

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études

Présenté en vue de l'obtention
du Diplôme de Master en Informatique

Thème:

**Conception de plate forme télé-TP basé
sur le web service**

Dirigé par :

Mr. Remdane Mohamed

Présenté par :

Mr. Aouam Djamel

Mr. Hadj Ali Yacine

Promotion 2012/2013

Remerciements

À travers ce modeste travail, nous tenons à remercier vivement notre promoteur Mr REMDANE Mohamed, Med Nabil et Mr qui nous as honoré en acceptant de nous encadrer et pour lces précieux conseils et orientations, ainsi que la confiance qu'il a placé en nous tout au long de la préparation de ce projet.

Nos remerciements et gratitudes s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner notre travail.

Nous remercions enfin tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Dédicaces

Je dédie cet humble travail :

A mes chers parents qui m'ont toujours soutenu par leur indéfectible amour, leur patience et leur réconfort.

A ma sœur Lilia et à son mari Malik,

A ma sœur Zola et son mari Fateh.

A mon petit frère Sofiane, à qui je tiens énormément.

Mes nièces Kenza-Manel et Ania que j'aime tant, ainsi que mon neveu Mohamed

A toute ma famille qui est, et qui sera toujours là pour moi.

A mon binôme et sa famille.

A tous mes amis(es) sans eux je n'arriverai pas à avoir tout le courage et toute la volonté de résister aux moments durs.

A toutes les personnes qui m'aiment, à ceux qui ne sont pas ici avec moi mais qui sont toujours dans mon cœur.

Aouam Djamel.

Dédicaces

Je dédie cet humble travail :

A mes chers parents qui m'ont toujours soutenu par leur indéfectible amour, leur patience et leur réconfort.

A mes sœurs dihya, monia et imen ;

A toute ma famille qui est, et qui sera toujours là pour moi.

A mon binôme et sa famille.

A tous mes amis(es) sans eux je n'arriverai pas à avoir tout le courage et toute la volonté de résister aux moments durs.

A toutes les personnes qui m'aiment, à ceux qui ne sont pas ici avec moi mais qui sont toujours dans mon cœur.

Hadj Ali Yacine.

SOMMAIRE

Introduction générale.....	1
Chapitre I : Services Web	
I. Introduction.....	2
II. Historique	2
III. Présentation des services Web	3
III.1.définition d'un service Web	3
III.2.Invocation des services Web	5
III.3.Fonctionnement globale des services web	6
IV. Avantages des web services	7
V. Les technologies des Web Services.....	8
V.1.XML et les services Web.....	8
V.2.Le protocole SOAP (<i>Simple Object Access Protocol</i>).....	9
V.2.1.Définition du protocole SOAP.....	10
V.2.2.Caractéristiques du protocole SOAP	10
V.2.3.Modèle SOAP d'échange de messages.....	10
V.2.3.1.Document XML comme message.....	11
V.2.3.2.Emetteurs et récepteurs	11
V.2.3.3.Chaines de messages	11
V.2.4.Structure des messages SOAP	12
V.2.5.Le transport	13
V.2.5.1.SOAP via http	13
V.2.5.2.En-tête SOAPAction.....	15
V.2.5.3.Codes de statut	15
V.2.6.SOAP et RPC.....	15
V.2.6.1.L'appel	15
V.2.6.2.Réponse.....	16
V.2.6.3.RPC et http.....	16
V.3.Le langage WSDL (Web Service Description language).....	17
V.3.1Présentation de WDSL.....	17
V.3.2.Pourquoi WSDL ?.....	18
V.3.3.Syntaxe WDSL	18
V.3.4.Exemple d'un document WSDL.....	19
V.4.L'Annuaire UDDI.....	23

SOMMAIRE

V.4.1. Définition	23
V.4.2. Pages de l'annuaire	23
V.4.3. Mécanismes d'accès aux services fournis par l'UDDI	24
V.4.4. Le modèle de données dans UDDI	25
V.4.5. L'interface UDDI.....	26
V.4.6. L'usage de l'annuaire UDDI	27
VI. La sécurité des web services	28
VII. Conclusion	30

Chapitre II: e-Learning et télé-TPs

I. Introduction.....	32
II. E-Learning	32
II.1. L'enseignement assisté par ordinateur	32
II.1.1- Les logiciels de l'EAO ou didacticiels	32
II.2. l'Enseignement Intelligemment Assisté par Ordinateur (EIAO).....	33
II.2.1. Architecture d'un système d'EIAO.....	34
II.2.1.1. L'interface élève (Home/Machine)	35
II.2.1.2. Le pédagogue	e35
II.2.1.3. L'expert	35
II.2.1.4. Le modèle élève	35
II.2.1.4.1. Le modèle de recouvrement (modèle overlay)	36
II.2.1.4.2. Le modèle des erreurs (modèle bugs).....	36
II.2.1.4.3. Le modèle différentiel	36
II.2.2. Les modes d'interaction	37
II.2. 2.1. Mode libre	37
II.2.2.2. Mode guidé.....	37
II.2.2.3. Mode mixte	37
II.3. Environnement Interactif pour Apprentissage avec Ordinateur (EIAO)	37
II.4. Hypertexte et Environnement Interactif pour Apprentissage avec Ordinateur EIAO	38
II.4.1. Définition de l'hypertexte	38
II.4.2. Influence des hypertextes sur l'EIAO	38
II.5. Environnement Interactif pour l'Apprentissage Humain (EIAH)	38

SOMMAIRE

II.6. E-Learning.....	39
II.6.1. Définitions des concepts	39
II.6.1.1. Qu'est-ce que l'enseignement à distance ou formation à distance (EAD/FAD) ?	39
II.6.1.2. Qu'est-ce que la formation ouverte et à distance (FOAD) ?40	
II.6.1. 3. Qu'est-ce que la Téléformation ?.....	41
II.6.1.4. Définition du e-Learning (e-formation) ?.....	41
II.6.2. Apport des nouvelles technologies à l'enseignement	42
II.6.3. Les formes d'e –Learning	42
II.6.3.1. Les formes de communication	42
II.6.3.1.1. La forme synchrone.....	42
II.6.3.1.2. La forme asynchrone	43
II.6.3.2. Les formes d'enseignement.....	44
II.6.4. Les avantages et les faiblesses du e-Learning.....	44
II.6.4.1. Les avantages du e-Learning	44
II.6.4.2. Les faiblesses d'e-Learning.....	45
II.7. Plates -formes d'enseignement à distance.....	46
II.7.1 Définition	46
II.7.2 Les principaux critères que doivent vérifier les plates-formes	46
II.7.3 Outils de communication	47
II.7.4 Acteurs de la plate-forme LMS.....	48
II.7.5 Quelques fonctionnalités de la plate-forme LMS	49
II.7.5.1 Modules d'édition de contenus de cours.....	49
II.7.5.2 Gestion des inscriptions	49
II.7.5. 3 Dépôts de fichiers.....	49
II.7.5.4 Gestion des notes.....	50
II.7.6 Architecture d'une plate forme	50
II.7.7 Exemples de plates formes.....	52
III. Travaux pratiques à distance télé-tps	53
III.1.Introduction	53
III.2. Situation des télé-TPs en e-learning	53
III.3.Les travaux pratiques.....	54
III.3.1. Les objectifs pédagogiques d'un TP.....	54
III.3.2 Typologie d'un TP.....	54

SOMMAIRE

III.3.3. Typologie de situations (TP)	55
III.4. Les travaux pratiques et l'informatique.....	58
III.5. Les travaux pratiques à distance (télé-TP)	58
III.5.1. Définition d'un télé-TP.....	58
III.5.2 Facteurs de développements.....	59
III.5.3. Les apports des téléTP	60
III.5.4. Les limites des téléTP.....	61
III.5.5. Typologie de situations (téléTP).....	62
III.6. Les téléTP sur des systèmes virtuels (laboratoires virtuels).....	62
III.7. Les téléTP sur des systèmes physiques (laboratoires distants)	62
III.8. Les télé TP sur des systèmes distants- virtuels (laboratoires hybrides)	64
III.9. Laboratoire virtuel et laboratoire virtuel à distance	64
III.9.1. Laboratoire virtuel	65
III.9.2. Laboratoire à distance (laboratoire en ligne)	65
III.9.2.1. Architecture d'un labo en ligne.....	66
III.10. Exemple de laboratoire électronique (e-lab)	67
III.10.1. Le projet LVEST (Laboratoires virtuels pour l'éducation en science et technologie)	67
III.10.2. Le projet VITELS (laboratoire virtuel de télécommunications pour la Suisse).....	68
III.10.3. Le projet eMersion.....	69
III.10.4. Exemples de laboratoires en ligne hébergés dans l'environnement de diffusion ECLATE	69
IV. Conclusion	70

Chapitre III : Analyse et conception

I. Introduction.....	71
II. Définition de l'UML.....	71
III. Objectif du projet	71
III.1. Identification des acteurs et des besoins.....	72
III.1.1. Identification des acteurs et leurs besoins fonctionnels.....	72
IV. Diagrammes représentatifs.....	74
IV.1 Le diagramme de contexte de l'application	74

SOMMAIRE

IV.2 Diagramme des cas d'utilisation	75
IV.3 Détermination des diagrammes de cas d'utilisations	76
IV.4. Description textuelle des cas d'utilisation.....	82
V. Réalisation des cas d'utilisation	89
V. Conclusion.....	97
Chapitre IV : Réalisation	
I. Introduction.....	98
II. Outils technologiques de développement.....	99
II.1. Les langages utilisés	99
II.1.1. Langage de programmation (Java).....	99
II.1.2. Le langage PHP : (Personal Home Page).....	100
II.1.3 Le langage HTML : (Hyper Text Markup Language)	101
II.1.4 Le langage JavaScript	102
II.1.5 Le langage de requête MySQL : (Structured Query Language)	102
II.1.6. Le SGBD (MYSQL)	102
II.1.7. Les Servlets	102
II.2. L'environnement de développement et d'implémentation	102
II.2.1. Environnement de développement (Eclipse Galilio)	103
II.2.2. WampServer.....	104
II.2.2.1. Installer Wampserver	105
II.2.2.2. Lancer wampserver	105
II.2.2.3 Utiliser le répertoire www ou des alias	105
II.2.2.4. PhpMyAdmin.....	106
III. Les serveurs.....	107
III.1. Le serveur	107
III.2. Serveur MySQL.....	108
III.3 Fonctionnalités de MySQL.....	109
III.4. Tomcat.....	110
III.5. Le module Tomcat.....	110
IV. Navigateurs	111
V. Présentation graphique de l'application	112
V.1. Page d'accueil de notre plate forme e-learning	112
V.2. Page d'inscription.....	113

SOMMAIRE

V.3. Espace apprenant	115
VI. Conclusion	118
Conclusion générale	119
Annexe A.....	121
Annexe B.....	138

Liste des figures

Figure I.1 – Invocation d’un service Web par un client.....	4
Figure I.2- Architecture d’invocation des services Web.....	5
Figure I.3- Fonctionnement des services web.....	6
Figure I.4 –Chemin emprunté par un message SOAP	11
Figure I.5– Structure d’un message SOAP	12
Figure I.6- Un exemple de message SOAP.....	13
Figure I.7- Place d’un document WDSL dans l’invocation de service.....	17
Figure I.8- Mécanisme d’accès aux services web	24
Figure I.9- Modèle de données de l’annuaire UDDI.....	25
Figure II.1- Naissance de l’EIAO	33
Figure II.2 - Les composantes d’un système d’EIAO.....	34
Figure II.3 Le modèle Overlay.....	36
Figure II.4 : Le modèle bugs.....	36
Figure II.5 : Les concepts : formation à distance, téléformation et e-Learning	41
Figure II.6 - Architecture d’une plate forme.....	51
Figure II.7- Situation des télé-TPs en e-learning	53
Figure II.8- Typologie de situations (TP).....	55
Figure II.9- Architecture type pour les laboratoires distants.....	64
Figure II.10- laboratoire en ligne	67
Figure III.1 : Objectif de notre projet.....	72
Figure III.2 : Les acteurs humains et les principaux cas d’utilisation.....	74
Figure III.3 : Diagramme de contexte de l’application	75
Figure III.4 : Structuration des activités en package.....	76
Figure III.5 : Diagramme de cas d’utilisation pour le paquetage « naviguer dans l’application ».....	77
Figure III.6 : Diagramme de cas d’utilisation pour le paquetage « identification »	78
Figure III.7 : Diagramme de cas d’utilisation pour le paquetage « Suivre une séance d’enseignement ».....	78
Figure III.8 : Diagramme de cas d’utilisation pour le paquetage « Suivre une séance d’enseignement ».....	79
Figure III.9 : Diagramme de cas d’utilisations pour le paquetage «suivi et assistance des apprenants».....	80

Liste des figures

Figure III.10: Diagramme de cas d'utilisations pour le paquetage « participer à la messagerie »	81
Figure III.11 : Diagramme de cas d'utilisation pour le paquetage « utiliser le forum »	81
Figure III.12 : Diagramme de séquence de réalisation du cas d'utilisation « consulter cours »	90
Figure III.13 : Diagramme de séquence de réalisation du cas d'utilisation « faire exercice »	91
Figure III.14 : Diagramme de séquence de réalisation du cas d'utilisation « évaluation »	92
Figure III.15 : Diagramme de séquence de réalisation du cas d'utilisation « Faire TP» ...	93
Figure III.16 : Diagramme de classe générale de cas d'utilisation « identification »	93
Figure III.17 : Diagramme de classe générale de cas d'utilisation « consulter cours »	94
Figure III.18: Diagramme de classe général de cas d'utilisation « Faire TP »	94
Figure III.19 : Diagramme de classe générale.....	95
Figure IV.1 : Architecture applicative générale	98
Figure IV.2 : Indépendance de Java des architectures matérielles.....	100
Figure IV.3 : Lancement d'éclipse GALILIO.....	103
Figure IV.4 : Accueil de l'IDE éclipse GALILIO	104
Figure IV.5 : Interface de WampServer	106
Figure IV.6 : Administration de MySQL à partir de PhpMyAdmin	107
Figure IV.7 : Serveur MYSQL	109
Figure IV.8 : Interface d'Apache Tomcat	111
Figure IV.9: Page d'accueil de notre plate forme	113
Figure IV.10 : Interface d'inscription a la plate-forme	114
Figure IV.11 : Formulaire d'inscription Apprenant	114
Figure IV.12 : Interface espace Apprenant	115
Figure V.13 : Interface Télé-TP	115
Figure IV.14: Résultats retournées après l'initialisation.....	116
Figure IV.15 : Interface qui montre les séquences de déplacement.....	116
Figure IV.16 : Fin de simulation	117
Figure IV.17: Message d'erreur lorsque serveur tomcat est hors service	117
Figure IV.18 : Erreur généré lorsque le protocole soap ne fonctionne pas	118

Liste des figures

Introduction générale

Introduction générale

Au cours de ces dernières années les nouvelles technologies de l'informatique et de la communication (NTIC) ont connu un bouleversement marqué par l'apparition d'internet. Avec l'arrivée de ce dernier, les distances se sont réduites et la communication est devenue plus facile, ses outils de communication permettent aujourd'hui d'organiser des groupes dont les membres peuvent être géographiquement dispersés ; l'utilisation actuelle de ce dernier couvre des activités extrêmes variées art et culture, banque et finance, commerce et enseignement.

L'un des plus grand défis sur internet est celui de l'enseignement on-line ou le e-Learning qui est l'objet de notre travail. En effet ce défi est né que l'internet est le meilleur moyen pour contourner la contrainte de la distance qui a été vue comme un handicap dans l'apprentissage. Aujourd'hui l'enseignement à distance couvre pratiquement toutes les activités pédagogiques qu'un enseignement traditionnel peut offrir, cours, TDs et encore les TP. Comment à ce jour, la complexité de ce travail peut être résolue d'une manière plus efficace avec un gain de temps remarquable et une performance meilleure ?

L'intégration de la télé-TPs en tant que web service dans l'enseignement à distance présente un grand défis pour l'informatique cela à cause de la complexité que présente ce dernier, des divers projets d'études et de conception sont menés sur cette voie afin de bien maîtriser ce domaine.

Donc, l'objectif de notre travail consiste à implémenter un télé-tps en tant que web service dans une plate forme d'enseignement à distance ce télé-TP devra être réutilisable dans d'autres plate forme, nous avons organisé notre mémoire de la manière suivante :

Notre mémoire comporte quatre chapitres, et deux annexes

Le premier chapitre est la présentation de la technologie des services web.

Le deuxième chapitre est la présentation de l'enseignement a distance e-Learning et les travaux pratiques à distance les télé-tps.

Le troisième chapitre consiste à l'analyse et la conception ; ce chapitre s'intéresse à l'étude conceptuelle du système à développer. Dans ce cadre, nous exposons la démarche pour la conception de l'architecture de notre application, ensuite nous détaillons les acteurs impliqués et les fonctionnalités à développer dans cette plateforme.

Le quatrième chapitre est la réalisation ; chapitre relatif à l'application que nous avons développé; dans lequel on retrouvera la description détaillée de notre solution, les outils utilisés pour la développer et quelques interfaces réalisées.

Des annexes ont été ajoutées afin d'apporter de plus amples informations sur certains points décrits dans le rapport.

CHAPITRE I

I. Introduction

Depuis toujours, les systèmes d'information ont cherché à offrir un accès rapide et générale à l'information pertinente. Ceci s'est traduit par un couplage en interne aux technologies de l'intranet et en extranet à celles d'internet, particulièrement dans le cadre du web. Les services Web sont une nouvelle manifestation de cet objectif afin de permettre la réalisation rapide et efficace de systèmes d'information distribués sur internet ou intranet, en intégrant des applications existantes ou nouvelles. Devant leur succès au développement d'internet et au succès du Web, les services Web sont désormais considérés comme la technologie qui aura fait rebondir l'informatique distribuée.

II. Historique

Les services web doivent leur origine à l'informatique distribuée et au développement du Web. Au cours des années 1980, les composants ont vu le jour marquant ainsi la fin de l'ère des applications monolithiques hébergeant l'ensemble des fonctionnalités nécessaires, et une nouvelle génération d'applications a vu le jour. Celle-ci est caractérisée par le fait que le code n'est plus compliqué en une seule unique entité, ce qui caractérisait les applications monolithiques, mais scindé en plusieurs composants compliqués indépendamment avec des interfaces et une sémantique bien définies. Ces composants sont construits pour être réutilisables et pour communiquer via une infrastructure fournie par le système d'exploitation, ouvrant ainsi la porte à leur exploitation de façon externe. Pour cela, il suffit que l'application appelante connaisse les appels de l'infrastructure par composants et l'interface des composants qu'elle veut appeler et utiliser.

Au fil des années, trois modèles de programmation distribuée à base de composants se sont imposés. Il s'agit de DCOM/COM (Distributed Component Object Model/Component Object Model), de Microsoft, JAVA/RMI (Remote Method Invocation), de Sun Microsystems et CORBA (Common Object Request Broker Architecture), de l'OMG (Object Management Group).

Néanmoins, les modèles de composants distribués sont généralement confirmés dans un fonctionnement au sein de réseaux maîtrisés, comme par exemple un intranet d'entreprise. Ils sont complexes à mettre en œuvre, péniblement interopérables et peu compatibles avec les pare-feux. Ajoutons à cela qu'ils s'appuient sur des solutions propres à un éditeur.

L'insuffisance des technologies de programmation distribuée existantes, l'évolution d'internet et le succès du Web, ainsi que l'émergence du langage XML comme norme de structuration de données ont tous contribué à favoriser l'essor des services Web.

III. Présentation des services Web [11]

III.1. Définition d'un service Web

Un service Web est un composant logiciel qui interagit avec d'autres composants logiciels autonomes au moyen de protocoles universels, via le réseau Internet ou n'importe quel réseau local basé sur les protocoles standard du Web. Les applications clientes accèdent aux services Web à travers les protocoles Web (http, SMTP, ou autres) en se servant du protocole SOAP comme http et XML, permet aux applications clientes être implémentées dans n'importe quel langage de programmation, avec n'importe quel modèle de composants, et tourner sur n'importe quel système d'exploitation.

La technologie des services Web, initiée par IBM et Microsoft puis normalisée par le W3C (World Wide Web Consortium), est maintenant acceptée par l'ensemble des acteurs de l'industrie informatique sans exception. C'est surtout ce point qui fait des services Web une technologie révolutionnaire et qui rend aussi populaires.

Les services web sont mis à disposition à partir d'un serveur Web ou serveur d'applications à l'intention des utilisateurs ou d'autres applications interconnectées sur Internet. Ils présentent les deux caractéristiques suivantes :

- Un enregistrement facultatif auprès d'un service de recherche.
- Une interface publique avec laquelle le client invoque le service.

Chaque service Web peut s'enregistrer auprès d'un dépôt central de telle sorte que le client puisse le trouver lors d'une consultation de l'annuaire de services. Une fois le service localisé, le client en récupère la référence. Le client utilise alors le service en invoquant diverses méthodes que ce dernier implémente et publie, ceci au moyen d'une interface publique. Ainsi, chaque service doit publier son interface à l'intention de ses clients :

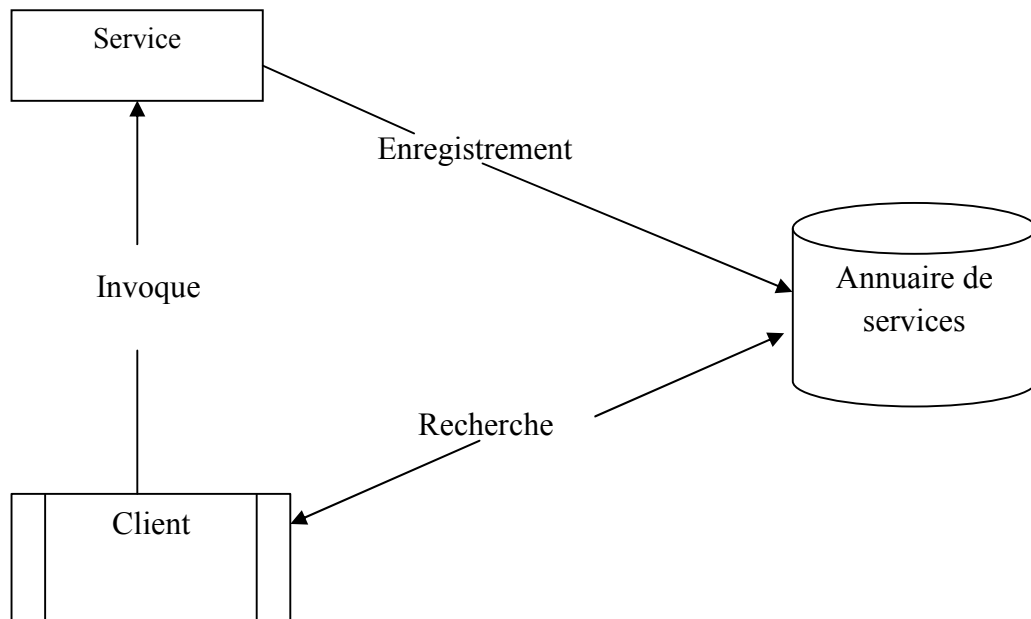


Figure I.1 – Invocation d’un service Web par un client

Le scénario d’invocation d’un service Web par un client sera explicité en détails plus loin dans ce chapitre.

Les services Web favorisent une approche modulaire des applications qui se caractérise par l’apparition de nouveaux modèles économiques. Plutôt que de vendre sa suite de logiciels en bloc, un éditeur commercialisera un composant fonctionnel qui sera invocable à travers le Web par une entreprise cliente. Ainsi, cette dernière assemblera les fonctions dont elle a besoin pour concevoir une application sur mesure. Elle sera à la même d’invoquer dynamiquement les systèmes informatiques des partenaires à travers le web pour fabriquer une offre personnalisée afin de répondre aux besoins de ses clients. Les services Web ont donc pour but d’utiliser l’Internet comme une infrastructure de communication pour le développement d’applications réparties à grande échelle.

Fonctionnellement, les services Web n’apportent rien de nouveau par rapport aux middlewares conçus pour appeler des services distants, ou par rapport à l’EDI qui décrit comment échanger des documents d’affaire. Toutefois, la façon par laquelle ils réalisent ces fonctions est entièrement novatrice, découplant ainsi le service lui-même de son implémentation.

III.2. Invocation des services Web [3]

D'après la figure I.2, l'invocation d'un service Web par une application s'appuie sur un modèle d'interactions dont les composants assurent trois rôles : le Service Provider (fournisseur de services), le Service Registry (annuaire de services) et le Service Requester (demandeur de services).

- Service Provider : C'est le fournisseur de services, dont il en est aussi le propriétaire. Il héberge un module logiciel, implémentant un ou plusieurs services Web, accessible par un client Web via le protocole SOAP.
- Service Registry : Annuaire contenant les descriptions des services Web publiés par les fournisseurs de services. Il est géré sur un serveur niveau application, entreprise ou mondial.
- Service Requester : Demandeur du service, qui est un responsable d'application cliente se liant à un service et invoquant ses fonctions par messages XML échangés via le protocole SOAP.

