interop%C3%A9rabilit%C3%A9--definition-et-enjeux-3750.htm>
- [18] http://wiki.april.org/w/Synth%C3%A8se_interop%C3%A9rabilit%C3%A9
- [19] <http://msdn.microsoft.com/fr-fr/library/dd233112%28v=vs.110%29.aspx>
- [20] N. Moalla, H. Panetto, X. Boucher; Interop%C3%A9rabilit%C3%A9 et partage de connaissances. (https://hal.archives-ouvertes.fr/hal-00722742/PDF/Moalla_et_al_-_ISI.pdf)
- [21] http://fr.wikipedia.org/wiki/Syst%C3%A8me_d'information
- [22] <http://fr.wikipedia.org/wiki/Syst%C3%A8me>
- [23] Syst%C3%A8me d'information cours. (http://perso.modulonet.fr/~placurie/Ressources/BTS1-Cgo1/Chap_4_Syst%C3%A8me_information.pdf)
- [24] Antoine Zimmerman ; « Conception de Syst%C3%A8mes d'Information » P%C3%B4le informatique, %C3%89cole Nationale Sup%C3%A9rieure des Mines de Saint-%C3%89tienne, 2013/2014 (http://www.emse.fr/~zimmermann/Teaching/CSI/cours_CSI.pdf)
- [25] <http://fr.wikipedia.org/wiki/Int%C3%A9gration>
- [26] http://www.csc.com/fr/ds/11439/13771-l_int%C3%A9gration_de_syst%C3%A8mes_et_de_solutions
- [27] <http://www.human-side.com/wilber/Integration/Integration.htm>
- [28] G. BRUNETTO « Fusion d'entreprises et int%C3%A9gration des syst%C3%A8mes d'information » ; Laboratoire crego, Universit%C3%A9 montpellier II. (www.strategie-aims.com/events/conferences/8-xveme-conference-de-l-aims/communications/2261-fusion-dentreprises-et-integration-des-systemes-dinformation/download)

[29] <http://www.dhis2.org/doc/snapshot/fr/implementer/html/ch07.html>

[30] Jacques PRINTZ ; « Génie logiciel, cours B: Le cycle de vie du logiciel »
CNAM/Médias, 1995

[31] « L'interopérabilité en informatique »

<http://storage.canalblog.com/60/99/974438/75906138.pdf>

[32] Sébastien GEORGE «Apprentissage collectif à distance. SPLACH : un environnement informatique support d'une pédagogie de projet », l'Université du Maine Soutenu le 11 juillet 2001

<https://tel.archives-ouvertes.fredutice-00000207document>

[33] (E-learning : présentation, aspects, enjeux et avenir) UNIVERSITE DE SFAX POUR LE SUD L'ECOLE NATIONALE D'INGENIEURS DE SFAX (ENIS), Mémoire présenté pour l'obtention du diplôme de mastère spécialisé en management de l'ingénierie ; Soutenu le Dimanche 16 Février 2003

http://procomptable.com/papier_recherche/mmbf.htm

[34] Ammar BRIKA, Développement du E-learning à l'I. N.I : Mise en ligne d'un outil de simulation d'un cours d'algorithmique, Soutenu le 04 juillet 2007.

<http://share.esi.dz/36/1/D%C3%A9veloppement%20du%20Elearning%20%C3%A0%20l'I.%20N.I%20%20mise%20en%20ligne%20d'un%20outil%20de%20simulation%20d'un%20cours%20d'algorithmique.pdf>

[35] Mr Kerbiche M'hand (Patron de conception pour une solution d'interopérabilité) 2008

[36] http://www.memoireonline.com/07/08/1413/m_modelisation-systeme-multi-agents-hypermedia-educatif4.html

[37] <http://tecfa.unige.ch/tecfa/publicat/mendel-papers/apslf92.html>