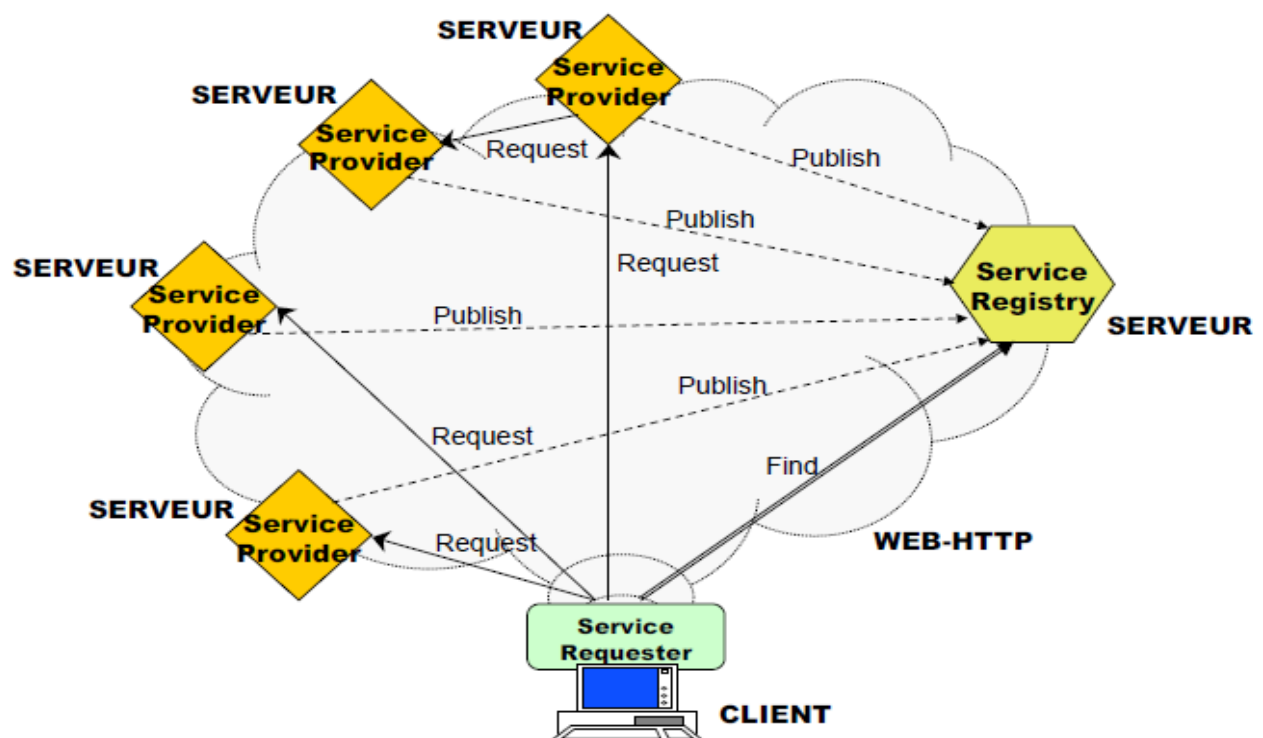


Figure I.2- Architecture d'invocation des services Web

La figure I.2 nous montre que les services Web peuvent être vus comme des composants logiciels, décrits au niveau de l'annuaire des services. Les demandeurs de services découvrent les caractéristiques fonctionnelles et techniques des services offerts dans l'annuaire. En suite, ils y entrent en liaison en invoquant les différents services depuis leur code applicatif pour réaliser leurs applications dont les services sont les briques de base. L'architecture permet aussi qu'un service Web puisse invoquer un autre service Web.

III.3.Fonctionnement global des services web

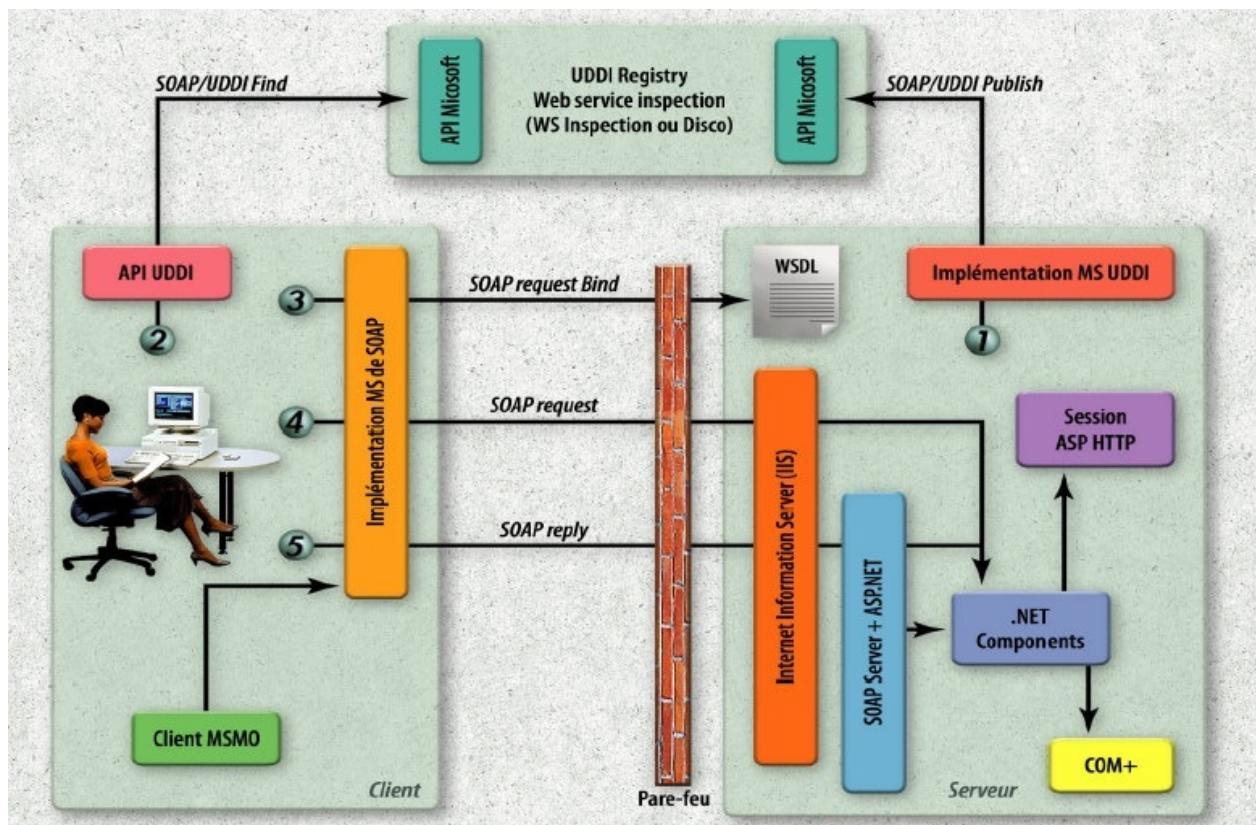


Figure I.3- Fonctionnement des services web

Dans le scénario de fonctionnement des services Web, un fournisseur de services (Service Provider) héberge un module logiciel (plate forme d'exploitation) implémentant un ou plusieurs services Web accessibles via le réseau. Ce module implique des interfaces de connexion permettant : d'être à l'écoute d'une application cliente (généralement un navigateur Web), interrogation dans le cas de la récursivité (un service Web peut appeler un autre service Web) et la publication des fichiers WSDL (enregistrement de la description du service Web sur

l'annuaire UDDI). L'annuaire des services est consultable sous plusieurs facettes (pages jaunes, pages vertes, pages blanches), cela dépend des critères transmis dans la requête de recherche (Find) émise du côté de l'application cliente. Ceux-ci peuvent englober les caractéristiques fonctionnelles, caractéristiques techniques des services Web, taxinomie utilisée pour classifier l'entreprise qui l'a publiée selon son activité, nom ou même la description de l'entreprise qui l'a publiée. Suite à cette requête un fichier WSDL (réponse) sera renvoyé à l'application cliente (l'annuaire UDDI suit le modèle requête réponse), qui le reçoit via son interface de connexion WSDL, ce fichier contient des informations sur le service Web (interfaces des services, emplacement sur le Web) et l'entreprise qui l'a publié. A ce niveau l'application cliente pourra se connecter physiquement à un emplacement bien précis sur le Web pour interroger le service Web qu'elle vient de découvrir via un protocole de transport (HTTP). Pour ce faire l'application cliente fait intervenir un processus de sérialisation/désérialisation (Encodeur SOAP) pour générer les messages SOAP, encapsulé dans une requête http. Le service Web réagit à son tour, en envoyant une réponse SOAP via le protocole http.

IV. Avantages des web services

De nos jours, la technologie des web services est populaire et couramment utilisée car elle offre des avantages intéressants pour les utilisateurs des systèmes distribués:

- Les web services réduisent le temps de mise en marche des services offerts par les diverses entreprises.
- Les web services permettent à des programmes écrits en des langages différents et sur des plates-formes différentes de communiquer entre eux par le biais de certaines normes. En d'autres termes, les web services permettent une meilleure interopérabilité entre les logiciels.
- Les web services utilisent des normes et protocoles ouverts.
- Grâce au protocole HTTP, les web services peuvent fonctionner malgré les pare-feu sans pour autant nécessiter des changements sur les critères de filtrage.

- Les protocoles et les formats de données sont offerts, le plus possible, en format texte pour que la compréhension du fonctionnement des échanges soit plus intuitive.
- Grâce aux web services, les coûts sont réduits par l'automatisation interne et externe des processus commerciaux.

V. Les technologies des Web Services [4]

XML, SOAP (*Simple Object Access Protocol*), WSDL (*Web Services Description Language*) et UDDI (*Universal Description, Discovery and Integration*) sont les technologies dominantes des *Web Services*. Une pléthore d'autres technologies viendront, au fil du temps, garnir l'architecture des *Web Services*. Sans entrer dans tous les détails techniques, nous vous exposons ici les grandes lignes de chacune de ces technologies et de celles, qui à court et moyen terme, risquent de se positionner à la suite.

Fondamentalement, SOAP, WSDL et UDDI sont des technologies issues de l'intérêt parmi des membres de la communauté Internet à développer un mécanisme pour échanger des documents XML sur le Web entre systèmes d'information.

V.1.XML et les services Web [7]

Le but de l'informatique distribuée était de permettre aux systèmes d'informations de faire apparaître leurs rouages (bases de données, services distribués, applications, etc.) comme des composantes fonctionnant avec une API commune. Cependant, les systèmes distribués à base de composants COBRA, JAVA/RMI et COM/DCOM ne répondaient que partiellement à ce but, du fait que leur implémentation utilisait des interfaces spécifiques, ce qui conduisait à des solutions propriétaires qui réduisaient la production coopérative des documents. L'arrivée des services Web XML qui proposaient un modèle objet distribué proche de celui de CORBA et DCOM sur le plan technique, a fait des systèmes d'informations de simples assemblages de composants de services XML indépendants capables de produire et de consommer des documents XML et qui communiquent entre eux à travers un canal de communication. Les services Web XML reposent sur une architecture objets distribués XML/HTTP, et ne nécessitent pas de disposer d'une API commune, ils étendent les modèles JAVA, COM, CORBA à Internet et assurent l'extension de l'interopérabilité des composants existants. Le déploiement des services Web (SOAP, WSDL, UDDI) est assuré par le langage XML, qui est utilisé comme un format d'importation et d'exportation d'informations et dans les appels RPC pour l'invocation des procédures à distance.

L'apport du langage XML et son impact sur l'informatique distribuée en particulier les services Web, sont résumés dans les points suivants :

- XML qui est un langage simple, définit un standard de description de données. Les messages XML sont auto descriptifs, ils permettent le traitement des données sans en connaître le format. De tels messages sont indépendants des systèmes d'exploitation, des plates-formes, des langages de programmation et des formats d'affichage. Ceci facilite l'échange de données entre les différents partenaires.
- XML et les technologies qu'y sont liées (SOAP, WSDL, UDDI) forment un Framework des services Web. Une association qui a révolutionné la manière dont les applications communiquent entre elles. Ce langage a fait des services Web des applications à accès universel à partir de n'importe quel ordinateur ou appareil, indépendamment de sa technologie propriétaire de base.
- Les services Web XML ont fait d'Internet un réseau orienté service, sur lequel à peu près tous les services ou applications dont nous pourrions avoir besoin s'y trouvent.
- XML suivi par tous les éditeurs (Microsoft, Sun, Oracle, etc.) ce qui rend disponible les outils nécessaires pour le développement des services Web, tel est le cas pour les plates-formes J2EE de Sun Microsystems et .Net de Microsoft.

V.2. Le protocole SOAP (*Simple Object Access Protocol*)

SOAP est un protocole de la famille XML servant à l'échange d'informations dans un environnement distribué et décentralisé. Il est considéré comme la technologie la plus importante des *Web Services*. Le standard SOAP a été proposé au W3C par Microsoft, IBM, Lotus, DevelopMentor et Userland.

Le standard SOAP définit trois éléments composant un message : l'enveloppe, l'entête du message et le corps du message. L'enveloppe définit le cadre pour décrire ce qui est dans le message et comment le traiter. Les règles d'encodages sont placées dans l'en-tête et servent à exprimer et définir le mécanisme de représentation des données. Le corps du message permet de transmettre les requêtes et les réponses entre les systèmes.

Plus de 80 implantations de la spécification SOAP v1.1 ont été développés jusqu'à présent. Cela démontre déjà la simplicité, la popularité et la justesse de l'approche qu'il offre pour transporter des données sur le Web.

V.2.1.Définition du protocole SOAP [8]

SOAP est standard du consortium W3C, définissant un protocole qui assure des appels de procédures à distance (RPC) s'appuyant principalement sur le protocole HTTP et sur XML, mais aussi sur d'autres protocoles de communication (SMTP, POP, etc.).

L'objectif du protocole SOAP est de permettre l'invocation de services Web qui offrent des services sur des ports de connexion. Pour cela, il offre à l'application cliente les outils nécessaires pour invoquer des services distants en lui donnant l'illusion qu'ils sont locaux. Comme nous l'avons signalé SOAP est un protocole indépendant de toute plate-forme et de tout langage de programmation : il réalise le codage des appels en XML, qui est un codage universel. Sur ce plan, il est beaucoup plus général par rapport à ce qu'offraient CORBA (IIOP/CDR), JAVA (RMI/JRMP), DCOM (DCE/ENDR) et RDA (SQL/XDR), tous spécifiques comme l'indiquent les sigles associés.

En fait, SOAP peut être vu comme un simple codage universel des appels et retours de procédure de type RPC. Il apporte cependant, un peu plus de fonctions.

V.2.2.Caractéristiques du protocole SOAP

- SOAP permet une normalisation des échanges de données. Les données sont encodées en XML et échangées par des appels de procédures à distance en utilisant HTTP/SMTP/POP comme protocole de communication.
- Simple, extensible et permet le diagnostic des erreurs.
- Message unidirectionnel (Requête -> Réponse).
- Fonctionne de manière synchrone et asynchrone.
- Indépendant de la plate-forme et du langage.
- N'est pas perturbé par un pare-feu.

V.2.3.Modèle SOAP d'échange de messages

La spécification SOAP définit un modèle d'échange de messages. Elle repose sur trois concepts de base : les messages sont des documents XML ; ils voyagent d'un émetteur vers un récepteur ; les récepteurs intermédiaires peuvent former une chaîne.

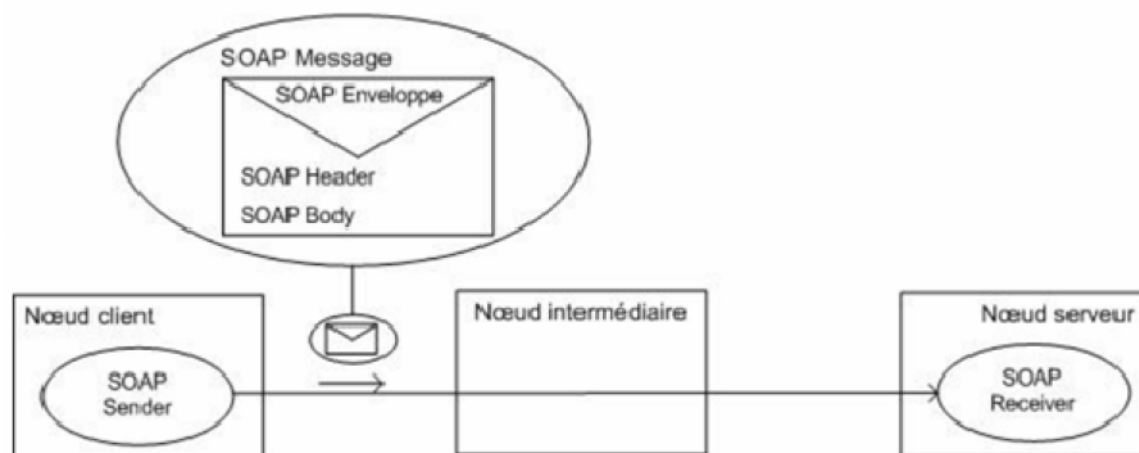


Figure I.4 –Chemin emprunté par un message SOAP

V.2.3.1.Document XML comme message

Les messages SOAP sont en langage XML, ce qui confère à SOAP plusieurs avantages par rapport aux autres protocoles de messagerie. Les messages XML peuvent être rédigés et lus par un développeur au moyen d'un éditeur de texte ce qui simplifie considérablement le processus de débogage par rapport à un autre protocole binaire complexe.

V.2.3.2.Emetteurs et récepteurs

L'échange d'un message SOAP implique deux parties : un émetteur et un récepteur. Le message transite de l'émetteur vers le récepteur. Cette opération qui représente la plus petite unité de travail, s'avère insuffisante dans de nombreux cas. Il est plus souvent nécessaire d'échanger des messages par paires de requête/réponse. Cette méthode est celle utilisée par le protocole SOAP avec le transport HTTP et/ou la convention RPC. Les échanges de messages en sens unique sont également possibles. On peut parvenir à composer des échanges plus complexes en se basant sur un échange de message à sens unique entre un émetteur et un récepteur, ce qui nous donne la possibilité de construire des chaînes de messages.

V.2.3.3.Chaines de messages

Les messages SOAP ne suivent pas nécessairement un modèle client-serveur classique. Ils pourraient être échangés de cette manière ou traités par une chaîne d'entités logiques. Ce concept d'entité logique qui effectue le traitement d'un message SOAP est appelé **point final**. Les points

finaux sont des récepteurs de messages SOAP. Chaque point final est responsable de l'examen d'un message et de l'extraction de la partie qui lui a été adressée pour la traiter.

Les points finaux peuvent fonctionner à la fois comme émetteur et comme récepteur. Cette faculté permet la mise en place d'un système de traitement qui peut éventuellement être effectué à chaque étape. Les points finaux qui fonctionnent à la fois comme émetteurs et comme récepteurs et transmettent à un autre point les messages qu'ils ont reçus sont appelés **intermédiaires**. Les intermédiaires et les points finaux sont identifiés par des URI (Universal Resource Identifier).

V.2.4. Structure des messages SOAP

La structure de message SOAP est représentée ci-dessous

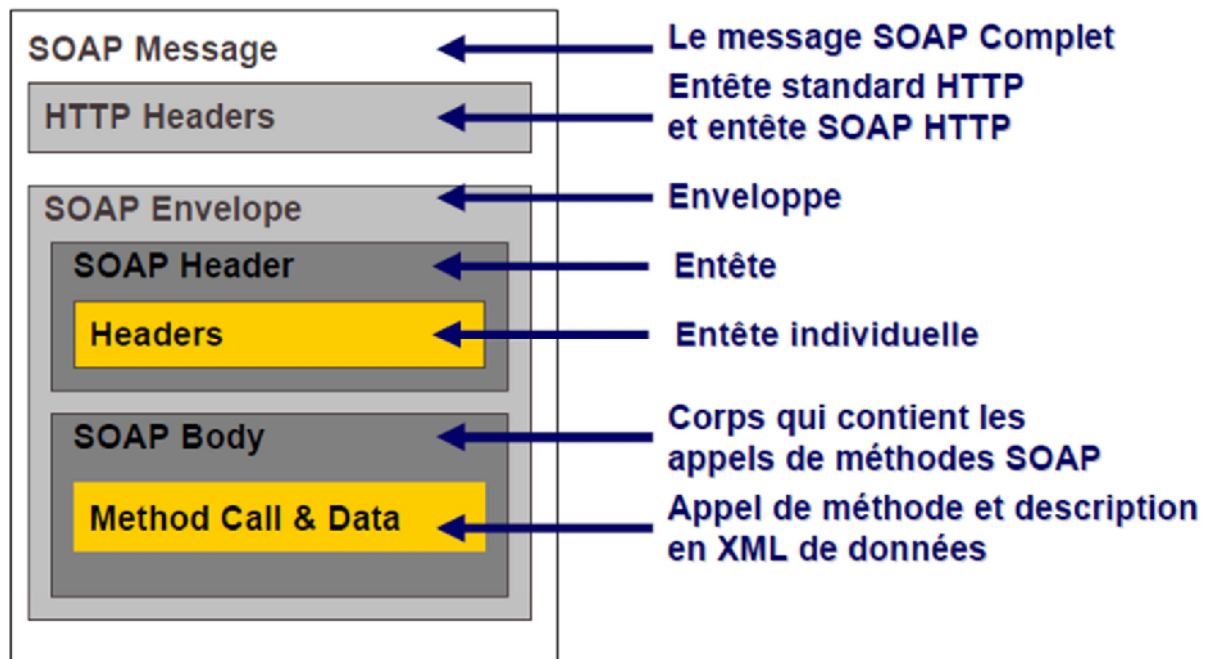


Figure I.5– Structure d'un message SOAP

Un message SOAP comprend une charge utile c'est-à-dire les informations spécifiques à l'application.

Voici un exemple de message SOAP :

```
<soap:Envelope xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
  Soap:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>
```

```
<soap:Header>
<h:from xmlns:h=http://www.wrox.com/header>SoapGuy@wrox.com</h:from>
</soap:Header>
<soap:Body>
<w:GetSecretIdentity xmlns:w=http://www.wrox.com/heroes/>
<w:codename>XSLT-Man</w:codename>
</w:GetSecretIdentity>
</soap:Body>
</soap:Envelope>
```

Figure I.6- Un exemple de message SOAP

Les messages SOAP dépendent fortement des espaces de noms XML. Tous les éléments présentés ici ont un préfixe d'espace de noms. En effet pour qu'un message SOAP puisse transporter n'importe quelle charge utile XML supplémentaire il faut déterminer la portée de chacun de ses éléments de manière à éviter tout conflit sur les noms de ces éléments.

L'ensemble des éléments du message qui sont associés à l'espace de noms soap constitue les éléments standards d'un message SOAP, de même que les attributs. Les autres éléments peuvent être reliés aux extensions ou à la charge utile du message. Notre exemple contient trois éléments SOAP standards : **Envelope**, **Header** et **Body**. Il existe un quatrième élément standard qui n'apparaît pas dans notre exemple, l'élément **Fault**.

V.2.5.Le transport

Le transport est la méthode qui permet d'acheminer un message SOAP d'un émetteur à un récepteur. La spécification SOAP a été développée de manière à permettre de séparer entre la définition du message et son transport.

V.2.5.1.SOAP via HTTP

HTTP constitue un excellent transport pour SOAP en raison de sa popularité. La combinaison de HTTP, protocole de transport pour le Web, et de SOAP, meilleur candidat au titre format de messagerie standard, permet d'obtenir un outil très puissant.

Il existe deux règles à respecter pour l'utilisation de HTTP comme transport de messages SOAP :

- Le mécanisme d'envoi d'un message SOAP via HTTP est la méthode **HTTP POST** standard. Un **HTTP POST** envoie un bloc de données à un URI donné sur le serveur Web. Dans le cas de messages SOAP, ce bloc de données est le message SOAP lui-même.
- Le message étant en XML, l'en-tête **Content-Type** du **HTTP POST** doit être **text/xml**. La réponse au message s'il y'en a une, est envoyée dans la réponse HTTP.

Voici un exemple de message SOAP transporté via HTTP :

- 1- POST/endpoint.asp HTTP/1.1
- 2- Content-Type text/XML
- 3- Content-Lenght : ###
- 4- SOAPAction : "urn :wroxheroes"
- 5- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
 Soap:encodingStyle="http://shemas.xmlsoap.org/soap/encoding/">
 <soap:Header>
 <h:from xmlns:h="http://www.wrox.com/header">
 SoapGuy@wrox.com
 </h:from>
 </soap:Header>
 <soap:Body>
 <w:GetSecretIdentity xmlns:w="http://www.wrox.com/heroes/">
 <CodeName>XSLT-Man</CodeName>
 </w:GetSecretIdentity>
 </soap:Body>
 </soap:envelope>

La première ligne contient la méthode **HTTP POST**, l'URI indiquant la position du point final ainsi que la version du protocole HTTP utilisée dans cette requête.

Les trois lignes suivantes sont des en-têtes HTTP. **Content-Type** indique le type MIME du contenu du POST : tous les messages SOAP doivent utiliser **text/xml**. **Content-Lenght** définit la taille en octets du contenu. Le dernier en-tête **SOAPAction** est spécifique à SOAP.

V.2.5.2.En-tête SOAPAction:

L'en-tête SOAPAction, indique aux serveurs qu'un HTTP POST donné contient un message SOAP et la valeur de cet en-tête contient un URI informant de l'objet de ce message, c'est-à-dire la destination du message. Cela permet aux pare-feux et autres serveurs d'exécuter un traitement conditionnel, basé sur la présence de l'en-tête SOAPAction.

La version 1.1 imposait la présence de l'en-tête SOAPAction, même si cet en-tête peut-être vide. Si tel est le cas, l'objectif du message est déduit de l'URI de la cible du POST. Dans les versions récentes du protocole HTTP, cet en-tête est retiré.

V.2.5.3.Codes de statut

HTTP renvoie des informations relatives au statut sous forme de code de statut. Ces codes sont des nombres entiers, divisés en classes de 100.

- Tout code compris entre 200 et 299 indique une opération réussie
- Le code 500 représente une erreur interne serveur

Dans le cas d'une opération réussie, la première ligne de la réponse renvoyée sera :

HTTP1.1/ 200 ok

V.2.6.SOAP et RPC

La convention RPC peut être décrite comme une technique de sérialisation des appels de procédures à distance et des réponses comme messages SOAP.

Comme son partenaire HTTP, RPC SOAP utilise un modèle requête-réponse pour l'échange de messages. L'exécution d'un appel de procédure à distance avec le protocole SOAP nécessite uniquement la construction d'un message SOAP. Le message de requête SOAP envoyé au point final représente l'appel, et le message de réponse SOAP représente le résultat de cet appel.

V.2.6.1.L'appel

L'exécution d'un appel de procédures à distance avec le protocole SOAP nécessite seulement la construction d'un message SOAP. La charge utile de ce message contient l'appel de méthode sérialisé, sous forme d'une structure dont le nom de l'élément correspond au nom de la méthode, et les éléments enfants de cette structure sont les paramètres entrants de cette méthode.

Considérons une méthode baptisée `GetNumeroMois` qui retourne le numéro du mois qu'elle reçoit à l'argument. L'espace de noms de cette méthode est `http://tempuri.org/`, et sa signature est :

```
Int GetNumeroMois(string mois) ;
```

L'appel de méthode sérialisé demandant le numéro du mois au moyen de la chaîne « Mars » ressemblerait à ce qui suit :

```
<q :GetNumeroMois xmlns :q="http://tempuri.org/">  
<q :mois xsi :type="xsd :string">Mars</q :mois>  
</q :GetNumeroMois>
```

V.2.6.2.Réponse

Les résultats de l'appel sont renvoyés dans le message de réponse SOAP. La charge utile du message de réponse contient également la valeur résultant de la méthode.

Le résultat sérialisé de l'appel de la méthode précédente ressemblerait à ce qui suit :

```
<q :GetNumeroMoisResponse xmlns :q="http://tempuri.org/">  
  <q :moisNumero xsi :type="xsd :int">3</q :mois>  
</q :GetNumeroMoisResponse>
```

Notons qu'aucune directive n'existe concernant le nom de la structure de réponse de la méthode, mais l'usage veut que l'on utilise le nom de l'appel de la méthode préfixé par **Reponse**

V.2.6.3.RPC et HTTP

Pour exécuter un appel de procédures à distance, il faut un moyen de déplacement du message jusqu'à la position à distance. C'est dans cette situation, lorsqu'on combine RPC et HTTP pour appeler les services Web, que le protocole SOAP montre sa puissance.

Supposons que nous devons appeler une procédure à distance sur un serveur afin de retourner le code postal d'une commune. Notre hypothétique service Web serait à l'adresse `http://www.retournecodepostal.com/call.asp`. La méthode pourrait être invoquée à l'URL `www.retournecodepostal.com/call.asp`. Elle serait associée à l'URI de l'espace de noms

`http://www.retournecodepostal.com/methode`, et l'en-tête `SOAPAction` pour cette méthode serait `urn :retournecodepostal`.

La signature de cette méthode : `string RetourneCodePostal (string willaya, string commune)` ;

V.3. Le langage WSDL (Web Service Description language)

V.3.1 Présentation de WSDL :

WSDL (Web Service Description language) est un format XML permettant de décrire un service Web, et ce, en définissant les opérations exposées par ce service et le format qu'un client doit employer pour le solliciter. Il met en place un contrat entre un client et le service, pour rendre la communication efficace entre les deux parties. Le client doit solliciter le service en lui envoyant une requête SOAP correctement formatée.

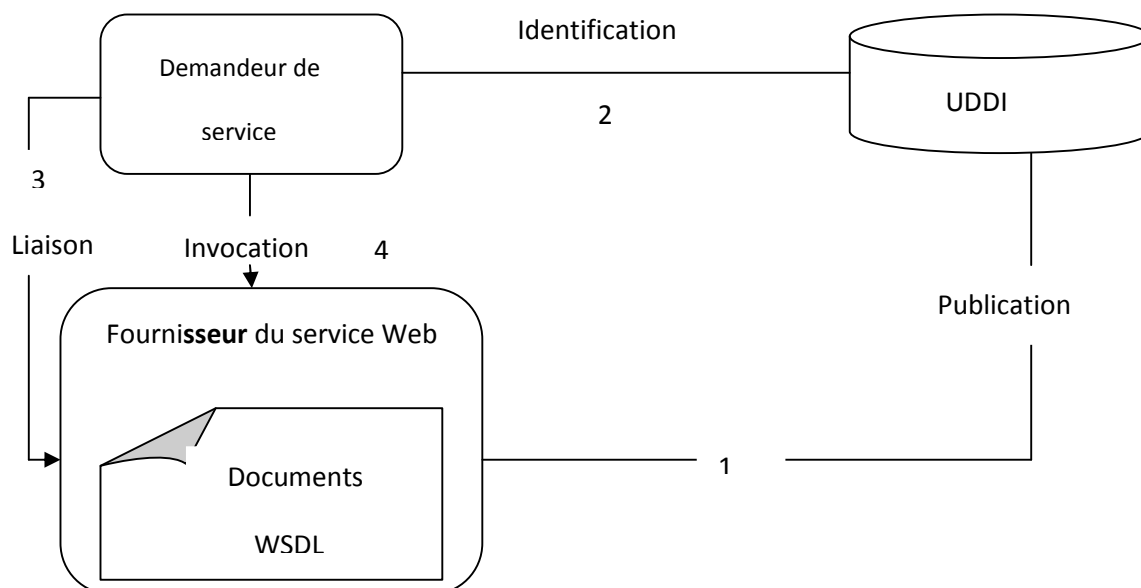


Figure I.7- Place d'un document WSDL dans l'invocation de service

WSDL est né d'un effort commun entre IBM, Microsoft et Ariba. Il repose sur Http, MIME et SOAP comme mécanisme d'invocation d'objets distants, et sur la spécification des

schémas XML comme système de type, tout en permettant l'utilisation d'autres langages de définition de type.

V.3.2. Pourquoi WSDL ? [2]

Pour correspondre à leur image de composants disponibles via Internet et utilisables sur n'importe quelle plate-forme et à partir de n'importe quel langage, les services Web doivent être auto-descriptifs. C'est là que le langage WSDL entre en action. En effet, il permet de décrire ce qu'un service fait et comment les clients peuvent l'exploiter. Ainsi, tout service décrivant ses interfaces avec WSDL peut être invoqué, par voie de programmation, par n'importe quel client sans que ce dernier ne soit contraint de connaître les détails d'implémentation du service, ni la plate-forme ou le système d'exploitation sur le quel il tourne.

Le modèle d'échange supporté par WSDL est inspiré de la navigation maritime : des messages contenus dans des enveloppes sont échangés de port en port. Pour décrire les services, WSDL introduit, en plus de la notion de service et de définitions de types de données avec les schémas, les notions port, liaison et messages.

V.3.3. Syntaxe WSDL

WSDL fournit une grammaire de description des services sous forme d'un ensemble de points finaux qui échangent des messages. Un document WSDL se compose de définitions présentant un **service** comme un ensemble de points finaux de réseau, ou ports. Chaque port est associé à une liaison spécifique. C'est cette liaison qui détermine comment un ensemble abstrait d'opérations et de messages est lié à un protocole particulier. Une liaison établit une correspondance entre un protocole donné et un **type de port**. Un type de port se compose d'un ensemble **d'opérations**, qui représentent un ensemble abstrait de « choses » que peut faire le service. Chaque opération est constituée d'un ensemble de messages abstraits, qui représentent les données communiquées au cours de l'opération. Chaque message contient une ou plusieurs données définies par des **types**.

La spécification d'un document WSDL offre un jeu de six définitions. La racine du document est <définitions>.

Les six éléments imbriqués sont :

- <types>
- <message>
- <portType>
- <binding>
- <port>
- <service>

V.3.4.Exemple d'un document WSDL

Considérent un service boursier, nommé **Bourse**, qui expose une méthode baptisée **QuelCours** qui délivre le cours d'une action. Cette méthode accepte un seul paramètre, **CodeAction**, qui peut être soit un entier, soit une chaîne de caractère, et retourne un nombre réel (float) représentant le cours de l'action dont le code est passé en paramètre. La communication avec le service **Bourse** se fait en utilisant le protocole SOAP via http, avec des requêtes de style **document**.

Voici la description du service **Bourse** :

```
<?xml version="1.0" encoding="utf-8">
```

Le premier du document WSDL est <definitions>, dans lequel nous définissons le nom du service ainsi que les différents espaces de noms utilisés.

```
<definitions name="Bourse" targetNamespace="http://exemple.com/bourse.wsdl"
    xmlns:wns="http://exemple.com/bourse.wsdl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsd1="http://exemple.com/bourse.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap1="http://schemas.xmlsoap.org/wsdl">
```

Ici, nous définissons l'espace de noms WSDL comme espace de noms par défaut, l'espace de noms wns où se trouve le document WSDL (c'est aussi le nom d'espace cible), l'espace des types des schémas xsd, l'espace de définitions de SOAP, l'espace de définitions http, ainsi que l'espace de noms de notre propre schéma.

En suite, il faut définir les types de données utilisées dans un message. Comme nous utilisons ici un type de données spécifiques pour le paramètre **CodeAction**, qui peut être soit un entier, soit une chaîne de caractères, l'élément `<types>` est nécessaire.

```
<types>
<schema targetNamespace="http://exemple.com/bourse.xsd">
<element name="CodeAction">
<complexType>
<choice>
<element name="symbole" type="xsd:string"/>
<element name="code" type="xsd:int"/>
</choice>
</complexType>
</element>
</schema>
</types>
```

Les types que nous avons définis vont être maintenant associés aux messages (éléments `<message>`). Chaque élément `<message>` contient 0 à N éléments `<parts>`, et chacun de ces derniers correspond à un paramètre d'appel ou de retour, et doit avoir le même nom et le même type que le paramètre de la méthode correspondante.

```
<message name="QuelCoursInput">
<part name="CodeAction" element="xsd1 : CodeAction" />
</message >
<message name="QuelCoursOutput" >
<part name="valeur" element="xsd :float" />
</Message>
```

Les messages sont définis indépendamment des opérations. Il nous faut maintenant définir les opérations à l'intérieur d'un élément `<opération>` pour préciser quel message d'entrée et de sortie de chaque opération. Les messages doivent être définis complètement avec leur nom

d'espace (le préfixe wns : est utilisé pour le nom d'espace du document WSDL) . L'ensemble des opérations exposées par le service sur un port est défini au sein de l'élément < portType>.

Ceci nous donne :

```
<portType name ="BoursePortType">
  <operation name ="QuelCours">
    <input message="wns : QuelCoursInput" />
    <output message ="wns : QuelCoursOutput" />
  </operation>
</portType>
```

Jusqu'ici , nous n'avons fourni qu'une description abstraite pour le service Bourse. Pour que ce dernier puisse être invoqué depuis SOAP , il faut préciser la liaison des éléments WSDL en SOAP. Il s'agit donc de passer du type abstrait à la représentation concrète pour la communication .C'est là que l'élément <binding> intervient :

```
<binding name ="SOAPBindingBourse" type ="wns BoursePortType ">
...
</binding>
```

A l'intérieur de l'élément <binding>, l'élément <soap : binding > permet de préciser le style de la requête (**rpc** ou **document** sont possible) ainsi que le protocole de transport utilisé. Dans notre exemple, il s'agit du style document et du protocole HTTP.

Nous aurons donc :

```
<soap : binding style ="document" transport ="http://shemas.xmlsoap.org/soap/http"/>
```

Ensuite, il faut spécifier l'action SOAP pour chaque opération. Cette action représente l'objectif de chaque opération. Notre exemple n'en comporte qu'une seule. Cela donne :

```
<operation name =QuelCours>
  <Soap :operationSOAPAction ="http://exemple.com /QuelCours"/>
  ...
</operation>
```

Finalement, il faut spécifier le style d'encodage pour les messages d'appel et de retour de l'opération. Ceci s'effectue à l'intérieure des éléments `<input>` et `<output>` pour l'opération Quel Cours. Nous aurons donc :

```
<input>
<soap:body use="literal" namespace="http://exemple.com/bourse.xsd"
    encoding="http://shemas.xmlsoap.org/soap/encordin/" />
</input>
<output>
</output>
<soap:body use="literal" namespace="http://exemple.com/bourse.xsd"
    encoding="http://shemas.xmlsoap.org/soap/encoding/" />
</output>
```

Enfin, nous définissons le service. Chaque service est défini dans un élément `<service>` dont le premier attribut représente le nom du service. A l'intérieur de cet élément nous définissons les différents ports par lesquels le service est accessible. Un port est caractérisé par un nom et une liaison (`<binding>`). Une adresse http du service lui est associée. Nous aurons donc :

```
<service name= ServiceBourse>
<documentation>un service d'accès aux cours de la bourse </documentation>
<port name=SOAPBoursePort binding =wsn :SOAPBindingBourse>
<soap:address location =http://exemple.com/bourse/>
</port>
</service>
```

Soulignons qu'un service Web peut être accédé directement par HTTP .il sera exposé via une adresse HTTP : par exemple : sur un port défini comme suit :

```
<port name=HTTPBourse = http //www.exemple.com /CoursBourse.asp/>
</port>
```

Bien sur, une liaison http doit être définie au préalable avant qu'elle soit associée au port défini ci-dessus.

En résumé nous avons détaillé comment construire un document WSDL pour la description d'un service Web. Dans la pratique, les développeurs utilisent des outils de développement générant

Le WSDL plutôt que de rédiger à la main une description de service. Ceci parce que dans la plupart des cas, un service Web publie plusieurs méthodes et, plus le nombre méthodes est important, plus la description WSDL devient importante, donc la rédiger manuellement sera une opération très complexe.

V.4.L'Annuaire UDDI [3]

V.4.1.Définition

L'annuaire UDDI permet aux utilisateurs de faire la publication des applications (des web services). En effet, UDDI peut contenir des informations sur les fournisseurs et les services qu'il publie.

L'inscription d'un fournisseur de services à l'annuaire UDDI lui permet de se présenter et présenter ses services, l'adoption de cet annuaire par les fournisseurs permet l'accélération des échanges surtout les échanges commerciaux de type B2B. L'enregistrement des web services dans un annuaire UDDI, s'effectue auprès d'un opérateur en accédant au site web de ce dernier à partir d'un navigateur ou d'un outil intégré dans l'environnement de développement. Des recherches précises peuvent s'effectuer dans l'annuaire par catégorie de fournisseurs en utilisant des standards de taxinomie et d'identification de fournisseurs.

V.4.2.Pages de l'annuaire

Les données retrouvées dans l'UDDI sont divisées en trois catégories :

Pages blanches : le référentiel comporte des informations sur les fournisseurs de services tels que le nom et les coordonnées du fournisseur.

Pages jaunes : le référentiel comporte des critères de catégorisation de services, les critères de catégorisation s'appuient sur des standards de classification de fournisseurs, les services sont décrits par des documents au format WSDL. Un fournisseur peut disposer de plusieurs entrées dans l'annuaire pour l'ensemble des différents services et produits qu'il publie.

Pages vertes : le référentiel comporte des informations techniques (WSDL) détaillées

sur les services fournis telles que les informations sur les processus métier, les descriptions de services et les informations de liaison sur les services.

Les annuaires offrent des mécanismes standards de classifications. Ils peuvent être publics ou privés

Les annuaires publics sont hébergés par des sociétés comme IBM ou Microsoft.

Les annuaires privés peuvent être hébergés par une société quelconque sur un réseau privé ou sur internet.

NB : Les annuaires publics sont moins développés que les annuaires privés en terme de sécurité.

V.4.3. Mécanismes d'accès aux services fournis par l'UDDI

La communication (requêtes / réponses) avec un annuaire UDDI repose sur le protocole de transport SOAP

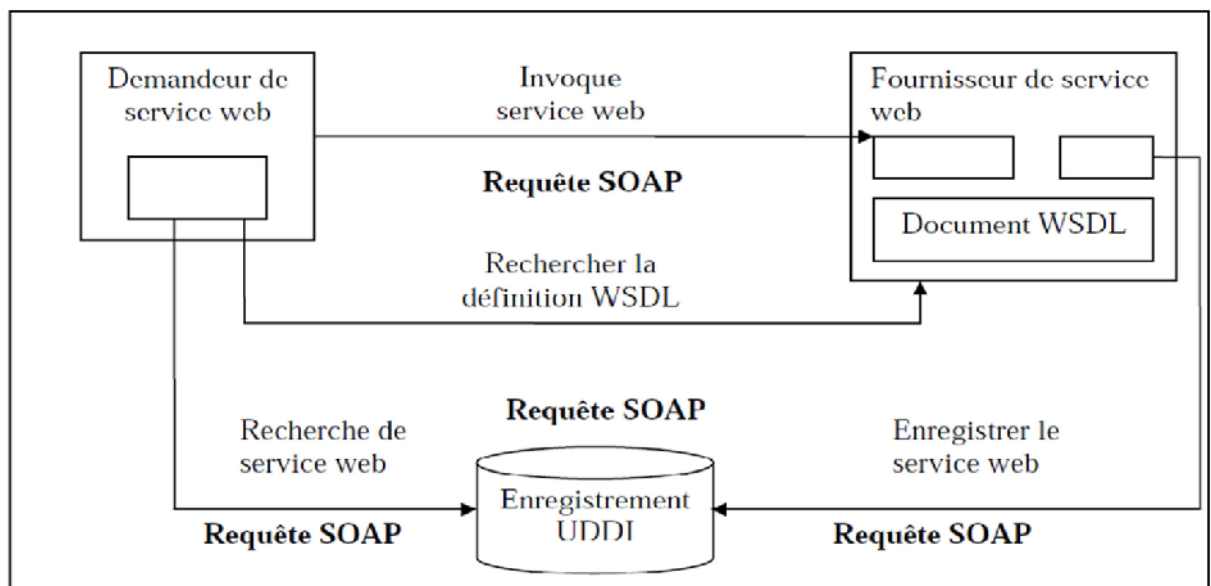


Figure I.8- Mécanisme d'accès aux services web

Un enregistrement UDDI, a deux types de clients :

Les fournisseurs de services.

Les utilisateurs de ces services.

UDDI comporte des pages qui fournissent des informations sur les fournisseurs (nom, coordonnées,...etc.) et des pages qui comportent la description, au format WSDL, des web services. Un fournisseur peut disposer de plusieurs entrées dans l'annuaire pour l'ensemble de services qu'il propose, et des pages qui disposent d'informations techniques détaillées sur les produits proposés.

V.4.4. Le modèle de données dans UDDI

Le modèle de données UDDI est défini sous forme de schéma W3C XML Schéma, ce schéma XML comporte cinq structures de données principales ;

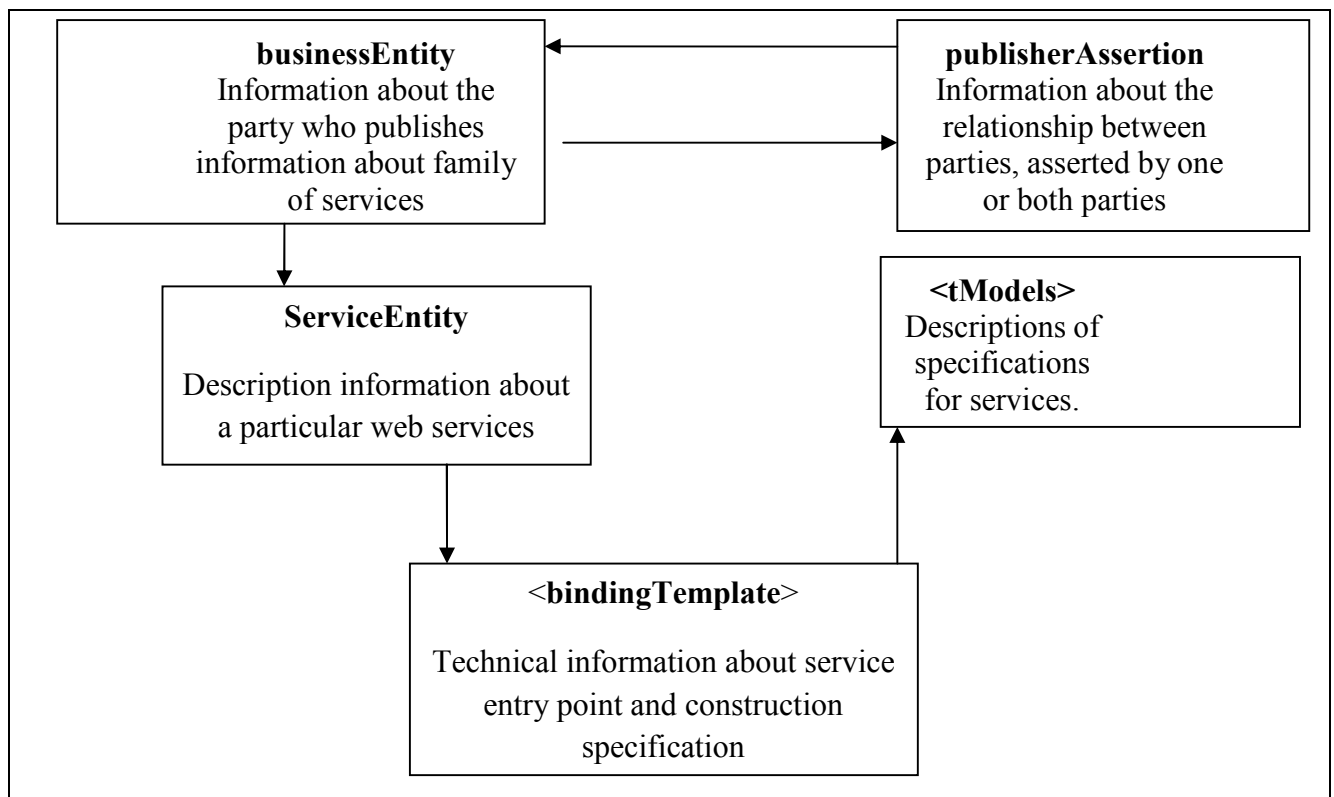


Figure I.9- Modèle de données de l'annuaire UDDI

BusinessEntity : chaque businessEntity est identifiée par une «businessKey». Les

«businessEntities» sont en quelque sorte les pages blanches d'un annuaire UDDI, elles

comportent des informations sur les fournisseurs de services ayant publié des services dans l'annuaire. On y trouvera notamment le nom de l'organisation, ses adresses (physiques et web), des éléments de classification, une liste de contacts, ...etc.

ServiceEntity : ce sont en quelque sorte les pages jaunes d'un annuaire UDDI, qui décrivent de manière non technique les services proposés par les différents fournisseurs. On y trouvera essentiellement le nom et la description textuelle des services ainsi qu'une référence à l'organisation proposant le service et un ou plusieurs «bindingTemplates».

BindingTemplate (coordonnées des services) : ce sont des informations qui concernent le lieu d'hébergement du service, elles donnent les coordonnées des services. UDDI permet de décrire des web services HTTP, mais également des services invoqués par d'autres moyens (SMTP, FTP, fax, téléphone,...etc.). Elles contiennent notamment une description, la définition du « point d'accès » (suivant les cas, une URL, un numéro de téléphone, ...) et les éventuels « tModels » associés.

tModel (descriptions techniques des services) : ce sont des informations qui concernent le mode d'accès aux services, ce sont les descriptions techniques des services. UDDI n'impose aucun format pour ces descriptions qui peuvent être publiées sous n'importe quelle forme et notamment sous forme de documents textuels (XHTML par exemple). C'est à ce niveau que WSDL intervient comme le vocabulaire de choix pour publier des descriptions techniques de services.

publisherAssertion : assertions contractuelles entre partenaires dans le cadre des échanges d'exécution d'un service.

V.4.5.L'interface UDDI

L'interface UDDI est définie sous forme de documents UDDI et implémentée sous la forme de web service SOAP. Elle est composée des modules suivants :

Interrogation : Cette interface permet de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements enregistrés suivant le modèle de données UDDI.

Publication : Cette interface permet de publier des informations dans un répertoire UDDI conformément à son modèle de données.

Sécurité : Cette interface est utilisée pour obtenir et révoquer les jetons d'authentification nécessaires pour accéder aux enregistrements protégés dans un annuaire UDDI.

Contrôle d'accès et propriété : Cette interface permet de transférer la propriété d'informations (qui est à l'origine attribuée à l'utilisateur ayant publié ces informations) et de gérer les droits d'accès associés.

Abonnement : Cette interface permet à un client de s'abonner à un ensemble d'informations et d'être avertis lors des modifications de ces informations. Tous les répertoires UDDI doivent gérer un avertissement par polling (le client interroge le serveur pour savoir si des modifications ont eu lieu sur les données auxquelles il est abonné). Une fonctionnalité optionnelle est également prévue permettant au client de communiquer au serveur la définition d'un web service sur lequel il souhaite être prévenu en cas de modification.

Réplication interne (Nœuds d'un même annuaire) : A côté des interfaces utilisateurs que nous venons de voir, UDDI définit également l'interface permettant de synchroniser les nœuds d'un même annuaire UDDI.

Réplication externe (interrogation, publication, abonnement) : La réplication externe par duplication d'informations entre différents annuaires UDDI n'a pas donné lieu à la définition d'une interface spécifique mais se fait en utilisant les interfaces d'interrogation (pour la lecture dans un annuaire), publication (pour la publication dans un autre annuaire) et éventuellement abonnement (pour pouvoir propager les modifications ultérieures).

V.4.6.L'usage de l'annuaire UDDI

L'annuaire UDDI permet, la publication et la découverte des web services:

1. La publication de services web: Un web service peut être publié en publiant sa description, après sa production bien sur. Cette description peut être générée manuellement ou automatiquement à l'aide des outils qui peuvent générer des parties WSDL et de créer des entrées dans UDDI à partir de méta données par exemple. Un UDDI peut être de plusieurs types et leurs utilisations dépendent du domaine des services à publier, à savoir :

Nœud UDDI pour une application interne : les nœuds UDDI se trouvent derrière le firewall, avec ce type de nœud on a plus de contrôle sur l'enregistrement, l'accessibilité, la disponibilité et les spécifications de publication lors de la publication de service.

Nœud portail UDDI : il se trouve à l'extérieur du firewall du fournisseur de service ou

entre les firewalls. Sur ce type de nœud on publie des web services pour les partenaires externes tout en implémentant des mécanismes d'accès sélectifs selon le profil des utilisateurs.

Nœud catalogue du partenaire avec UDDI : les web services peuvent être publiés sur un nœud du catalogue du partenaire avec UDDI (derrière le firewall). Le partenaire est choisi avec une autorisation d'accès spécifique.

Nœud de place de marché UDDI : il s'agit de relations interentreprises et de contrôle du partage de l'information entre systèmes d'information.

2. La découverte de services web: C'est la recherche et la localisation d'un web service particulier dans un annuaire de services décrivant le nom du fournisseur, l'objectif de chaque service. Donc il s'agit de l'acquisition des descriptions de web services et leur utilisation.

Opérateurs d'annuaires UDDI actuels

Opérateurs	Adresse du site
HP	http://uddi.hp.com/uddi/index.jsp
IBM	http://www.3.ibm.com/services/uddi/v2beta/protect/registry.html
Microsoft	http://uddi.microsoft.com/register.aspx
NTT	www.ntt.com
SAP	www.sap.com

VI. La sécurité des web services [8]

Vu que le domaine d'application majeur des web services est les échanges interentreprises (*Business to Business* ou *B2B*), la gestion de la sécurité est un élément fondamental de la technologie des web services.

Notre but ici ne consiste pas à traiter en détail des méthodes utilisées dans le domaine de la sécurité informatique. Par contre, il est nécessaire de rappeler les concepts rencontrés :

- Confidentialité et intégrité garantissent qu'un échange est confidentiel et intact.
- L'authentification consiste à déterminer l'identité, au sens large (utilisateur, machine, application).
- La non-répudiation qui garantit qu'un acteur ne puisse renier la dernière opération qu'il a effectuée.

En règle générale, ces fonctionnalités s'appuient sur des processus de cryptage/décryptage grâce à des clés (publiques et privées).

Au sein d'Internet, la mise en œuvre de ces concepts s'appuie, au niveau du transport, sur le protocole *Secure Socket Layer* (SSL¹). C'est ainsi que les web services sécurisés utilisent non pas SOAP/HTTP, mais SOAP/HTTPS. Il s'agit d'une utilisation de HTTP en combinaison avec le protocole SSL.

Une alternative prometteuse au chiffrement par SSL, est le standard proposé par W3C : *XML Encryption Standard*. Ce dernier définit les spécifications de cryptage et de décryptage d'un document XML, ou de seulement une partie.

Bien que les standards au formalisme XML soient en perpétuelle évolution, un consensus semble se faire autour de propositions suivantes : SAML (*Security Assertions Markup Language*), XKMS (*Public Key management*) et WS-Security. De plus, il ne faut pas oublier l'importance des "pare feu" (*firewall*) éléments techniques indispensables à la prévention des intrusions dans les réseaux informatiques.

- **SAML**

Ce langage, au formalisme XML, permet de définir les informations d'identification et d'autorisation. Sa spécification prévoit que les informations d'authentification puissent se propager de service Web en service Web, en utilisant n'importe quel protocole de transport.

Chaque document SAML comporte un élément XML "Request". Les éléments-fils obligatoires sont "AuthenticationQuery" et "Subject".

Lorsque l'authentification est assurée, le document SAML s'enrichi d'un élément "Response" donnant le résultat de la demande d'authentification. Le document peut alors se propager à l'intérieur de l'en-tête d'un message SOAP afin d'interagir avec les services Web invoqués.

- **XKMS**

La spécification XKMS (*XML Key Management System*) est proposée par la société VeriSign afin d'offrir une infrastructure de gestion des clés publiques. Cette proposition a été retenue par le W3C et fait maintenant partie de ses spécifications.

Se basant sur une autorité de confiance, XKMS devrait fournir un service Web de confiance qui permettrait :

- D'enregistrer une paire de clés et d'obtenir un certificat
- De rechercher une clé publique
- De supprimer une paire de clés.

Autrement dit, l'entité qui fournit l'implémentation du web service XKMS joue le rôle d'autorité de certification. Actuellement, la société VeriSign propose une utilisation gratuite de son implémentation XKMS.

- **WS-Security**

La spécification WS-Security est supportée depuis 2002 par de grands acteurs du domaine de la sécurité informatique, notamment, Microsoft, IBM et VeriSign.

Cette proposition est la plus sérieuse car elle définit une infrastructure complète de la gestion de la sécurité, adaptée aux web services. En effet, WS-Security prévoit une extension du protocole SOAP afin de supporter tous les mécanismes de sécurité : authentification, confidentialité et intégrité. Mais surtout, cette spécification offre la possibilité de véhiculer des jetons de sécurité. En fait, le message SOAP d'un service Web serait crypté en utilisant la spécification XML-Encryption et intégrerait le jeton, vraisemblablement au format X509 ou Kerberos. Toutes ces informations prendraient place dans l'entête du message SOAP.

En résumé, SAML est un langage qui couvre les problématiques d'authentification et d'habilitations des transactions. XKMS vise à standardiser les opérations relatives au traitement des clés publiques à des fins de confidentialité, d'authentification et de répudiation. En fin, WS-Security est un langage proposé pour sécuriser les web services, il s'agit d'une extension de SOAP incluant cryptage, signature, etc.

VII. Conclusion

Les *Web Services* sont le résultat de la collaboration exceptionnelle des joueurs majeurs des technologies de l'information qui se sont entendus sur un certain nombre de protocoles et d'approches qui favoriseront l'interopérabilité entre les plates-formes, les systèmes d'exploitation, les langages et les programmes. De ce fait, les *Web Services* sont

donc plus un phénomène ou un concept, voire un contexte, qu'une technologie. Il est évident que les *Web Services* reposent sur diverses technologies, mais ils représentent surtout une volonté commune des manufacturiers, des organismes de standards et des utilisateurs de développer des outils permettant une réelle interopérabilité.

Dans ce chapitre nous nous sommes intéressés aux technologies sur lesquelles sont basés les web services. Nous avons analysé le moyen de transmission et de transaction qu'est SOAP. Ce dernier permet l'échange d'informations structurées grâce au format XML. D'autre part, nous avons vu un langage de description de services, le WSDL basé aussi sur XML, qui permet de procurer aux applications clientes de nombreux renseignements sur le contenu d'un service et sur les façons de l'invoquer. Puis, nous avons abordé le registre annuaire pour les web services, nommé UDDI et considéré comme un standard. Enfin, nous avons illustré quelques moyens fondamentaux qui participent à la sécurisation des web services.

CHAPITRE II

I. Introduction

Le processus d'apprentissage repose sur une série d'activités soigneusement planifiées par l'enseignant qui a été pendant longtemps considéré comme une ressource irremplaçable.

Dans les années 50 ont eu lieu les premières tentatives d'enseignement à l'aide de machine,

L'évolution de ces systèmes amènera par la suite à les rendre adaptatifs pour personnaliser l'enseignement.

C'est au début des années 80 que la notion l'intelligence artificielle(IA) fut introduite ce qui a permis de prendre en compte tous les aspects de cette technique d'enseignement. Malgré tout, les résultats ne sont pas à la hauteur des espérances.

Avec l'avènement des nouvelles technologies de l'information et de la communication, il faut, dès à présent, «penser à un apprentissage rapide et efficace», avec un minimum de problèmes d'organisation, de logistique et surtout de perte de temps.

Le E-Learning est la solution qui est décrite comme le mariage du multimédia (son, image, texte) et internet (diffusion en ligne, interactivité) le E-Learning apparait à beaucoup comme le second souffle de marché des technologies.

II. E-Learning [17]

II.1. L'enseignement assisté par ordinateur

L'enseignement assisté par ordinateur est l'utilisation de l'outil informatique dans un objectif pédagogique, de formation (enseignement).

II.1.1- Les logiciels de l'EAO ou didacticiels

Un didacticiel est un logiciel d'EAO, spécialisé dans une ou plusieurs matières, mettant un apprenant en interaction avec la matière enseignée dans le but de lui faire acquérir des connaissances.

Il existe plusieurs sortes de didacticiels :

- Le didacticiel de test ;
- Le didacticiel informatif ;
- Le didacticiel d'entraînement ou drill, qui combine les deux types précédents ;
- Le didacticiel de simulation ; Le tutoriel qui utilise toutes les formes précédentes ; le pédagogique, possédant en plus les caractéristiques suivantes :

- Une adéquation pédagogique qui prime tout raffinement concernant les canaux de communication ;
- Les questions « ouvertes » le sont réellement (il n’y a pas de QCM caché) et le traitement des réponses en langue naturelle se fait intelligemment ;
- Les modules d’aides sont appuyés par des filtres logiciels pédagogiques, accédant à une encyclopédie thématique.

II.2. l’Enseignement Intelligemment Assisté par Ordinateur (EIAO)

L’introduction des techniques de l’intelligence artificielle dans l’enseignement assisté par ordinateur a donné naissance à une nouvelle génération de systèmes appelés systèmes d’Enseignement Intelligemment Assisté par Ordinateur (EIAO).

Plusieurs approches ont été explorées. La décennie 80-90 a été marquée par celle des Systèmes Tutoriels Intelligents (STI), fortement liée au développement des systèmes à base de connaissances en intelligence artificielle (figure I.2).

Une première expérimentation dans ce sens a été menée à l’aide d’un système expert nommé GUIDON réalisé par W.J. Clancey à la fin des années 70. L’idée initiale était simple : si on dispose d’un système de résolution de problèmes de niveau expert, avec une base de connaissances explicites, on peut l’utiliser pour former des étudiants, en lui ajoutant des modules adéquats pour assurer la transmission d’information et de cette compétence du système vers l’apprenant.

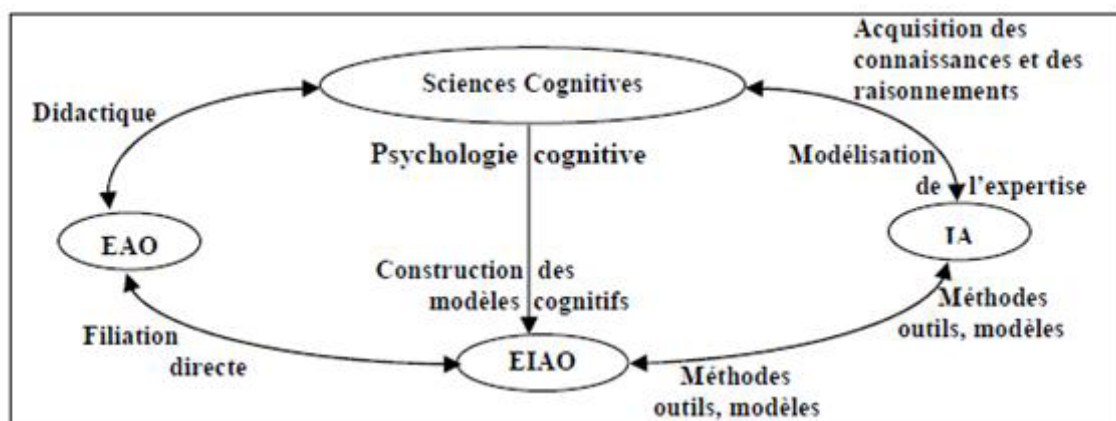


Figure II.1- Naissance de l’EIAO

II.2.1. Architecture d'un système d'EIAO

Un système d'EIAO doit avoir plusieurs aspects :

- L'aspect connaissance de la matière enseignée ;
- L'aspect pédagogique ;
- L'aspect connaissance de l'auditoire.

Pour un enseignement de qualité, le système d'EIAO doit assurer les fonctions suivantes :

- disposer de connaissances sur le contenu à enseigner ;
- savoir diagnostiquer les difficultés de l'élève et pouvoir s'adapter à son profil;
- adopter des méthodes ou stratégies d'enseignement ;
- avoir une interface de communication conviviale avec l'élève.

Pour cela un système d'enseignement (EIAO) se compose de quatre modules :

- L'interface élève ;
- Le module pédagogique ;
- Le modèle élève ;
- L'expert.

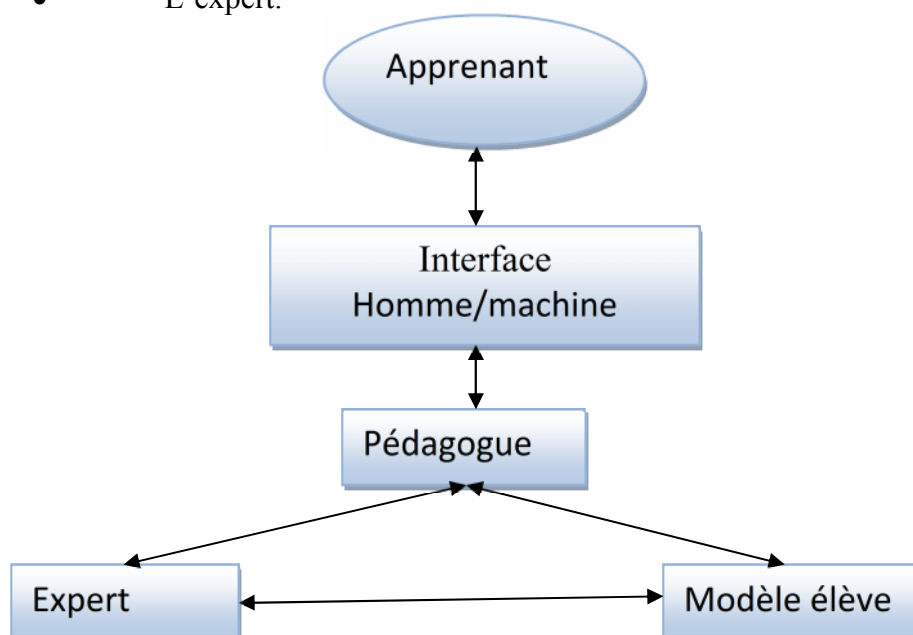


Figure II.2 - Les composantes d'un système d'EIAO**II.2.1.1. L'interface élève (Home/Machine)**

Cette interface permet à l'apprenant de communiquer avec le pédagogue. Il est important qu'elle soit ergonomique, conviviale et facile à utiliser afin que l'apprenant se sente à l'aise dans son apprentissage.

II.2.1.2. Le pédagogue

Le pédagogue est le noyau du système. Il a la charge de coordonner les interactions avec l'apprenant pendant la session d'enseignement. Il choisit le scénario à lancer ; en tenant compte du niveau de connaissance de l'apprenant, de l'objectif d'enseignement fixé et d'un certain nombre de variables pédagogiques.

II.2.1.3. L'expert

L'expert a pour tâches :

- La génération de problèmes,
- La résolution de problèmes,
- Le test des réponses de l'élève (diagnostic des erreurs),
- L'explication.

II.2.1.4. Le modèle élève

Les systèmes d'EIAO se caractérisent par l'individualisation de l'enseignement, d'où l'utilisation d'un modèle propre à chaque apprenant.

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.

Plusieurs modèles sont utilisés pour représenter les connaissances de l'élève, on distingue :

- Le modèle de recouvrement (modèle overlay),
- Modèle des erreurs,
- Le modèle différentiel

II.2.1.4.1. Le modèle de recouvrement (modèle overlay)

Les connaissances de l'élève sont un sous-ensemble des connaissances de l'expert du domaine. Les connaissances appartenant à ce modèle sont considérées acquises par l'apprenant et celles n'appartenant pas au modèle sont considérées non acquises.

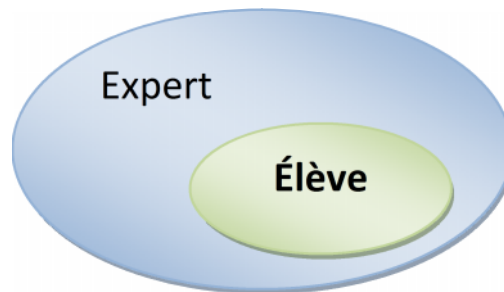


Figure II.3 Le modèle Overlay

II.2.1.4.2. Le modèle des erreurs (modèle bugs)

Les connaissances de l'élève constituent ici une perturbation des connaissances de l'expert. En plus des connaissances maîtrisées par l'apprenant, ce modèle considère les erreurs susceptibles d'être faites par l'élève.

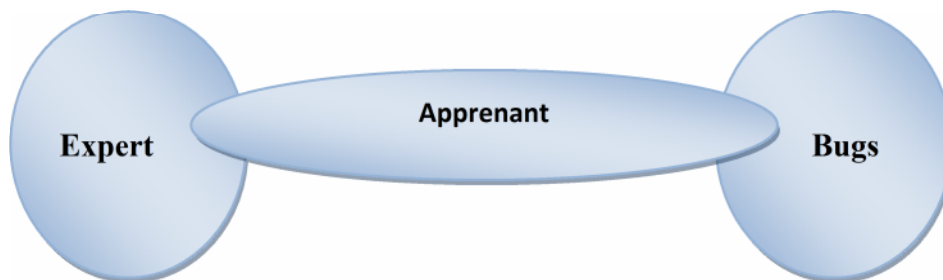


Figure II.4 : Le modèle bugs

II.2.1.4.3. Le modèle différentiel

Dans ce modèle, les points forts de l'élève et ses points faibles sont mis en évidence par la différence entre la démarche de l'élève et celle de l'expert.

II.2.2. Les modes d'interaction

Plusieurs modes d'interaction entre l'élève et le module pédagogique existent allant des plus guidés aux plus libres. Ils dépendent de la nature des connaissances à transmettre et de celles du public visé.

II.2. 2.1. Mode libre

Ce mode convient plus à l'enseignement de concepts abstraits et à la formation destinée aux experts. Ici l'élève doit avoir une grande liberté d'action et le pédagogue ne se doit d'intervenir qu'en cas d'extrême urgence.

II.2.2.2. Mode guidé

Il convient à l'enseignement de la résolution de problèmes dans un domaine spécifique. Le pédagogue intervient à des moments bien choisis.

II.2.2.3. Mode mixte

C'est un mode où l'initiative est partagée entre l'élève et le pédagogue. L'élève découvre lui même son erreur sans que le pédagogue ne l'aide trop tôt ou ne le laisse trop longtemps persister dans l'erreur.

II.3. Environnement Interactif pour Apprentissage avec Ordinateur (EIAO)[16]

Parallèlement aux systèmes d'EIAO, se sont développés des logiciels de type micromondes. Ces logiciels sont d'avantage centrés sur la notion d'apprentissage, plutôt que sur celle de l'enseignement, et en proposant une approche constructiviste.

Contrairement aux Tuteurs Intelligent (TI), ce n'est pas l'ordinateur qui joue le rôle du tuteur et enseigne à l'apprenant, mais c'est ce dernier qui gère lui-même son apprentissage, en totale autonomie et à l'aide d'ordinateur.

Dans un environnement de type micromonde l'apprenant évolue dans un univers abstrait représentant une simulation simplifiée du domaine qu'il étudie. En manipulant les concepts de ce monde abstrait, l'élève a ses propres représentations et connaissances à propos du domaine. L'exemple le plus connu de micromonde est LOGO.

Cette catégorie de logiciels trouve néanmoins sa faiblesse dans le manque de guidage et d'assistance offert à l'apprenant, qui travaille en total autonomie et peu se trouver assez vite

perdu lorsqu'il rencontre des difficultés. Cette évolution du sigle EIAO permet de spécifier les nouvelles orientations de recherche en informatique éducative.

Tout d'abord, le terme «Enseignement» a été remplacé par le terme «Apprentissage» car ces systèmes s'intéressent désormais d'avantages à l'activité d'apprentissage de l'apprenant et à la construction de ces connaissances, qui lui sont transmises. Ensuite le mot «Assisté » par Ordinateur est remplacé par «Avec » Ordinateur, pour mettre en évidence la nouvelle place de l'ordinateur dans le processus éducatif. L'ordinateur n'est plus le tuteur. Il devient l'un des acteurs du scénario d'apprentissage induit par l'EIAO, au même titre que l'enseignant ou encore les co-apprenants. Nous pouvons citer comme exemple de système d'EIAO, le système multi-agent réalisé par Claude Moulin.

II.4. Hypertexte et Environnement Interactif pour Apprentissage avec Ordinateur EIAO [19]

II.4.1. Définition de l'hypertexte

C'est une organisation permettant la lecture non linéaire des documents sur l'initiative du lecteur. Toutefois, la racine texte dans l'hypertexte n'a pas un caractère restrictif au contenu de l'hypertexte.

Ce dernier n'est pas en effet limité au seul aspect textuel des documents, mais il concerne toutes les formes de documents y compris bien entendu les formes multimédia (son, image, vidéo, ect.

II.4.2. Influence des hypertextes sur l'EIAO

Certaines fonctionnalités des systèmes hypertextes semblent fort intéressantes dans les systèmes d'EIAO, par contre lorsqu'on veut réaliser une application éducative avec les hypertextes existants, on s'aperçoit que certaines caractéristiques de ces systèmes s'opposent à la conception d'un environnement d'enseignement à guidage contrôlé.

Les systèmes hypertextes existants donnent plus de responsabilité à l'apprenant que les systèmes d'EIAO*, tant dans l'accès à l'information, dans le déroulement d'une session que dans l'interprétation des informations. Ces nouvelles responsabilités nécessitent de la part de l'utilisateur, la mise en œuvre de nouveaux processus cognitifs.

II.5. Environnement Interactif pour l'Apprentissage Humain (EIAH)

Nous assistons depuis quelques années à un glissement des EIAO vers des environnements permettant la communication et l'interaction entre des machines et des

humains distribués dans l'espace. Dans ce sens, un nouveau terme est apparu récemment pour qualifier les environnements d'apprentissage avec ordinateur : les Environnements Informatiques pour l'Apprentissage Humain (EIAH).

D'un point de vue diachronique, ce sigle permet de prendre en compte l'évolution récente des Technologies de l'Information et de la Communication (TIC), et élargit le champ d'étude à l'apprentissage humain dans toutes ses déclinaisons (enseignement, formation, autodidaxie, diffusion de connaissances, etc.).

Il recouvre à la fois les EIAO tels que nous les avons décrits précédemment, et des EIAO qui seraient plus tournés vers la formation à distance (FAD). Cette dimension apparaît actuellement de plus en plus centrale dans le contexte des travaux fondés sur les capacités offertes par la généralisation d'Internet.

II.6. E-Learning : [16]

II.6.1. Définitions des concepts

II.6.1.1. Qu'est-ce que l'enseignement à distance ou formation à distance (EAD/FAD) ?

La formation à distance est un système de formation qui permet de se former sans se déplacer sur le lieu de formation et sans la présence physique d'un formateur. La transmission des connaissances et les activités d'apprentissage se situent en dehors de la relation directe en face à face, dite « en présentielle » entre l'enseignant et l'apprenant. La formation à distance est une forme d'enseignement caractérisée par :

- La séparation quasi-permanente entre le formateur et l'apprenant tout au long du processus d'apprentissage (ceci différencie la formation à distance (FAD) de la formation présentielle) ;
- L'influence d'une organisation administrative aussi bien en ce qui concerne la planification et la préparation des matériaux pédagogiques que la mise à la disposition des apprenants des services d'accompagnement et de support ;
- L'utilisation de médias techniques (imprimerie, audio, vidéo, ordinateurs) pour assurer le lien entre le formateur et l'apprenant et médiatiser le contenu de la formation ;

- L'existence de mécanismes de communication bidirectionnelles afin que l'apprenant bénéficie mais prenne aussi l'initiative de dialogues avec le formateur (ceci distingue la FOAD des autres usages de la technologie dans l'éducation) ;
- La quasi-absence de la notion de groupe tout au long du processus d'apprentissage, de sorte que les apprenants sont toujours vus comme des individus isolés et non comme faisant partie d'un groupe, avec la possibilité d'organiser occasionnellement des rencontres, soit en présentielle, soit via des moyens électroniques.

II.6.1.2. Qu'est-ce que la formation ouverte et à distance (FOAD) ?[18]

La formation ouverte ou Open Learning, permet aux apprenants des entrées et des sorties permanentes. Elle correspond à un mode d'organisation pédagogique diversifié qui s'appuie sur des apprentissages à distance, en auto-formation. En formation ouverte, l'apprenant peut alterner des séquences individuelles et collectives. L'auto-formation est un processus par lequel l'individu détermine son itinéraire d'apprentissage (rythme, contenu, temps de travail) de façon autonome et en étant éventuellement en relation avec un tuteur ou un groupe structuré.

Une personne qui s'inscrit à un dispositif de formation ouverte et à distance peut, à partir de sa maison ou de son poste de travail, avec une connexion Internet, utiliser un navigateur pour accéder à la plate-forme du cours. Une fois enregistrée (identifiant et mot de passe) elle peut suivre la formation en consultant les documents pédagogiques, participer aux séances de cours en mode synchrone ou en asynchrone, envoyer des questions au tuteur du module. Une formation ouverte et à distance :

- Est un dispositif organisé, finalisé, reconnu comme tel par les acteurs.
- Prend en compte la singularité des personnes dans leurs dimensions individuelle et collective.
- Repose sur des situations d'apprentissage complémentaires et plurielles en termes de temps, de lieux, de médiations pédagogiques humaines et technologiques et de ressources.
- Se positionne avec les NTIC.

II.6.1. 3. Qu'est-ce que la Téléformation ?

Elle peut être définie comme une formation effectuée par des moyens de télécommunication en temps réel ou différé. Le concept de téléformation est aujourd'hui largement décliné, tout en recouvrant des pratiques et en mobilisant des acteurs dont les objectifs bien qu'identiques en apparence (former à distance), obéissent cependant à des logiques différentes, selon le contexte dans lequel ils sont poursuivis :

S'il y a évidemment dans tous les cas une démarche pédagogique, il est clair qu'elle pourra avoir, selon le cas, une finalité commerciale, scientifique ou liée à des impératifs de gestion de ressources humaines.

II.6.1.4. Définition du e-Learning (e-formation) ?

L'e-Learning est un processus d'apprentissage à distance, qui repose sur la mise en place d'un dispositif de contenus pédagogiques via un réseau de type Internet ou Intranet et permet ainsi à une ou plusieurs personnes de se former à partir d'un ordinateur.

Les supports multimédias utilisés peuvent combiner du texte, des graphismes, du son, de l'image, de synthèse, de l'animation et même de la vidéo. Ces supports permettent une nouvelle approche pédagogique, avec l'emploi de méthodes plus attrayantes où l'interactivité joue un grand rôle, et avec la possibilité de s'adapter davantage au processus d'apprentissage de l'apprenant.

L'utilisateur peut 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. Le schéma suivant résume les concepts définis ci-dessus

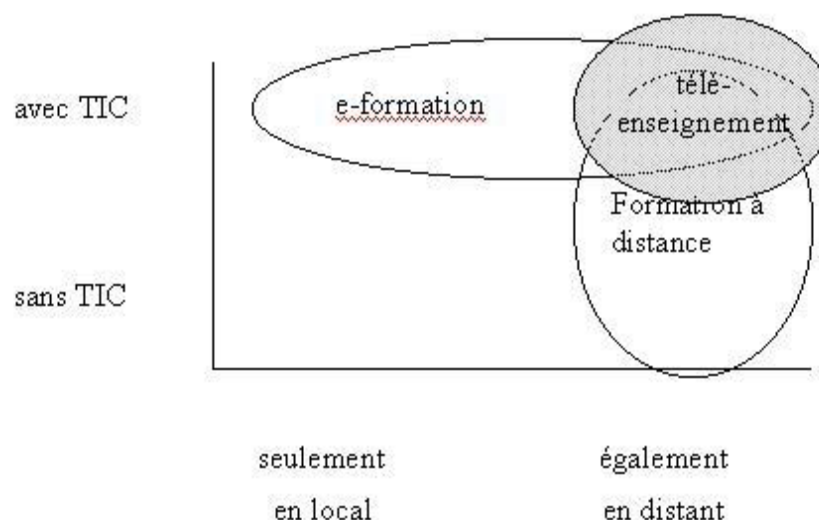


Figure II.5 : les concepts : formation à distance, téléformation et e-Learning

II.6.2. Apport des nouvelles technologies à l'enseignement

Les progrès de l'informatique et des réseaux tel que Internet, ou plus largement des nouvelles technologies de l'information et de la communication (NTIC) a permis de créer des systèmes d'enseignement ou le maître mot est « multimédia ».

N'importe qui peut désormais mettre un cours en ligne, mais construire une ressource pédagogique de qualité avec ces nouvelles technologies ne veut pas dire qu'il suffit de mettre en ligne un polycopié en y ajoutant quelques animations. Les Technologies de l'Information et de la Communication Educatif (TICE) doivent apporter une réelle plus value par rapport aux ressources informatiques déjà existantes.

II.6.3. Les formes d'e -Learning

Nous commencerons par différencier les formes de communication dans l' e-learning puis les types d'apprentissage sous-jacents.

II.6.3.1. Les formes de communication

Il existe deux formes de communications, la forme synchrone et la forme asynchrone :

II.6.3.1.1. La forme synchrone :

Elle correspond à une situation dans laquelle les apprenants peuvent se communiquer directement via des chats, des visioconférences ou audioconférence. L'interactivité est aussi immédiate qu'en présentielle (un ou plusieurs formateurs et tous les apprenants sont regroupés au même instant pour la réalisation de la formation).

- **Les principaux avantages de la formation synchrone**

- ❖ La dynamique de groupe s'installe plus rapidement et les apprenants s'interagissent intensivement à l'écran avec les formateurs et entre eux ;
- ❖ On peut rapidement créer du contenu prêt à diffuser ;
- ❖ Utilisation de langage oral et visuel.

- **Les principaux inconvénients de la formation synchrone**

- ❖ Il faut posséder le matériel technique nécessaire (Webcam, casque, carte vidéo) et une bande passante suffisamment large ;

- ❖ Elle ne permet pas à chaque apprenant d'évoluer à son propre rythme et de choisir l'ordre des éléments du cours à sa guise ;

- ❖ Réduit le taux de contributions des acteurs faute de temps limité des séances d'enseignement ce qui ne permet pas aux apprenants d'approfondir certains aspects du cours ;

- ❖ Volatilité des sessions synchrones.

II.6.3.1.2. La forme asynchrone :

Décrit une situation où il n'y a pas de contact simultané (temps réel) entre l'apprenant et le formateur ; elle correspond à un mode de communication alterné. La communication s'opère essentiellement par courrier électronique (e-mail) et forum de discussion.

Le problème principal de cette forme de communication est le temps qui peut s'écouler entre une question d'un apprenant et la réponse du formateur, temps durant lequel l'apprenant évolue dans son apprentissage, si bien que la réponse reçue ne sera peut être plus en rapport avec sa nouvelle situation d'apprentissage.

- **Les principaux avantages de la formation asynchrone**

- ❖ Les apprenants évoluent à leur propre rythme ;

- ❖ Ils peuvent adapter l'ordre dans lequel ils appréhendent les éléments du cours ;

- ❖ Ils peuvent revoir et approfondir certains aspects du cours à leur guise ;

- ❖ Un temps de réflexion est donné à chaque acteur (apprenant, formateur) pour donner leurs contributions.

- **Les principaux inconvénients de la formation asynchrone**

❖ L'apprenant est seul devant son écran. En cas de difficulté, il ne peut s'appuyer sur aucune aide extérieure (qui renforce le sentiment d'isolement) ;

❖ Le temps qui peut s'écouler entre une question d'un apprenant et la réponse du formateur peut être long.

II.6.3.2. Les formes d'enseignement

L'e-Learning peut être classé en cinq méthodes d'enseignement principales :

- L'auto-formation en solitaire,
- L'auto-formation avec tutorat,
- L'auto-formation avec tutorat asynchrone,
- La classe virtuelle en temps réel,
- La visio formation personnalisée.

II.6.4. Les avantages et les faiblesses du e-Learning

II.6.4.1. Les avantages du e-Learning

Pour les apprenants

L'e-Learning offre de nombreux avantages pour les apprenants. Il permet notamment :

- D'acquérir un diplôme ou une qualification dans un domaine donné,
- De compléter un enseignement ou une formation (perfectionnement),
- De se doter de compétences nouvelles à des fins d'enseignement ou professionnelles (actualisation de connaissances).
- D'étudier dans des conditions particulières mieux adaptées,
- À chacun d'être plus responsable de son apprentissage ou de sa formation,
- Un gain de temps considérable,
- Une facilité d'accès (dématérialisation du lieu),
- Un suivi personnalisé (tuteur en ligne),
- D'aller plus loin par le biais de tuteurs adaptés à chaque classe d'apprenants ou à l'aide de liens Internet vers d'autres sites traitant du même sujet.

- La possibilité d'échange avec les autres apprenants à une grande échelle à l'aide de forum ou de chat.

Pour les formateurs

Le e-Learning permet pour le formateur:

- Un gain de temps, pas de déplacement,
- Un enrichissement dynamique du contenu,
- Une évaluation des pré-requis plus facile grâce à l'utilisation de jeux ou de QCM interactifs,
- Une évaluation plus précise grâce à des tests en continu tout le long de l'apprentissage.

Pour les entreprises

Dans le cadre de la formation à distance le e-Learning permet :

L'économie des coûts liés au déplacement (transport, hébergement, coût de l'indisponibilité),

- La possibilité d'avoir un nombre très important d'apprenants,
- Plus de flexibilité vis à vis des formations,
- La réduction des coûts globaux.

II.6.4.2. Les faiblesses d'e-Learning

Les faiblesses d'e- Learning sont nombreuses. Elles sont parfois liées à la manière dont les dispositifs sont conçus et parfois aux acteurs eux- mêmes:

- La formation et la motivation des tuteurs.
- Les apprenants ne disposent pas des compétences techniques minimales.
- Les forts préjugés des novices.
- La nécessité d'un équipement coûteux.
- Les activités prennent plus de temps que prévu.
- Les effectifs se concentrent très souvent autour de compétences qualifiées.
- Les formations ne sont pas accessibles à tous.

- Le challenge des contenus pédagogiques à améliorer n'est pas encore gagné.
- L'effort d'apprentissage par l'e-Learning est plus important que par la formation classique.

En effet l'apprenant ne peut être passif : il est l'acteur de sa formation.

- Les contraintes techniques de la plate-forme utilisée, chaque connexion peut être un problème potentiel, et le choix d'une plate-forme adaptée à tout le monde n'est pas évident.

II.7. Plates -formes d'enseignement à distance[17]

II.7.1 Définition :

Une plate-forme d'enseignement à distance (EAD), appelée parfois LMS (Learning Management System), est un système informatique de gestion de formation et de gestion de contenu de formation. Les dernières générations intègrent ou s'interfacent avec des outils de communication tels que : chat, messagerie instantanée, visioconférence, etc.

Au sein de projets concrets de formation, elles favorisent l'apprentissage en ligne et le travail collaboratif.

Il s'agit d'une composante d'un dispositif e-Learning mais ce n'est pas la seule. Il est entre autre un des outils des Technologies de l'information et de la communication (TIC) et des Technologies de l'information et de la communication pour l'éducation (TICE).

Une plate-forme pédagogique est un logiciel qui assiste la conduite des formations présentielles et à distance. Elle est basée sur des techniques de travail collaboratif et regroupe les outils nécessaires aux trois principaux acteurs de la formation : apprenant, tuteur, administrateur.

Elle fournit à chaque acteur un dispositif qui a pour première finalité l'accès à distance au contenu pédagogique, l'auto apprentissage, l'autoévaluation et la télé tutorat via l'utilisation des moyens de travail et de communication à plusieurs : visioconférence, e-mail, forums, chats, etc.

Le but est donc de combler la perte de cohésion et de stimulation de la salle que peut sentir l'apprenant devant sa machine.

II.7.2 Les principaux critères que doivent vérifier les plates-formes :

1. Le coût total de la mise en place de la plate-forme (gratuite ou non), dans le cas d'une plate-forme payante, vérifier les prestations fournies.
2. Les besoins de maintenance de la plate-forme d'un point de vue strictement informatique, tout ce qui devra être géré par les administrateurs (voir également le coût en personnel de maintenance).
3. La sécurité de la plate-forme d'un point de vue purement informatique (filtrer les IPs qui auront accès aux cours) et au niveau des cours (login, mot de passe).
4. Le nombre d'utilisateurs maximum que pourra supporter la plate-forme.
5. La possibilité de mettre en place des QCM (Questions à Choix Multiples) sur la plateforme, s'il est possible d'insérer une image, si le système de questions est interactif (avec animations), s'il y a possibilité de faire un tirage aléatoire des questions.
6. Un forum de discussion dont l'accès sera étroitement lié à la sécurité interne de la plateforme.
7. La technologie utilisée, du point de vue du langage pour pouvoir éventuellement faire évoluer le système et sa capacité d'accueil (XML, MySQL, PHP...).
8. Un système de gestion de mails, pour qu'un formateur puisse éventuellement envoyer des mails à l'une de ses classes, également lui permettre de gérer des groupes.
9. La façon de poster un cours.
10. Les moyens requis pour consulter la plate-forme (browser, système d'exploitation...).
11. La possibilité de diffuser les cours par vidéo soit en temps réel, soit en différé.

II.7.3 Outils de communication :

Les plates-formes intègrent un certain nombre d'outils de communication, la liste des outils disponibles dépend de la richesse de la plate forme.

Ces outils sont souvent très usités sur le Web et ne sont en rien spécifiques au e-Learning, ils font simplement partie intégrante de la plate forme et ne nécessitent pas de configuration spécifique sur les postes des apprenants (par exemple, il n'est pas nécessaire de configurer un client messagerie pour profiter de ce service).

On peut citer :

La messagerie : C'est un espace de communication entre les membres du groupe apprenants et formateurs. Ce système permet l'envoi et la réception de mails avec ou sans fichier attaché. Il peut être interne à la plateforme et ne nécessite pas d'avoir un e-mail personnel.

Le forum : pouvant être public ou réservé à un groupe, permet aux apprenants et aux formateurs, de poster des messages qui seront accessibles à l'ensemble des membres du groupe de formation, de répondre aux messages déjà postés et ainsi engager une discussion sur un sujet donné.

Le Chat : permet à l'ensemble des membres du groupe de discuter en temps réel.

Les documents pédagogiques partagés : Une zone commune peut être utilisée pour mettre à disposition du groupe différents documents sous forme numérique (Word, Excel, PDF, etc.)

Les news : La possibilité de créer des news permettra, par exemple, d'informer ses apprenants de nouveaux rendez-vous, ou de leur communiquer diverses informations.

Les sondages : permettra de poser des questions ouvertes ou de proposer des choix de réponses. Ultérieurement, les réponses obtenues sur un cours en particulier pourront être analysées.

Le Bloc-notes : C'est un espace privé à chaque utilisateur. Il peut y noter toutes les informations qu'il souhaite ce qu'il a retenu de sa formation, des actions à mener...

Les Glossaires : Des glossaires peuvent être créés, destinés à l'ensemble des utilisateurs ou un groupe en particulier. Les différents termes avec leur signification sont enregistrés, un moteur de recherche permet aux utilisateurs de faciliter la recherche d'un terme dans les glossaires.

II.7.4 Acteurs de la plate-forme LMS :

Il y a *trois* principaux types d'utilisateurs accédant à la plate-forme : *l'administrateur, les apprenants et les enseignants (formateurs)*.

1. Enseignant :

Chaque enseignant est responsable d'un ou plusieurs modules d'enseignement. La plate-forme permet à l'enseignant de structurer les modules en différentes briques de contenus de cours. Il peut ajouter un contenu de cours en téléchargeant sur le serveur de la plate-forme n'importe quel type de fichier. Il peut mettre à jour et supprimer chaque contenu.

2. Apprenant :

Un apprenant peut consulter ces différentes briques de cours et éventuellement les télécharger pour une lecture ultérieure hors connexion.

3. Administrateur :

L'administrateur, de son côté, assure l'installation et la maintenance du système, gère les droits d'accès, crée des utilisateurs, les intitulés de module etc. Il y a quatre types de contenus de cours : exercice, ressource, test, concept.

II.7.5 Quelques fonctionnalités de la plate-forme LMS :

II.7.5.1 Modules d'édition de contenus de cours:

Les modules sont divisés en topic (semaines de travail) différents selon les enseignants concernés dans chaque topic, il y a une liste de travaux à faire de 4 types (exercices, ressources à lire, concepts, tests).

Chaque module doit avoir un code l'identifiant. Par exemple, pour système d'exploitation, le code serait SE (il faut savoir comment déterminer automatiquement ce code de manière à ce qu'il soit significatif). Il est important de mettre ce code à côté du nom de module et des contenus de cours proposés. Ainsi, l'étudiant pourra facilement se repérer.

Par ailleurs, le forum est associé au module et un forum général où les étudiants peuvent communiquer librement sur tout un tas de sujets indépendants de leur formation.

II.7.5.2 Gestion des inscriptions :

La plate-forme dispose d'un système de gestion des inscriptions. Ce système permet l'inscription aux formations utilisant la plate-forme. Il s'agit d'un formulaire à remplir en ligne, qui permet différents types d'inscriptions :

- Inscription en ligne automatique avec validation d'administrateur de type « staff »
- Inscription manuelle par un administrateur en rentrant directement un utilisateur.
- Inscription manuelle par un administrateur en rentrant un fichier contenant une liste d'utilisateurs (*en format Excel*).

II.7.5.3 Dépôts de fichiers :

Le dépôt de fichiers sert aux étudiants pour rendre les tests et aux professeurs pour récupérer ces fichiers « rendus de test » et mettre à disposition d'éventuelles corrections.

Il y a un dépôt commun par module, les étudiants disposent d'un certain temps pour mettre les « rendus de test » dans le dépôt.

Chaque élément du dépôt contient les informations suivantes :

- Auteurs (nom, prénom) ;
- Type (rendu de test ou rendu d'exercices ou correction) ;
- Nom physique du fichier ;
- Module associé ;
- Rendu avant la date limite (oui/non).

Les professeurs du module associé au dépôt peuvent lire tous les fichiers déposés et ils peuvent déposer les fichiers de type « correction » dans le dépôt.

Les étudiants peuvent lire les fichiers qu'ils ont déposés et les fichiers de type correction.

Les administrateurs peuvent avoir accès aux dépôts des modules et supprimer ou modifier ces dépôts. La modification des dépôts consiste à supprimer ou ajouter des fichiers dans le dépôt.

II.7.5.4 Gestion des notes :

La plate-forme dispose d'un système de gestion des notes de chaque apprenant. Il y a une partie de cette gestion des notes qui sera gérée par les enseignants eux-mêmes et une autre partie qui sera gérée par l'administrateur.

La gestion des notes comprend différentes fonctionnalités :

- Ajout des notes manuellement ;
- Ajout des notes par import de fichiers ;
- Consultation des notes par les étudiants ;
- Création automatique de bulletins de notes ;
- Création de tableaux récapitulatifs de notes.

II.7.6 Architecture d'une plate forme :

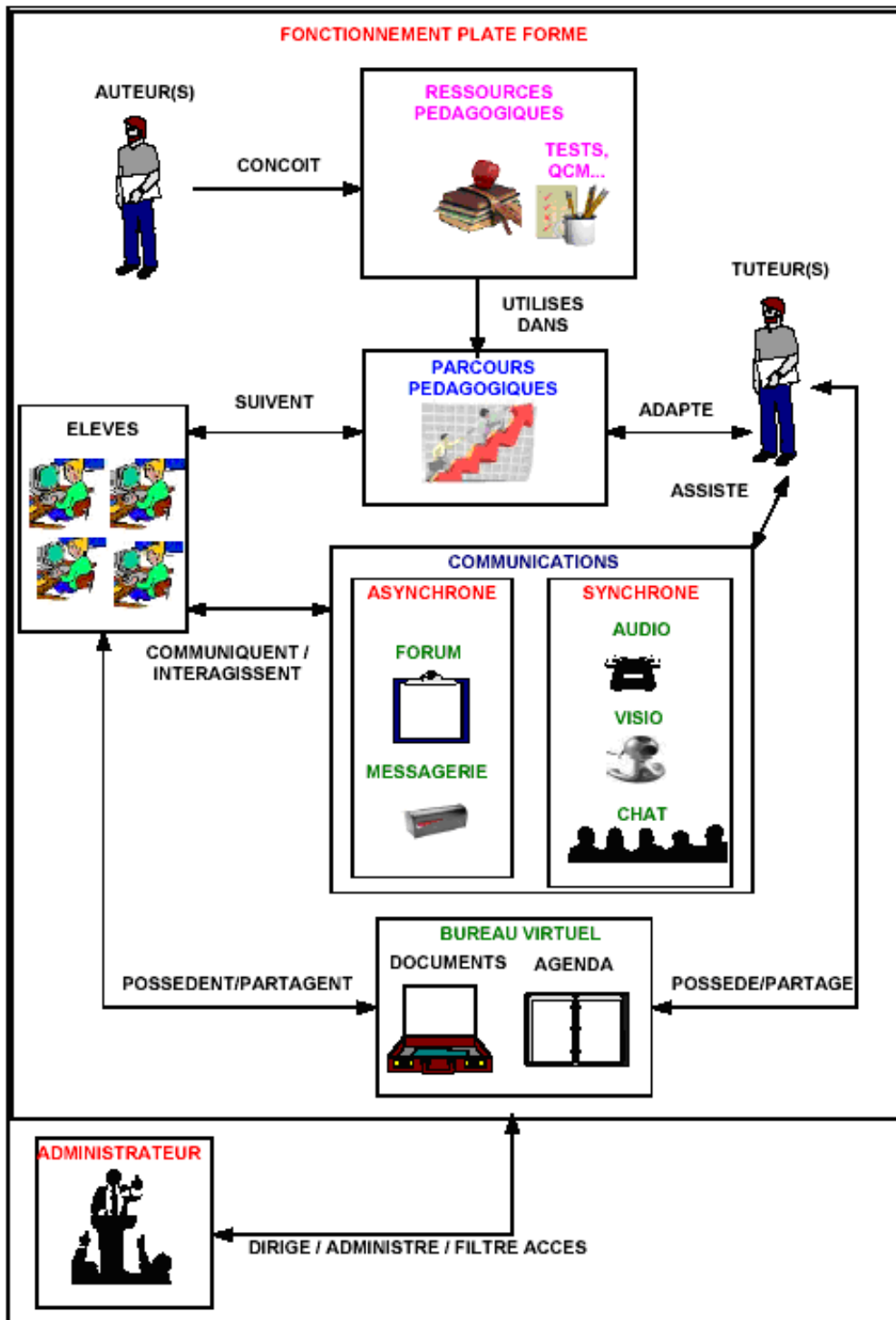


Figure II.6 - Architecture d'une plate forme

II.7.7 Exemples de plates formes :

- **Moodle [20]**

C'est une plate-forme qui a été lancée par un étudiant australien, qui en a fait comme un sujet de thèse de doctorat. Il a réalisé une plate-forme qu'il a mise à disposition à source ouverte.

Cette plate-forme vue sa modularité et la dynamique des développeurs qui ont travaillé dessus, a vite évolué et a séduit des milliers d'établissements. Actuellement elle s'inscrit parmi les plus grandes plates-formes dans le monde avec plus de 175 pays et 75 langues prises en charge. Il existe des plates-formes d'e-Learning basé sur Moodle et qui dispense des cours à plus de 100000 étudiants.

Moodle est le nom d'un programme qui permet d'étendre la notion de la classe sur internet.

Ce programme met à disposition une zone commune pour les élèves pour accéder à des ressources de cours. En utilisant Moodle, vous pouvez éditer de nouveaux sujets, assigner et collecter des devoirs, publier des ressources et plus encore...

Ce manuel est une introduction à l'utilisation des différentes possibilités du logiciel.

Dans sa version de base, Moodle propose trois formats de cours, qui sont en fait différents modes de présentation.

Pour faire simple, un cours se présente comme une série de *boîtes* qui contiennent chacune une liste de ressources et d'activités.

1. Le format thématique (*topic*) propose une *boîte* par sujet abordé dans le cours, de façon similaire aux chapitres d'un livre.
2. Le format hebdomadaire (*weekly*) propose une *boîte* pour chaque semaine pendant la durée du cours.
3. Le format informel (*social*) quant à lui, propose simplement un forum de discussion sans boîte de contenu supplémentaire.

Cette plate-forme offre plusieurs avantages tels-que :

- ❖ Grâce à l'architecture modulaire de Moodle, le développement d'activités complémentaires est relativement aisé;
- ❖ L'interface utilisateur est simple et conviviale;
- ❖ Les outils proposés dans Moodle ne renforcent pas la tendance à produire des encyclopédies en ligne;
- ❖ La communauté des utilisateurs et développeurs est internationale, large et active.

III. Travaux pratiques à distance télé-tps

III.1.Introduction

Les travaux pratiques à distance constituent un pan de l'enseignement à distance qui n'a été que peu abordé dans les recherches scientifiques actuelles. Pourtant ce type d'enseignement est indispensable dans les disciplines scientifiques et techniques et répond à un véritable besoin : les installations industrielles lourdes et onéreuses ne peuvent ni être déplacées, ni être dupliquées. Or, il faut qu'à un moment donné de sa formation, l'apprenant soit confronté à leur pilotage en vraie grandeur. Comment, dans ces conditions (à distance), y donner accès avec à la fois le réalisme d'instruments réels et la télémessure synchrone, avec rétroaction par canal audio – vidéo - informatique.

Nous ne considérons pas que la simulation d'un système physique relève du TP à proprement parler dans la mesure où elle ne reproduit que ce que l'on y a mis (absence de contexte et de prise en compte de l'environnement du système, le plus souvent).

Toutefois, elle constitue un excellent support pédagogique intermédiaire entre l'académique et la réalité.

III.2. Situation des télé-TPs en e-learning[16]

Situés au même niveau que les télé-cours, les télé-TDs, les télé-projets, les intranets pédagogiques et les jeux de rôle, les télé-Tps sont considérés comme un pôle pratique de l'e-Learning comme le montre la figure suivante :

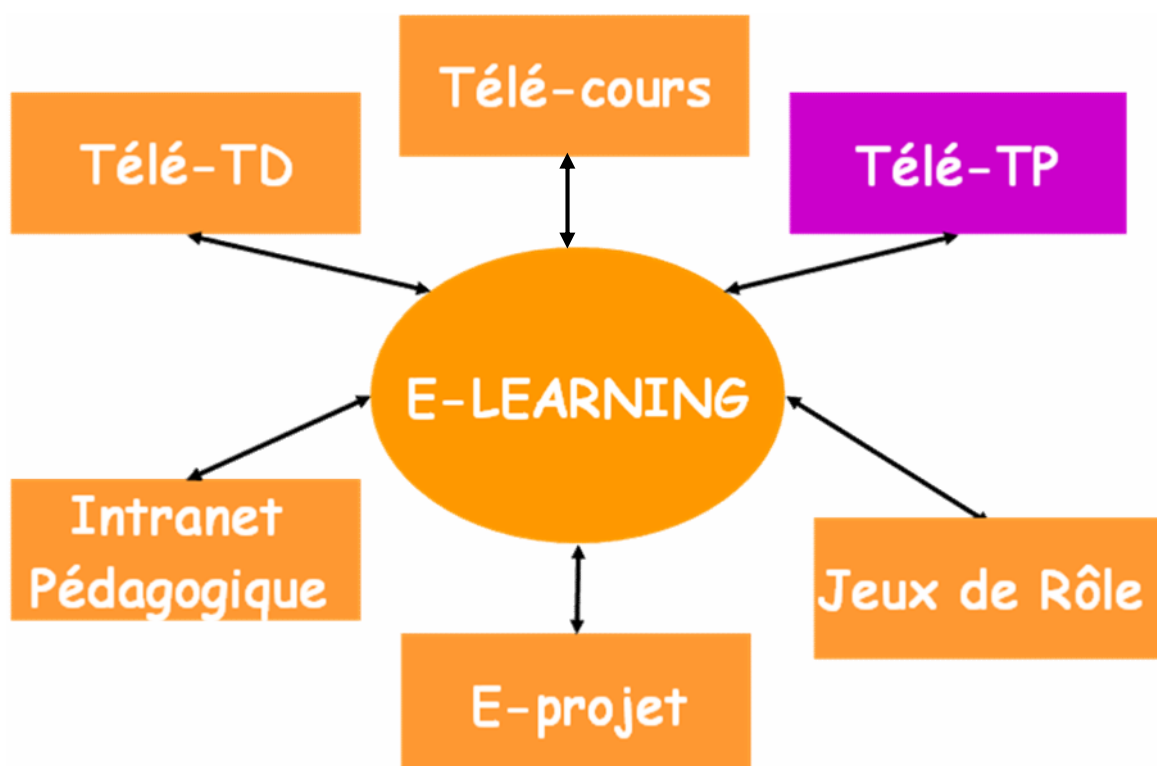


Figure II.7- Situation des télé-TPs en e-learning[16]

Télé-cours : suivi des cours ou de conférences à distance.

Télé-TD : réalisation de travaux dirigés et d'exercices à distance.

Télé-projet : réalisation de projets collectifs et à distance grâce à une architecture permettant le travail collaboratif.

Intranet pédagogique : Un intranet pédagogique est un réseau local d'ordinateurs constitué d'au moins un serveur Web et d'un ensemble de clients connectés à ces serveurs. Les clients sont munis d'un logiciel de navigation permettant de consulter les documents disponibles sur le ou les serveurs.

Jeux de rôle : méthode de formation qui vise une mise en situation imaginaire autour d'un thème de la vie courante ou professionnelle.

III.3. Les travaux pratiques

Les travaux pratiques dans l'enseignement sont une forme d'activité d'apprentissage qui se fait dans un laboratoire et qui permet à un groupe d'élèves d'expérimenter les principes théoriques appris-en suivant les consignes de l'enseignant.

L'importance des travaux pratiques a été considérée comme étant un élément favorisant les interactions entre apprenants et apprenants-formateurs d'un côté et mettant en œuvre le principe de l'apprentissage par essai/erreur d'un autre côté.

III.3.1. Les objectifs pédagogiques d'un TP

Les objectifs pédagogiques des travaux pratiques sont:

- Fournir illustrations et démonstrations des principes enseignés et donc une meilleure assimilation des apprenants.
- Motiver les élèves et focaliser les interactions entre apprenants et entre apprenants-formateurs.
- Développer des compétences pratiques considérées comme importantes d'un point de vue professionnel.
- Développer des compétences de travail collaboratif en équipe.
- Introduire les élèves dans la communauté de pratique des scientifiques.

III.3.2 Typologie d'un TP

Rappelons que généralement, une situation d'apprentissage collective selon (Faerber, 2005) peut appartenir à l'un des types suivants :

1. La situation – problème

2. *Le débat*
3. *Le projet*
4. *La résolution de problème*
5. *L'étude de cas.*
6. *L'analyse critique*
7. *La cyber-enquête*
8. *Exercices*

Un TP est une situation qui pourrait bien prendre une des formes présentées ci-dessous, mais l'aspect pratique, consistant en la manipulation d'objets pédagogiques avec des instruments spécifiques (au domaine de l'apprentissage) à l'intérieur d'un laboratoire conçu à cet effet en vue d'observer un comportement, produire un effet, assimiler les concepts théoriques avec des essais/erreurs, ou réaliser un projet, favorise plutôt les types 1,3,4,5,8

La section suivante expose trois expériences de mise en ligne de TP en explicitant les principes sur lesquels ses expériences ont été réalisées, et en montrant les problèmes que peut engendrer cette mise en ligne et les éventuelles propositions d'amélioration.

III.3.3. Typologie de situations (TP) [15]

Les objets utilisés, appelés « objets de tp » peuvent être physique, virtuels, réels ou hybrides.

Un TP peut s'organiser autour de :

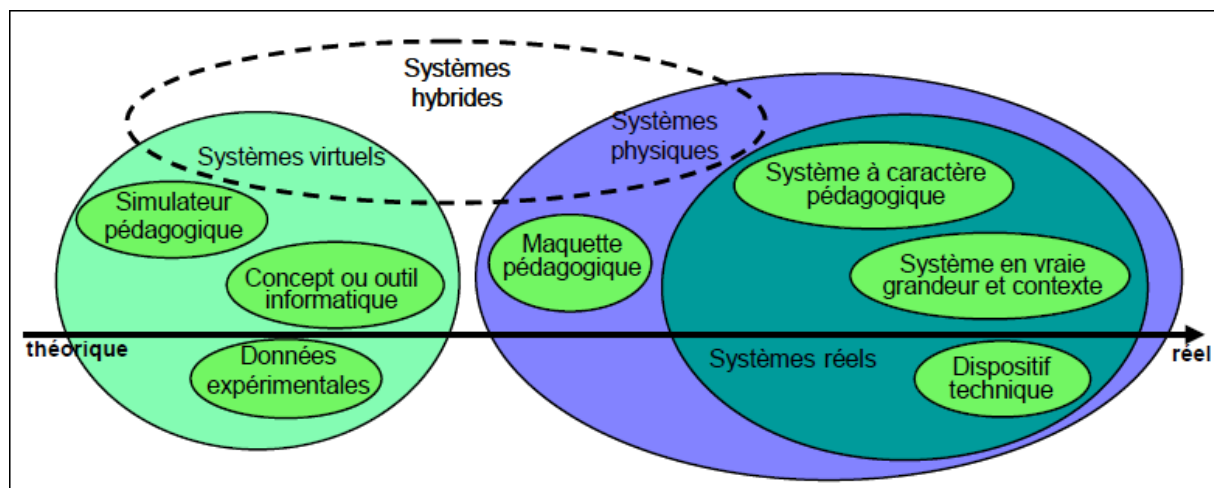


Figure II.8- Typologie de situations (TP)

- **un système en vraie grandeur et contexte** : (unité industrielle de production, atelier, magasin,..)

Généralement, il s'agit de former du personnel qualifié à la conduite de ces unités. Ce cas est rarissime, ces systèmes ne peuvent pas être utilisés dans un tel contexte pour plusieurs raisons :

- La plupart des installations fonctionnant en continu, des séances de formation dédiées seraient synonymes d'heures d'immobilisation de l'installation donc d'une perte de production inenvisageable.
- Risque de voir une conduite hésitante provoquer des alarmes dangereuses est très élevée et risque d'être économiquement désastreuse.
- Le but principal de l'apprentissage n'est pas seulement de conduire l'installation en situations normales et stationnaires mais également d'apprendre à faire face aux situations de dysfonctionnement de la plus fréquente à la plus rare, situations qui n'apparaîtraient pas forcément lors d'un fonctionnement courant.

- **un système réel à caractère pédagogique** :

Il s'agit d'un « système réel » (souvent reformé) déconnecté de son contexte d'origine pour être dédié à une formation quelconque.

Parmi les inconvénients majeurs :

- La perte du contexte (50% de l'intérêt pédagogique)
- Le coût élevé de l'acquisition initiale de l'installation et sa maintenance.
- La difficulté de faire évoluer les technologies employées. Sans ce renouvellement, représentatif de l'évolution du système réel, la formation n'est plus performante.

- **un simple dispositif technique** :

Cette catégorie inclut l'ensemble des équipements de mesures utilisées en chimie, biologie, mécanique...etc.

Exemple : chimie (pH-mètre, agitateur,...), mécanique (dynamomètre,..)

- **une maquette pédagogique** :

Il s'agit du modèle réduit d'un dispositif réel. Ce modèle réduit à une réalité physique (Magasin vertical, four industriel,...) reproduisant l'ensemble ou une partie du système initial. Il est constitué de capteurs, d'actionneurs et de briques d'assemblage.

Cette maquette peut être considérée comme le simulateur du système réel.

➤ **un simulateur pédagogique :**

Il s'agit d'un « programme qui imite partiellement le comportement d'un système réel ou abstrait, et qui inclut un certain contrôle pédagogique de l'apprenant ».

Il peut s'agir d'une simulation d'une expérimentation de laboratoire ex :(chute libre en physique), un procédé industriel (métallurgie), un moyen de transport (avion, train,...), un phénomène naturel ou un système (dangereux, trop long, trop rapide,..)

Le but principal d'une simulation pédagogique est l'apprentissage, de ce fait la modélisation du système peut être simplifiée ou exagérée pour faire ressortir les aspects pédagogiquement les plus intéressants.

➤ **l'analyse des données expérimentales :**

Généralement, ces données expérimentales (réelles) sont issues d'instruments sophistiqués et coûteux (comme en astronomie), nécessitant une longue période d'acquisition (climatologie, océanographie), nécessitant des manipulations dangereuses (physique nucléaire) ou même le fruit de la recherche scientifique (géologie, biologie). Ces données peuvent faire l'objet d'une analyse (mise en évidence de facteurs, diagnostic, traitement,...) et peuvent être utilisées pour valider des modèles théoriques acquis lors des cours magistraux ou TD.

➤ **la mise en application d'un concept ou outil informatique :**

TP portant sur les langages de programmation (java, c, ...), les systèmes d'exploitation (Windows, Unix,...) ou l'utilisation d'outils de modélisation (Merise, UML,..).

Limites des TPs

Les travaux pratiques classiques présentent quelques limitations intrinsèques :

- **Le temps**: les tranches horaires sont limitées dans l'emploi du temps et les rotations de TPs (du fait du faible nombre de systèmes accessibles comparé au nombre d'élèves) peuvent ne pas avoir lieu dans l'ordre le plus approprié pour certains groupes. En outre, les contraintes des emplois du temps font que certains TPs arrivent avant le cours magistral.

- **La documentation** : dans le pire des cas, le sujet est donné au moment de la séance et est parfois indisponible en dehors des séances de TP.

- **L'équipement** : certains TPs nécessitent un équipement lourd et onéreux que l'on ne peut pas dupliquer.

III.4. Les travaux pratiques et l'informatique [9]

L'utilisation de l'informatique dans le cadre des travaux pratiques s'est banalisée.

Outre la puissance de calcul et la capacité élevée de stockage des données, l'une des caractéristiques les plus importantes de l'ordinateur est la diversité de ses usages. Lors un TP peut être utilisé comme outil de calcul, de traitement de données, de support multimédia, ou même comme un assistant à l'expérimentation (avec la possibilité de récupération automatique des données).

III.5. Les travaux pratiques à distance (télé-TP)

Le principal objectif des travaux pratiques (TP classique qu'on peut trouver dans les lycées ou une école) est de confronter les connaissances des apprenants à la réalité.

Une activité des travaux pratiques en ligne désigne un moyen fiable pour rendre les tps étendu; afin d'y être accéder à distance via internet.

III.5.1. Définition d'un télé-TP [5]

Par analogie à la téléformation (formation à distance), nous utiliserons le terme téléTP pour désigner les travaux pratiques à distance.

La distance indique la dispersion géographique et/ou temporelle des acteurs humains et matériels (apprenant, tuteurs, instruments de laboratoire, maquettes pédagogiques,...) impliqués dans un système de téléTP.

Les travaux pratiques à distance ou téléTP constituent un pan de l'enseignement à distance qui n'a été que peu abordé dans les recherches scientifiques actuelles. Pourtant ce type d'enseignement est indispensable dans les disciplines scientifiques et techniques et répond à un véritable besoin : les installations industrielles lourdes et onéreuses ne peuvent ni être déplacées, ni être dupliquées. Or, il faut qu'à un moment donné de sa formation, l'apprenant soit confronté à leur pilotage en vrai grandeur.

En ce qui concerne les *téléTP*, une dimension didactique et générique vient s'ajouter aux problèmes classiques engendrés par la conduite à distance. Ce vide relatif est d'autant plus préoccupant que les autres contextes d'apprentissage (télé-cours, classe virtuelle, intranet pédagogique, jeux d'entreprise en ligne, ...) sont de mieux en mieux maîtrisés, mais il est vrai qu'ils n'engendrent pas la même complexité.

III.5.2 Facteurs de développements

L'augmentation des formations scientifique et techniques en ligne et l'accroissement de la communauté de chercheurs sur le thème de téléTP, nous laisse penser que les téléTPs seront dans quelques années une réalité à l'instar des autres composants d'un environnement de téléformation (téléCours, téléTD).

Cette conviction s'appuie sur trois constats :

- **Contexte favorable :**

Les dispositifs didactiques ou technologiques industriels sont souvent extrêmement coûteux [Bayard et al, 2002] et posent le problème de leur opérationnalité et de leur renouvellement. Il faut pourtant garantir la qualité des formations professionnelles proposées. L'une des solutions à ce dilemme est la mutualisation de ces ressources coûteuse.

- **Dimension pédagogique :**

L'émergence de nouveaux standards (internet, serveurs puissants..) dans la e-formation répondant à des besoins de structuration et de réutilisation de contenu et d'interopérabilité entre plates-formes.

Facilite aussi l'intégration des plates-formes de téléTP au sein de plates-formes de e-formation, ou téléCour et téléTD coexistant déjà.

- **Dimension technologique :**

L'un des obstacles au pilotage d'un système physique est la nécessité d'avoir des échanges rapides ; alors la montée actuelle des débits des réseaux informatiques dont internet et l'accroissement de la puissance des ordinateurs participent à la résolution de ce problème.

III.5.3. Les apports des téléTP

Les travaux pratiques classiques présentent des limitations intrinsèques :

- **Les temps :**

Les tranches horaires sont limitées dans l'emploi du temps ainsi les tps peuvent ne pas avoir lieu dans l'ordre le plus approprié. Dans ce cas, le bon sens voudrait que ces TP soient pédagogiquement reconstruits (remplacement d'une pédagogie de l'expérimentation des connaissances à une pédagogie de découverte), ce qui n'est jamais le cas.

- **La documentation :**

Dans le pire des cas, le sujet est découvert en début de séance même distribué avant la séance, il est rarement lu avant par les élèves.

- **L'équipement :**

Certains TP nécessitent un équipement lourd et onéreux que l'on ne peut pas dupliquer (exemple typique : centre d'usinage numérique).

Les téléTPs dépassent certains de ces limites en facilitant :

- **Sur le plan économique :**

- Le partage d'instruments et d'équipement lourds et coûteux entre institutions.
- L'accès aux instruments d'un « laboratoire » n'importe quand et de n'importe où.

- **Sur le plan pédagogique :**

- L'illustration de téléCours ou téléTD pour des apprenants distants ne pouvant pas accéder à des dispositifs réels (équivalent à une démonstration),
- L'intégration d'outil (simulateurs, réalité virtuelle,..) apportés par les TIC,

- L'ouverture à des utilisateurs autre que la population d'origine : scénarios pédagogiques adaptés à des niveaux scolaires différents (du lycée à l'école d'ingénieur) et même au grand public en dehors des créneaux réservés aux élèves.
- Les démonstrations mettant en valeur l'équipement et le savoir-faire de l'établissement auprès d'un grand public

III.5.4. Les limites des téléTP

Malgré leurs atouts, les télétp souffrent de nombreux problèmes qu'il faut s'attacher à résoudre. En effet, la mise à distance d'une manipulation engendre des difficultés en terme de :

- **Commandabilité :**

À distance, on ne peut plus directement agir sur le système qu'à travers un clavier, une souris, un joystick,...L'utilisation de techniques plus avancées telles que celles utilisées dans la réalité virtuelle peut résoudre en partie ce problème.

- **Observabilité :**

L'observation sensorielle (visuelle, auditive, gustative, etc) du système piloté et de son environnement est également dégradée. Alors les technique de la réalité augmentée (superposition de données virtuelles sur des images d'un environnement réel) et de retour haptique (restitution tactile ou retour d'effort) ont ici un terrain d'application fertile (riche).

- **Sécurité :**

Nous distinguons deux niveaux de sécurité :

Le premier est la sécurité du système informatique gérant les manipulations ; Une bonne expertise en sécurité informatique est nécessaire pour éviter un usage malveillant de systèmes réels pouvant provoquer leur immobilisation.

Le deuxième niveau est la sécurité du matériel télémanipulé, notamment celui qui est accessible en permanence. Une des solutions possibles consiste à placer un agent technique, à côté des manipulations en cas de besoin, ce qui limite leur usage à la présence de cette personne et réduit les plages horaires disponibles.

III.5.5. Typologie de situations (téléTP)

Par opposition aux TP en présentiel, nous distinguons trois formes de téléTP ; Les téléTP sur des systèmes virtuels (laboratoire virtuels), les téléTP sur des systèmes physiques (laboratoire distants) et les téléTP sur des systèmes distants- virtuels (laboratoires hybrides).

III.6. Les téléTP sur des systèmes virtuels (laboratoire virtuels)[15]

Les premiers téléTP étaient fondées sur l'utilisation de simulations ; l'usage de logiciels de simulation tels que Matlab/Simulink et LabView a contribué à harmoniser les solutions proposées ; Cependant ces logiciels sont difficiles à intégrer dans un processus pédagogique.

Les simulateurs informatiques offrent la possibilité de manipuler virtuellement des systèmes :

- à taille non humaine, dans des disciplines telles que l'astronomie, la physique quantique,...
- trop dangereux pour l'homme
- trop coûteux
- dont les phénomènes se déroulent sur une durée dépassant celle de la session d'apprentissage (échanges thermiques en génie civil)
- dont les phénomènes sont trop rapides pour pouvoir être observés (électricité, lumière).

III.7. les téléTP sur des systèmes physiques (laboratoire distants)

Les laboratoires distants résultent simplement d'une mise à distance de plate-forme de TP physique. Ils permettent l'accès, le pilotage et la commande à distance du système à télémanipuler (instruments d'un laboratoire par exemple) via des liaisons informatiques (en particulier internet).

Suivant le type de manipulation ou de contrôle distant, nous distinguons quatre catégories de laboratoires :

1. les laboratoires distants fournissant un accès à des données expérimentales :

Il s'agit de données mises à la disposition à la foi des chercheurs scientifiques et de la communauté éducative. Le but pédagogique est de comparer les données à des informations issues de la théorie. Dans ce cas les apprenants et les tuteurs n'ont aucune relation directe avec les instruments de mesure. Ce qui revient à avoir une confiance aveugle sur les conditions de collectes (problème de la qualité des données).

2. les laboratoires distants offrant des moyens de télémesure : contrairement à la catégorie précédente, les données ici sont récupérées en temps réel, via internet.

3. les laboratoires distants proposant un type de manipulation : les utilisateurs ont la possibilité d'interagir de façon limitée avec le système à télémanipuler. Il s'agit d'exécuter des commandes simples. Le « Bradford RoboticTelescope » de l'université de Grande-Bretagne est un exemple de cette catégorie.

Tout utilisateur, inscrit sur le site Internet, peut effectuer des observations à l'aide de ce télescope. Ce système n'est pas destiné seulement au grand public et aux écoliers, mais aussi aux chercheurs.

2. les laboratoires distants offrant un ensemble de manipulation

L'utilisateur a la possibilité de contrôler entièrement l'expérimentation à distance et de recevoir des réponses en temps réel.

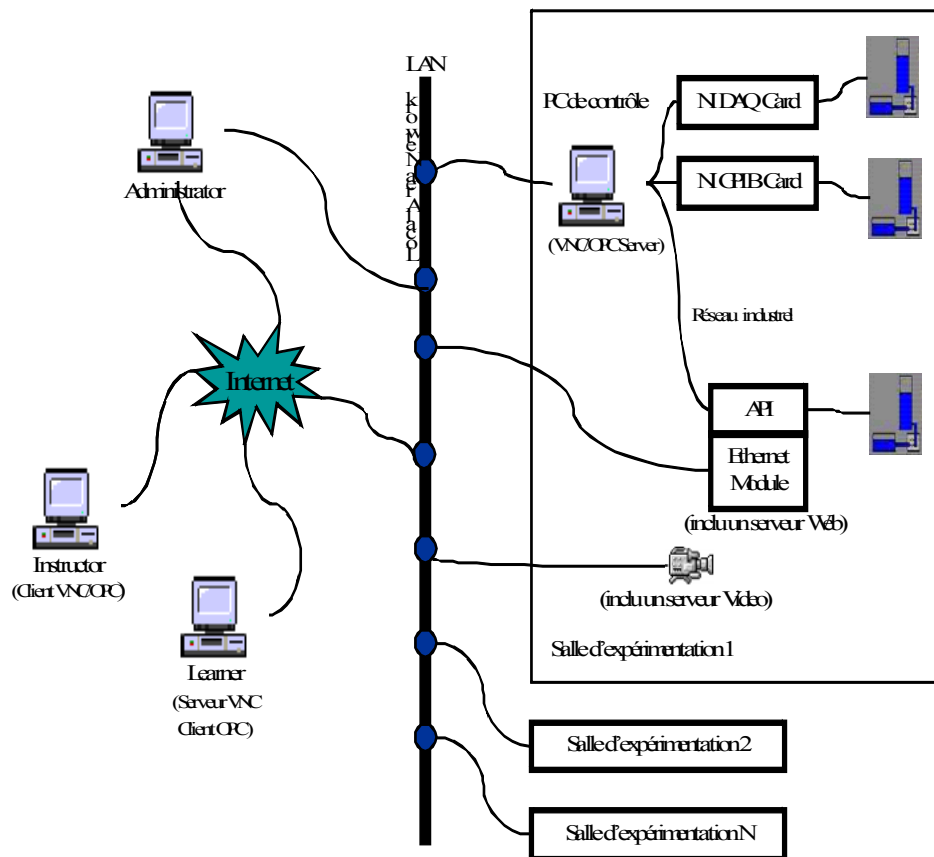


Figure II.9- Architecture type pour les laboratoires distants

III.8. les télé-TP sur des systèmes distants- virtuels (laboratoires hybrides)

Comme son intitulé l'indique, cette dernière forme de télé-TP est une combinaison des deux. Prenons l'exemple simple d'une chaîne de production. Il est possible à mettre à distance (laboratoire distant) une partie de la chaîne (une machine par exemple) et de compléter le reste de la chaîne par un simulateur informatique (laboratoire virtuel). Ainsi, les laboratoires hybrides peuvent profiter, à la fois, des avantages des laboratoires distants et des avantages de laboratoires virtuels. [15]

III.9. Laboratoire virtuel et laboratoire virtuel à distance

Compte tenu de l'évolution rapide des Technologies de l'Information, la création de nouvelles méthodes d'enseignement s'impose afin de permettre aux étudiants de maîtriser les contours difficiles du programme d'enseignement. De nos jours, dans le souci d'amélioration

de l'enseignement des Technologies de l'Information et de la Communication (TIC), plusieurs Universités et Institutions voudraient mettre à la disposition des étudiants une solution basée sur les environnements virtuels d'étude pour leur permettre d'acquérir des aptitudes pratiques suite à la théorie acquise dans l'environnement classique d'enseignement.

III.9.1. Laboratoire virtuel

La réunion a adopté un large perspectif et défini un laboratoire virtuel comme "un espace de travail électronique pour la collaboration à distance et l'expérimentation dans la recherche ou dans d'autres activités créatives, en vue de générer et de diffuser des résultats au moyen de technologies partagées de l'information et de la communication".

Un LV se distingue d'un laboratoire "réel" (LR) ou "traditionnel". Cependant, un LV n'est pas conçu comme un substitut ou un concurrent d'un LR ; En revanche, les LV peuvent être des extensions de LR et ouvrent de nouvelles opportunités qui ne peuvent se réaliser entièrement au sein d'un LR à un coût abordable.

Un LV est caractérisé par des simulations de scénarios expérimentaux modulaires conçus pour être mis en œuvre à partir d'un ou plusieurs ordinateurs. Les modèles mathématiques sont mis à contribution pour s'approcher le plus près de la crédibilité des simulations représentant des concepts théoriques ou des dispositifs réels. Dans certains cas de figure, il n'est pas possible de simuler des scénarios expérimentaux ou le comportement de dispositifs réels. C'est le cas où les modèles mathématiques sont trop complexes, par manque de disponibilité de puissance de calcul ou quand les délais des traitements informatiques sont longs.

III.9.2. Laboratoire à distance (laboratoire en ligne)

Traditionnellement, les expériences sont conduites par une personne présente dans un laboratoire physique. Grâce aux avancés technologiques en informatique, en réseau et en communication durant les vingt (20) dernières années, deux nouveaux moyens d'expérimentation ont été découverts à savoir : l'expérimentation virtuelle et l'expérimentation à distance.

Dans l'expérimentation virtuelle, le cadre des expériences est un modèle mathématique sur un ordinateur. Dans l'expérimentation à distance, il y a un laboratoire physique, mais l'expérimentateur se situe hors du laboratoire et fait ses expériences sur un réseau de

communication (Internet ou Intranet). La Figure suivante illustre un cadre de laboratoire à distance.

Les laboratoires en ligne sont aussi des environnements informatiques distribués et flexibles qui permettent à un apprenant d'effectuer, par les réseaux, des expériences seul ou en collaboration avec d'autres participants dans un contexte d'apprentissage à distance. Les participants sont des acteurs qui jouent chacun des rôles pendant une séance interactive en mode synchrone. Les laboratoires en ligne permettent le partage de ressources matérielles et d'expertises.

III.9.2.1. Architecture d'un labo en ligne :

On peut voir comment, à travers une plateforme de télécommunications appropriée, dotée d'un système de gestion dédié et d'un certain nombre d'interfaces de logiciels et de matériels, il est possible à des utilisateurs géographiquement dispersés d'accéder à des ressources disponibles dans plusieurs sites distants. Il est ainsi possible de miser sur l'accessibilité des ressources matérielles et humaines distribuées pour réaliser des expérimentations non possibles dans un contexte local. Le schéma de la Figure suivante permet d'illustrer les diverses formes que peut prendre un laboratoire en ligne. Par exemple, un usager au site A peut manipuler le bras robot articulé du site B ; étant donné que le bras robotisé articulé est composé de dispositifs motorisés et qu'ils sont commandés à distance à partir du poste A1 de télécommande, nous dirons qu'il s'agit là d'un contexte de télé-laboratoire. Également, à partir du poste de visioconférence (B) et de télémesure, il est possible d'effectuer des travaux pratiques à distance en mettant en jeu les équipements réels distants tels que l'analyseur de spectre et l'oscilloscope branché à un montage électronique, tous au site A.

Les autres acteurs d'un scénario pédagogique donné, avec les privilèges d'accès appropriés, pourraient intervenir parce qu'ils auraient, pendant la séance de travail pratique, les outils de collaboration en réseau adéquats pour agir ou observer les activités d'apprentissage à distance. La configuration permet également d'effectuer des séances de travaux pratiques basés uniquement sur des simulations distribuées ou sur la visualisation d'animation ou de produits multimédias diffusés par les réseaux informatiques.

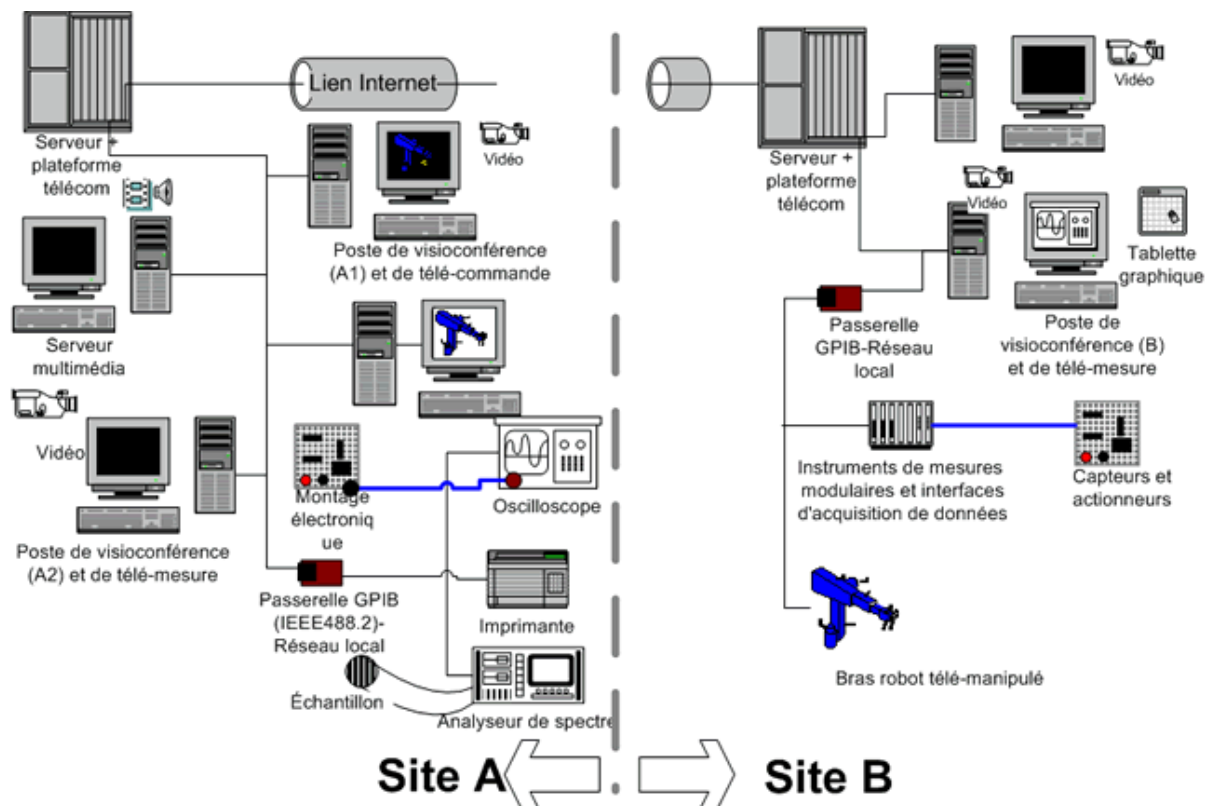


Figure II.10- Laboratoire en ligne

III.10. Exemple de laboratoire électronique (e-lab)

III.10.1. Le projet LVEST (Laboratoires virtuels pour l'éducation en science et technologie)

Le projet LVEST expérimente la diffusion en temps réel de séquences vidéo filmées ou simulées, la simulation interactive d'expériences, la télémanipulation et la télémessure. De plus, le projet offrait des outils d'acquisition et d'analyse de données ainsi que des outils de communication entre apprenants et avec le formateur, ce qui permet l'échange de données expérimentales et la communication synchrone en modes texte, audio et vidéo.

Il existe dans le même projet LVEST plusieurs laboratoires en ligne, à savoir :

- **Le laboratoire virtuel de génie électrique** : est un environnement de simulation tenant compte des technologies novatrices émergentes relatives au développement des produits commerciaux en instrumentation

L'intégration de ces produits dans un contexte de formation à distance posait des défis technologiques de taille, dont le choix du langage décrivant des dispositifs interconnectés, leurs adresses et leurs fonctions.

- **Le laboratoire de Génie Informatique** est un environnement de simulation visant l'apprentissage de la démarche expérimentale et des concepts scientifiques à l'œuvre dans les expériences. Le laboratoire de réseautique comprend des expériences de divers types réalisables individuellement ou en équipe et une interface de navigation entre les domaines, groupes d'expériences. Expériences et espaces de ceux-ci, offrant de plus un accès vers les outils d'analyse des données et les services de communication éventuellement offerts par la plateforme.

III.10.2. Le projet VITELS (laboratoire virtuel de télécommunications pour la Suisse)

Fait partie du projet Campus Virtuel Suisse (CVS). Le but du projet VITELS est de développer un cours modulaire accompagné d'un ensemble de travaux pratiques effectués sans contrainte de présence physique dans les laboratoires. Les étudiants n'ont besoin que d'une connexion Internet. Au total, sept modules et une infrastructure d'authentification et d'autorisation sont développés et maintenus par les différents instituts et universités suisses impliqués dans ce projet.

Le Télé-laboratoire Linux est une expérience pratique en ligne du projet VITELS permettant d'installer et de configurer Linux « Mandrake » sur une machine réelle et à distance. Cette distribution de Linux propose un logiciel graphique (DrakX) (Figure suivante) pour l'installation de Linux. L'interface graphique de DrakX est facile d'installation.

Il est possible, pendant l'activité, de retourner à tout moment à une étape précédente. Le type d'installation peut être prédéfini selon le niveau de connaissance de GNU/Linux de l'utilisateur. Le modèle de laboratoire en ligne mis au point reproduit l'environnement DrakX le plus fidèlement possible pour permettre aux apprenants et aux formateurs de se retrouver facilement au moment d'une vraie installation sur un poste de travail.

La Figure suivante représente un écran de l'une des étapes de l'installation, soit le partitionnement du disque dur. À gauche, les différentes phases d'installation sont identifiées et indiquent l'étape à laquelle se trouve l'utilisateur pendant la procédure. Les étapes accessibles et celles non accessibles par l'utilisateur sont également présentées sur l'interface.



Partition du disque dur d'une machine

Le Télé-laboratoire Linux permet deux types d'installations. Le premier est une installation réelle faite à distance sur une machine réelle et vierge et le deuxième type est une installation virtuelle. Ce dernier permet à un apprenant de simuler une installation, au cas où toutes les machines du laboratoire distant seraient occupées. D'autres fonctionnalités permettent de répliquer par la suite l'installation virtuelle sur une machine réelle.

III.10.3. Le projet eMersion

L'École Polytechnique Fédérale de Lausanne (EPFL), par l'intermédiaire du projet eMersion du Département de génie mécanique, fait des recherches dans le domaine des laboratoires en ligne. Ce projet intervient dans le domaine du génie mécanique, dans des modules de formation flexible en ateliers de simulation et en laboratoires d'expérimentation. Le projet eMersion a pour but de réintroduire l'apprentissage par l'erreur dans la formation des ingénieurs en développant et en flexibilisant les possibilités de simulation en réalité virtuelle et d'expérimentation en laboratoire. Un corollaire est de favoriser la rénovation du schéma actuel d'études grâce à la mise à disposition de moyens alternatifs de conduite d'activités pratiques, essentiels tant sur le site qu'à domicile.

III.10.4. Exemples de laboratoires en ligne hébergés dans l'environnement de diffusion ECLATE

ECLATE est un Environnement de Cours et de Laboratoires Assistés par les Technologies de l'Enseignement développé au centre de recherche LICEF sous la direction de Hamadou Saliah Hassane.

IV. Conclusion

L'objectif de cette étude consiste à développer un modèle générique de télé-TP conforme aux standards actuels de l'e-learning permettant les échanges et la réutilisation de scénarios pédagogiques de télé-TP.

Ce chapitre nous a permis de définir les activités de « travaux pratiques », qui servent de référence pédagogique pour les téléTP. Nous avons également présenté comment l'informatique s'est immiscée dans ce mode d'enseignement et les nouvelles formes de travaux pratiques qui en ont émergé. Partant de cet état des lieux, nous avons pu définir précisément ce que représente le terme « téléTP » dans la littérature scientifique et les caractéristiques générales de ce mode particulier d'enseignement à distance.

Les laboratoires virtuels décrits dans ce chapitre offrent des solutions open source qui peut fournir tous les instruments nécessaires pour les exercices pratiques dans le domaine des TIC ; et permettre aux étudiants d'acquérir des aptitudes pratiques suite à la théorie acquise dans l'environnement classique d'enseignement.

CHAPITRE III

I. Introduction

Dans ce chapitre, nous nous proposons de modéliser une plate-forme télé-tps basée sur le web services. Pour ce faire, nous avons adopté le formalisme UML car il permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation, tout en utilisant son extension pour le web.

Les principaux modules constituant notre plate-forme, seront :

- ✓ La consultation d'objets pédagogiques (cours, exercices, évaluation...);
- ✓ La communication entre les différents acteurs de la plate-forme ;
- ✓ Le suivi des apprenants ;
- ✓ Gestions des acteurs ;
- ✓ Accès aux différents services de télé-tps ;

II. Définition de l'UML [14]

UML (*Unified Modeling Language*, que l'on peut traduire par "*langage de modélisation unifié*") est un langage graphique de modélisation des données et des traitements.

De plus, il est une notation permettant de modéliser un problème de façon standard. Il est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE, principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson.

UML est à présent un standard défini par l'Object Management Group (OMG), et est devenu désormais la référence en termes de modélisation objet.

Pour plus de détails, (voir l'annexe B).

III. Objectif du projet

Derrière cette application qu'on a réalisée un objectif bien précis a été visé, est d'implémenter des tps en tant que service dans une plate-forme d'enseignement à distance e-Learning pour que ces tps soient accessibles de n'importe quelle plate-forme et sa grâce aux messages XML.

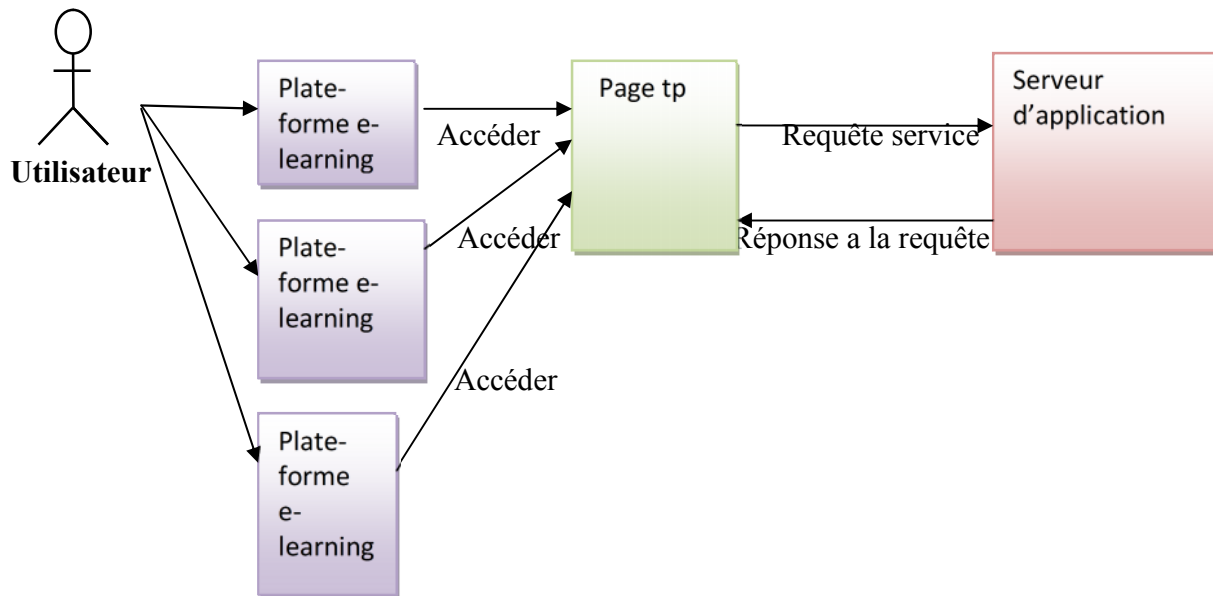


Figure III.1 : Objectif de notre projet

III.1. Identification des acteurs et des besoins

III.1.1. Identification des acteurs et leurs besoins fonctionnels

Dans notre plate forme on distingue quatre principaux acteurs, pour chaque acteur on a identifié les fonctionnalités lui permettant d'accomplir son rôle dans le système :

➤ Administrateur :

L'administrateur représente l'acteur chargé de réguler l'activité sur le système, il assure un ensemble d'actions telles que :

- Accès direct à la base de données de la plate-forme ;
- Il s'occupe entre autre de l'inscription finale des acteurs dans la plate-forme (validation des prés inscriptions) ;
- Gestion des comptes utilisateurs (apprenants et formateurs) ;
- Gestion des sessions de prés inscription des acteurs ;
- Gestion de la pédagogie.

➤ Formateur :

Le formateur est la personne chargée de suivre les apprenants durant le processus d'apprentissage et cela en :

- Assurant le suivi des tps ;
- Rechercher et faire un tp ;
- Communiquant et animer ses apprenants via la messagerie interne, le forum ;

➤ Apprenants :

L'apprenant est toute personne qui suit un cours dans son propre espace réservé, et il s'agit aussi de la personne effectuant un télé-TP, seul ou au sein d'un groupe, avec ou sans tuteur dans le but d'atteindre les objectifs pédagogiques visés par le télé-TP:

- Consulter et télécharger des cours, des exercices ;
- Communiquer avec les autres apprenants et son formateur via les outils collaboratifs (forums, messagerie interne) ;
- Faire un TP à distance ;
- Rechercher un TP.

➤ Auteur :

Appelé aussi concepteur de matériels pédagogiques, il est chargé de concevoir et de réaliser les scénarios pédagogiques de télé-TP de préférence via un outil auteur spécifique.

- recherche d'un TP ;
- (forums, messagerie interne) ;
- ajout d'un TP ;
- suppression d'un TP ;

➤ Visiteur :

Le visiteur est toute personne qui visite le site sans être ni un apprenant ni un formateur.

L'ensemble des actions qu'un visiteur peut effectuer sont :

- Se préinscrire en tant qu'apprenant ;
- Se préinscrire en tant que formateur ;
- Contacter l'administrateur pour d'éventuelles informations ;

- Contacter l'administrateur pour d'éventuelles informations ;
- Naviguer dans le site pour avoir une idée des possibilités des formations.

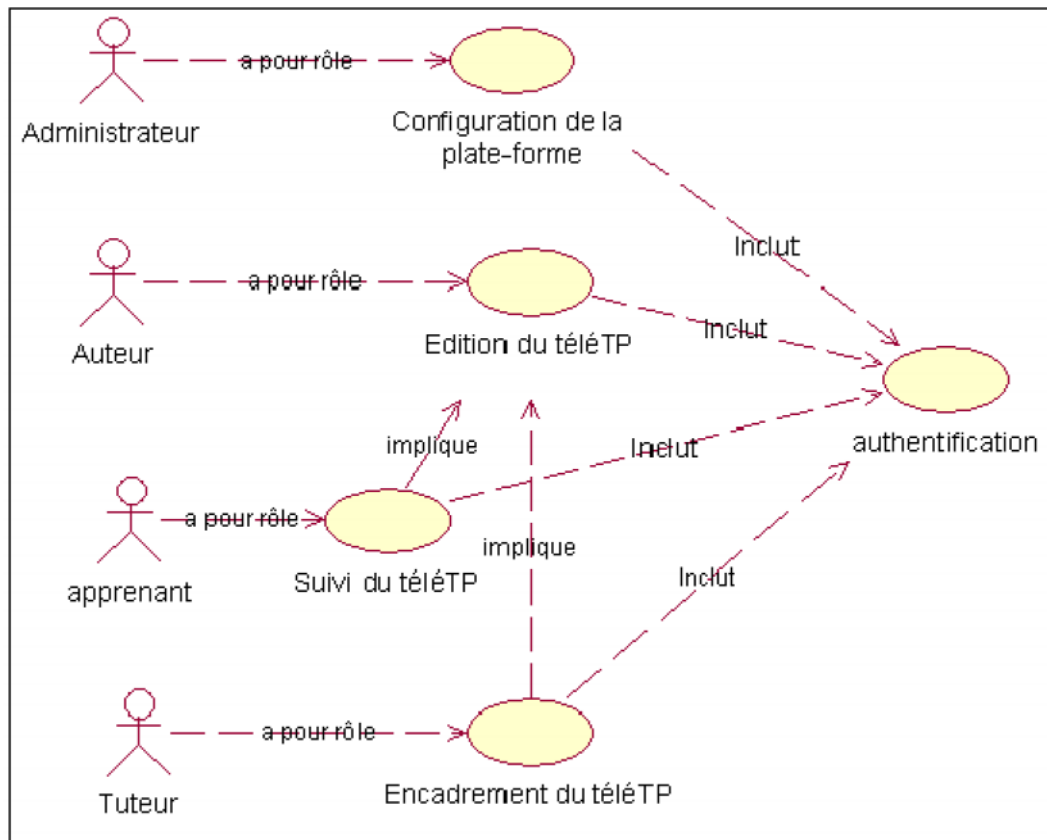


Figure III.2 : Les acteurs humains et les principaux cas d'utilisation

IV. Diagrammes représentatifs

IV.1 Le diagramme de contexte de l'application

Le diagramme de contexte permet de spécifier le nombre d'instances d'acteurs connectés à ce système à un moment donné.

La définition des besoins fonctionnels des acteurs présentée dans le paragraphe (1.1) nous permet de dégager le diagramme de contexte suivant :

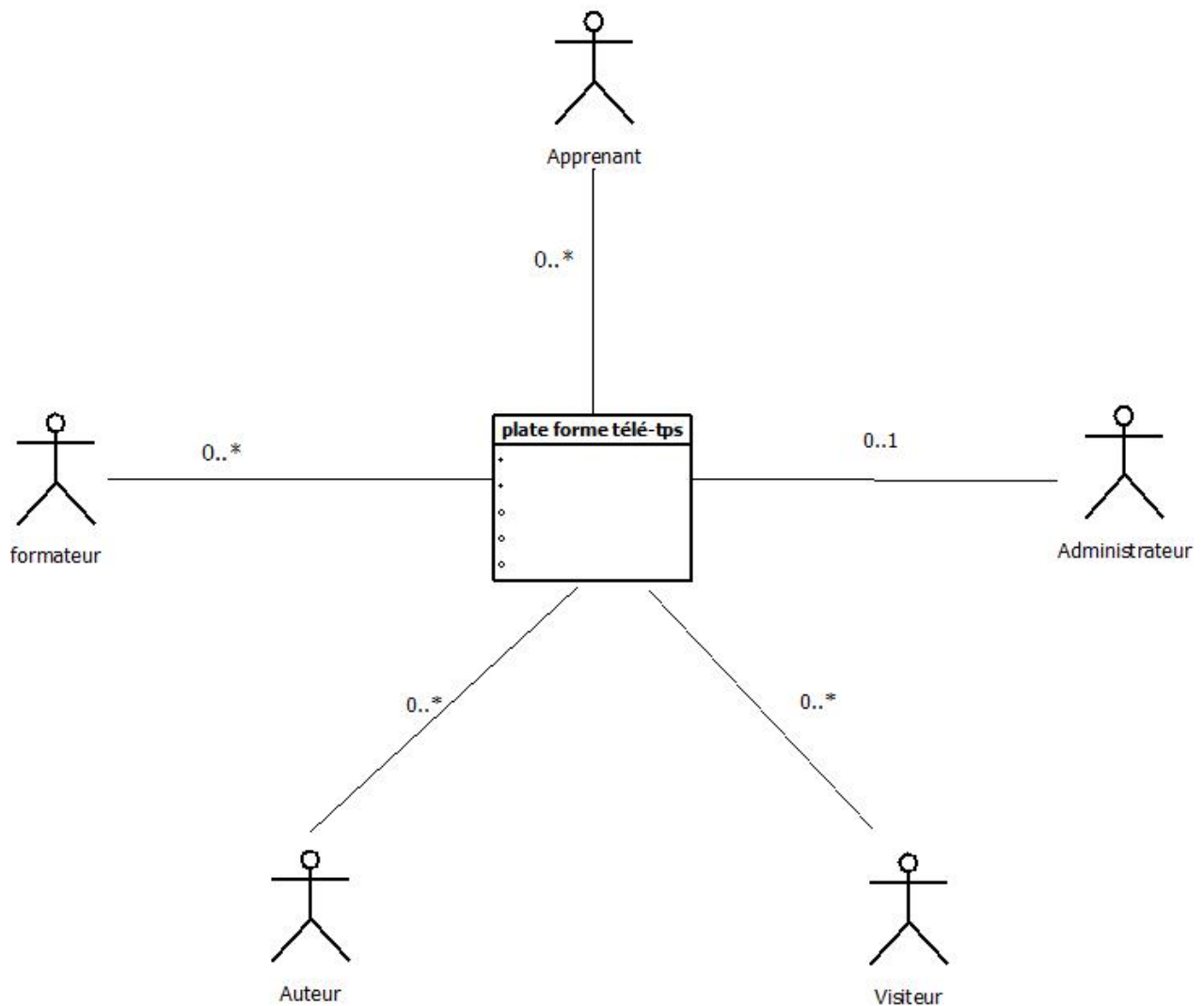


Figure III.3 : Diagramme de contexte de l'application.

IV.2 Diagramme des cas d'utilisation

Un cas d'utilisation décrit un ensemble de séquences dans lequel chaque séquence représente l'interaction des éléments qui se trouvent à l'extérieur du système (ses acteurs) avec le système lui-même.

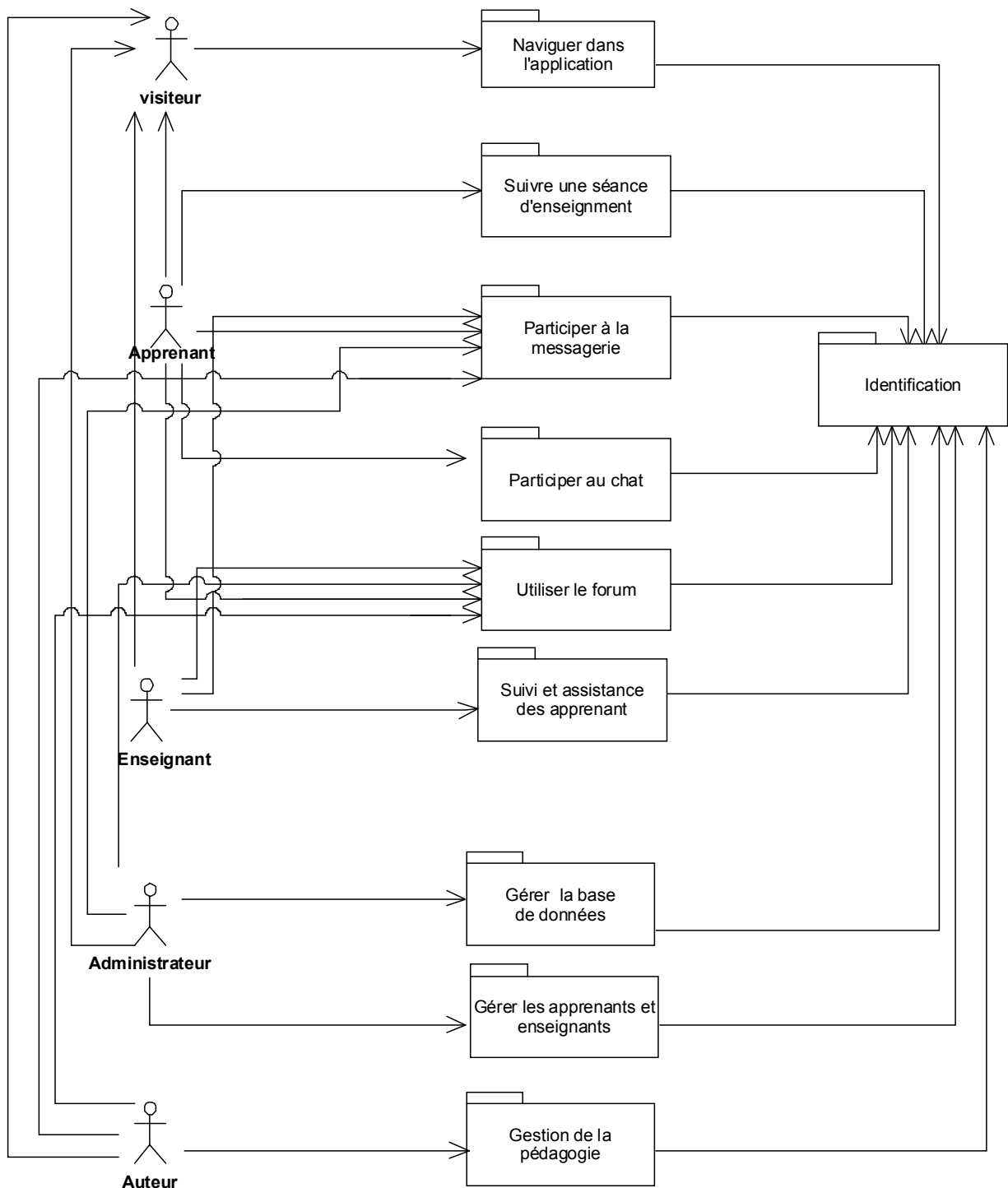


Figure III.4 : Structuration des activités en package.

IV.3 Détermination des diagrammes de cas d'utilisations

Etant donné que les acteurs et les packages de leurs activités sont identifiés, nous sommes en mesure de modéliser les cas d'utilisation de ces acteurs, qui sont les détails des packages définis dans la figure III.3. Un cas d'utilisation décrit ce que l'utilisateur veut fondamentalement faire avec

le système. Il s'agit d'une représentation macroscopique des interactions entre acteurs et le système fondée sur une sémantique "faible" qui rend ce modèle aisément compréhensible par les utilisateurs.

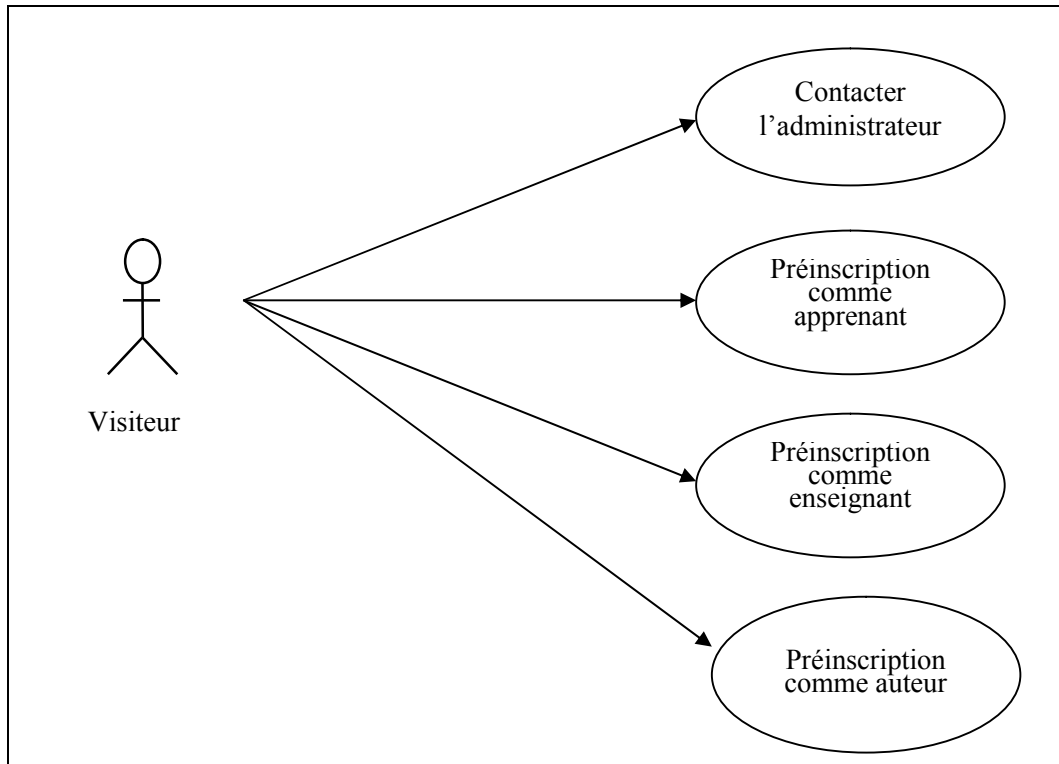


Figure III.5 : Diagramme de cas d'utilisation pour le paquetage « naviguer dans l'application »

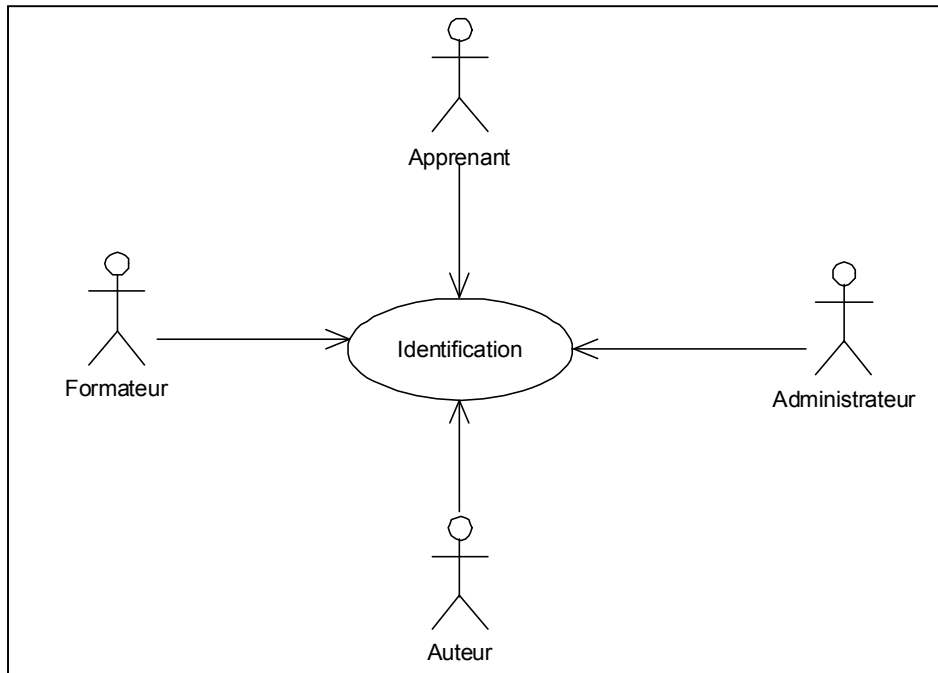


Figure III.6 : Diagramme de cas d'utilisation pour le paquetage « identification ».

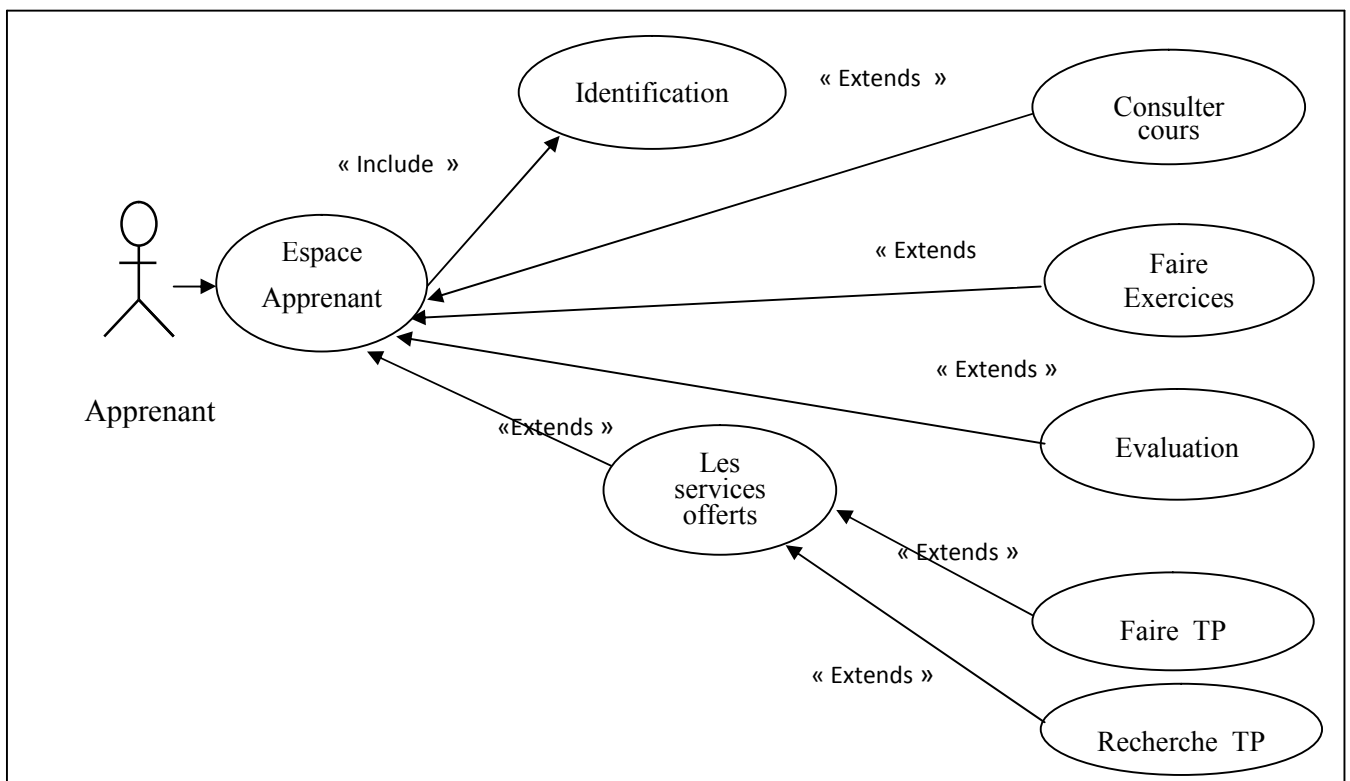


Figure III.7 : Diagramme de cas d'utilisation pour le paquetage

« Suivre une séance d’enseignement »

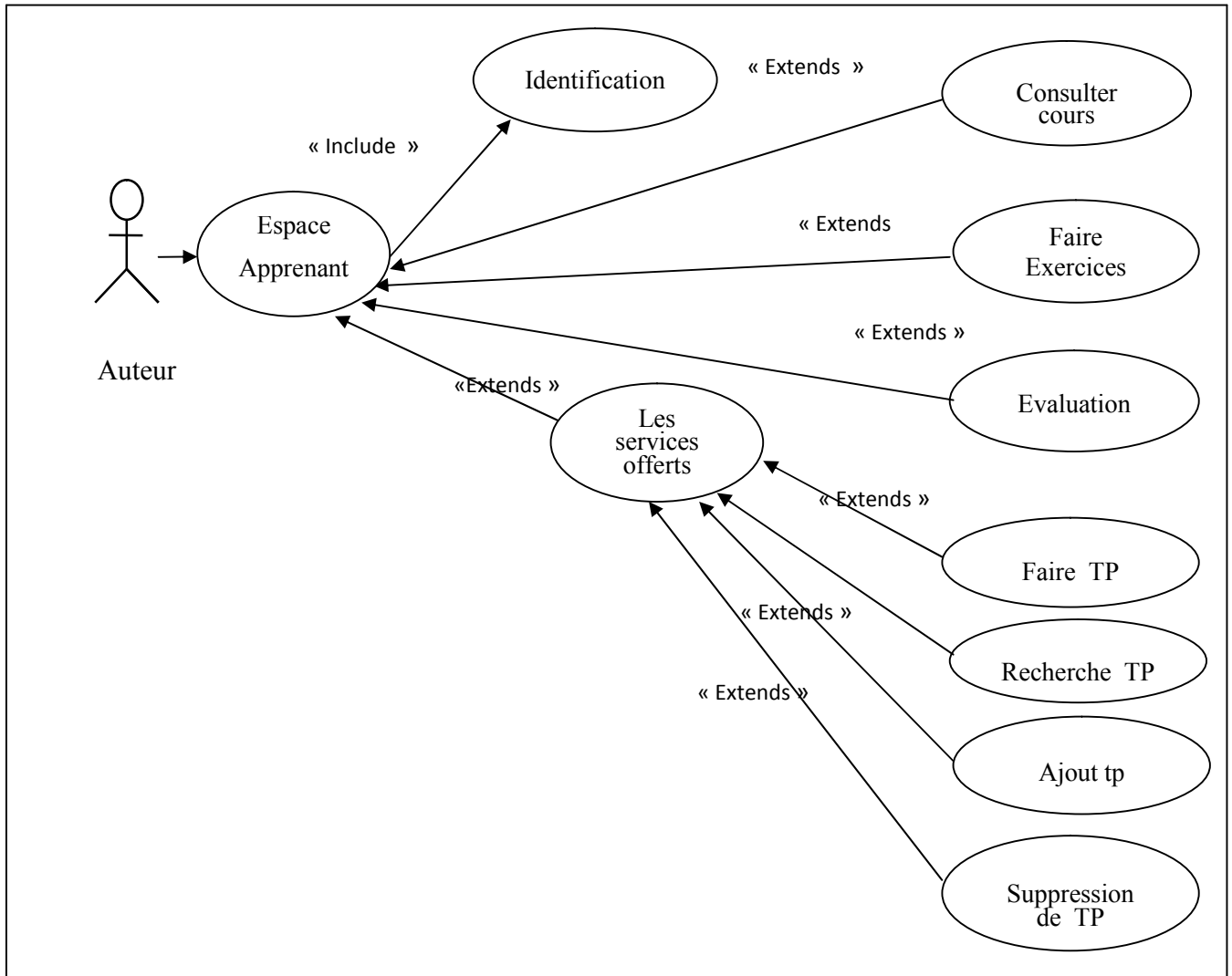


Figure III.8 : Diagramme de cas d’utilisation pour le paquetage

« Suivre une séance d’enseignement »

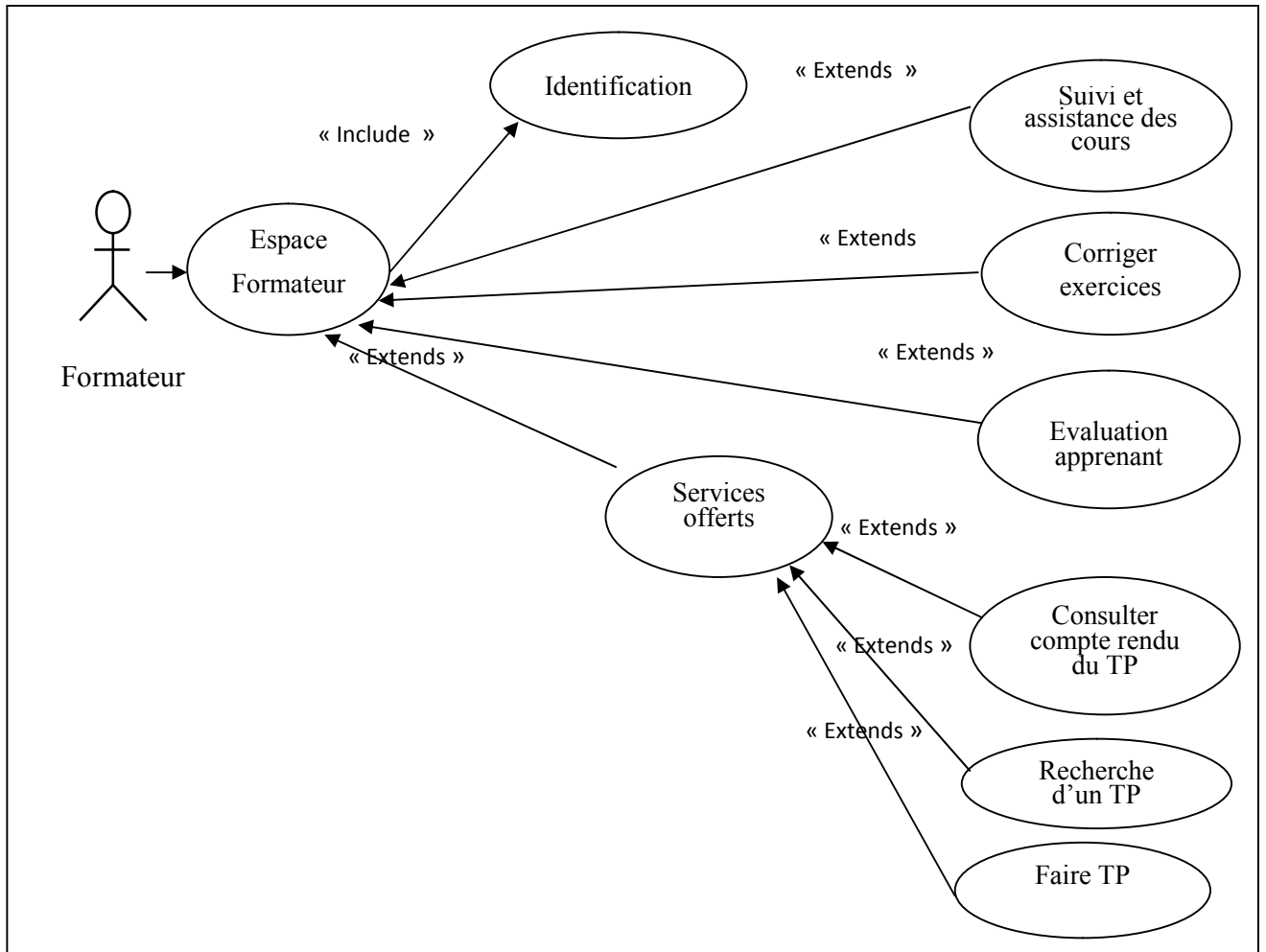


Figure III.9 : Diagramme de cas d'utilisations pour le paquetage «suivi et assistance des apprenants».

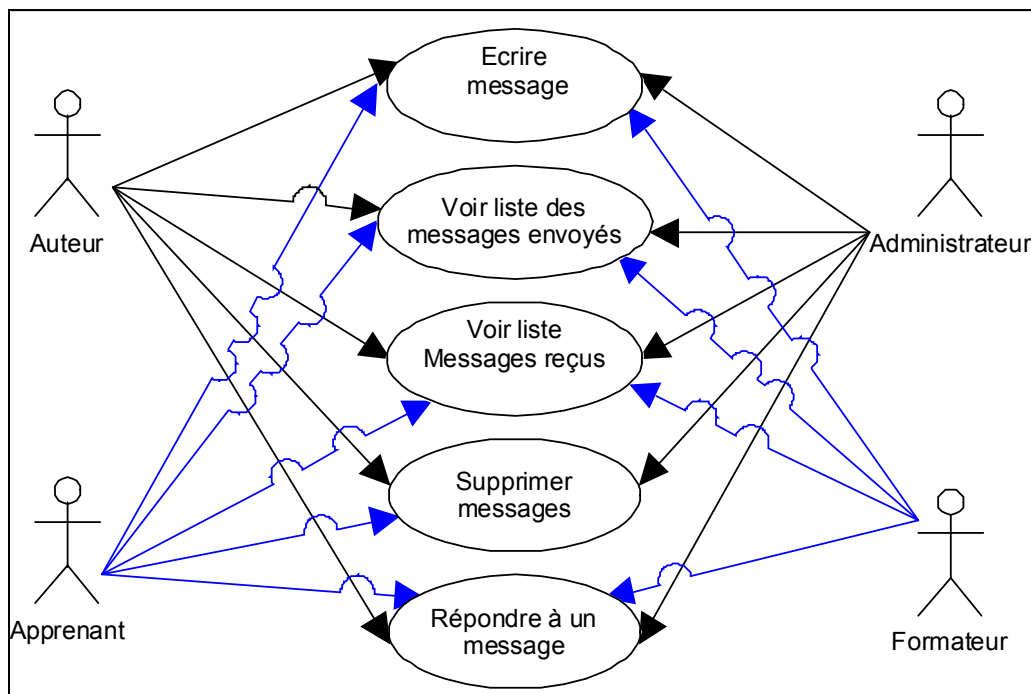


Figure III.10 :Diagramme de cas d'utilisations pour le paquetage «participer à la messagerie».

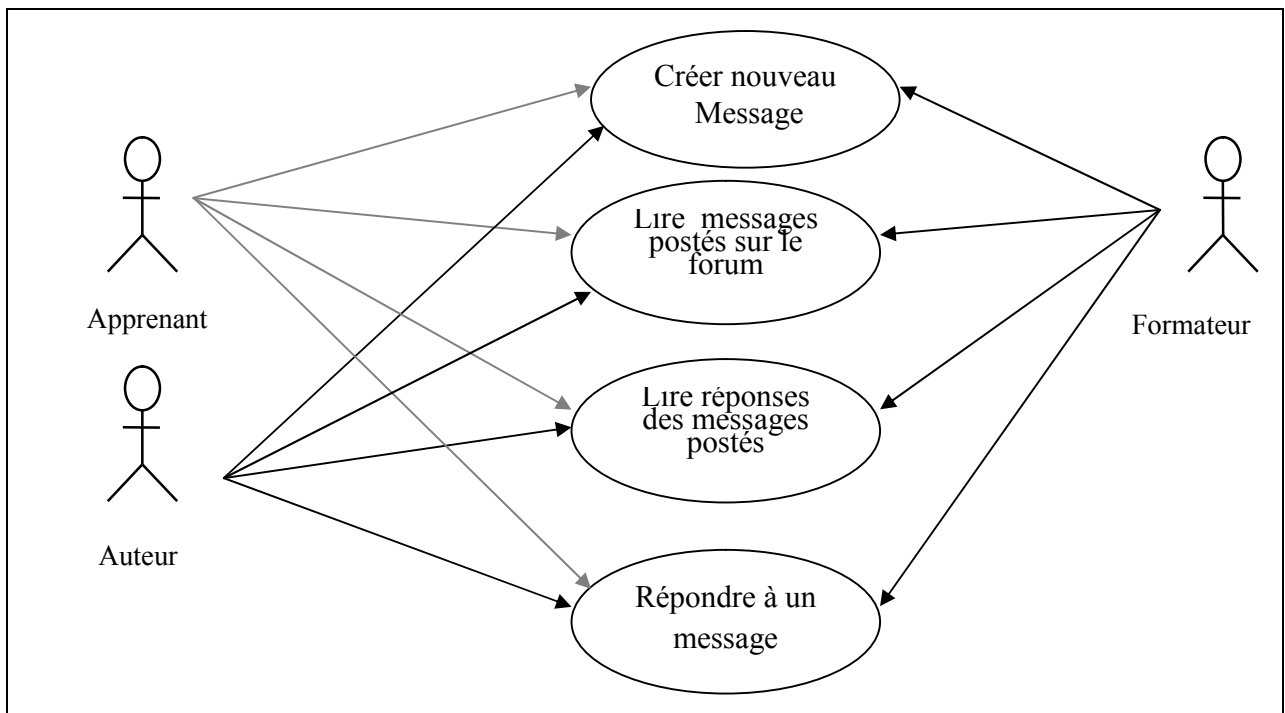


Figure III.11 : Diagramme de cas d'utilisation pour le paquetage « utiliser le forum ».

IV.4. Description textuelle des cas d'utilisation

Définition d'un scénario : un scénario décrit comment le système doit être utilisé, c'est un chemin logique traversant les cas d'utilisation (c'est l'exécution pas à pas d'un cas d'utilisation).

- Pré inscription :

Titre : Pré inscription

Acteur : visiteur.

- Scénario nominal 1 :

1. L'utilisateur atteint le site ;
2. Le système affiche la page d'accueil ;
3. L'utilisateur clique sur le lien préinscription ;
4. Le système affiche le formulaire de pré inscription ;
5. L'utilisateur remplit le formulaire de pré inscription et clique sur le bouton « envoyer » pour le valider ;
6. Le système fait les contrôles suivants :
 - ✓ Vérifie le remplissage des champs du formulaire obligatoires ;
 - ✓ Vérifie dans la base de données si le login saisi n'est pas pris par un autre apprenant ;
 - ✓ Vérifie que le premier mot de passe saisi est identique au mot de passe de confirmation ;

- Enchaînement alternatif 1 :

A : S'il y a des erreurs dans le formulaire.

Le scénario nominal démarre à l'étape « 6 » et enchaîne :

7. Le système signale les erreurs et propose de les corriger ;

Le scénario nominal démarre à l'étape « 5 ».

- Enchaînement alternatif 2 :

B : S'il y a des erreurs dans le formulaire

Le scénario nominal démarre à l'étape « 6 » et enchaîne :

8. Le système affiche toutes les données saisies par l'utilisateur et propose de confirmer ou de mettre à jour les données du formulaire ;

➤ Enchaînement alternatif 2.1 :

B.1 : Si l'utilisateur valide le formulaire.

L'enchaînement alternatif 2 démarre à l'étape « 7 » et enchaîne :

9. Le système enregistre l'utilisateur autant qu'apprenant ou formateur préinscrit, et lui envoyant un message contenant le login et le mot de passage de messagerie interne pour qu'il puisse confirmer son inscription en cas d'acceptation.

➤ Enchaînement alternatif 2.2 :

B.2 : l'utilisateur décide de mettre à jour le formulaire.

Le scénario nominal démarre à l'étape « 4 » avec un formulaire rempli des données à mettre à jour.

- Identification

Titre : Identification

Acteurs : Apprenant, enseignant, auteur ou administrateur

➤ Scénario nominal 2 :

1. L'utilisateur atteint sa page d'identification ;
2. Le système répond en affichant le formulaire d'identification en lui proposant de créer un compte, de s'identifier pour accéder à son propre espace ou encore de récupérer son mot de passe en cas d'oubli ;

➤ Enchaînement alternatif 1 :

A : L'utilisateur possède un compte :

L'enchaînement alternatif 1 démarre à l'étape « 2 » et enchaîne :

3. L'utilisateur remplit le formulaire d'identification en tapant son login et mot de passe et valide en cliquant sur le bouton « connexion » ;
4. Le système compare les données saisies par l'utilisateur avec celles de la base de données.

➤ Enchaînement alternatif 1.1 :

A.1 : Login et mot de passe corrects.

Le scénario nominal démarre à l'étape « 4 » et enchaîne :

5. Le système charge l'espace approprié à l'utilisateur.

➤ Enchaînement alternatif 1.2 :

A.2 : Login ou mot de passe incorrect

Le scénario nominal démarre à l'étape « 4 » et enchaîne :

5. Le système redirige vers la page d'identification avec le message : « Echec de connexion vérifiez votre login et mot de passe » ;

➤ Enchaînement alternatif 2 :

B : L'utilisateur clique sur le lien « créer »

L'enchaînement alternatif 2 démarre à l'étape « 2 » et enchaîne :

3. le système rédige vers la page d'accueil en incluant le formulaire de préinscription ;

➤ Enchaînement alternatif 3 :

C : L'utilisateur clique sur le lien « oublier ».

L'enchaînement alternatif 1 démarre à l'étape « 2 » et enchaîne :

3. Le système affiche la page de récupération de mot de passe avec un champ de formulaire invitant l'utilisateur à taper son login ;

4. L'utilisateur saisit son login et valide le formulaire ;

5. Le système compare les données transmises avec celles de la base de données ;

➤ Enchaînement alternatif 3.1 :

C.1 : Login correct.

L'enchaînement alternatif démarre à l'étape « 5 » et enchaîne :

6. Le système récupère le mot de passe et le login de la messagerie interne de l'utilisateur dans la base de données et envoie le mot de passe à ce login ;
7. Le système redirige vers la page d'accueil avec le message « Votre mot de passe est envoyé avec succès à votre compte dans la messagerie interne fournie lors de votre inscription ;

➤ Enchaînement alternatif 3.2 :

C.2 : Login incorrect.

L'enchaînement alternatif démarre à l'étape « 5 » et enchaîne :

6. Le système redirige vers la page d'accueil avec le message « Votre login est incorrect ».
 - Suivre une séance d'enseignement

Titre : Suivre une séance.

Acteur : Apprenant.

➤ Enchaînement alternatif 3 :

1. Après identification, l'apprenant atteint sa page d'accueil.
2. Le système répond en affichant la page d'accueil. Sur la page se retrouvent les liens hypertextes suivants : cours, exercice, évaluation, TP.

➤ Enchaînement alternatif 1 :

A : l'apprenant décide de consulter le cours.

Le scénario nominal démarre à l'étape « 2 » et enchaîne :

3. L'apprenant clique sur le lien « cours » ;
4. Le système charge depuis la base de données le cours associé à la dernière session de consultation du cours ;
5. L'apprenant consulte son cours.

➤ Enchaînement alternatif 2 :

B : L'apprenant décide de faire un exercice.

Le scénario nominal démarre à l'étape « 2 » et enchaîne :

3. L'apprenant clique sur le lien « exercice » ;
4. Le système charge l'exercice associé ou cours ;
5. L'apprenant fait l'exercice et l'envoi à la correction ;

➤ Enchaînement alternatif 3 :

D : L'apprenant décide de consulter les résultats de l'évaluation.

Le scénario nominal démarre à l'étape « 2 » et enchaîne :

3. L'apprenant clique sur le lien « Evaluation » ;
4. Le système interroge la base de données et affiche un tableau regroupant les résultats de l'évaluation ;
5. L'apprenant consulte les résultats.

➤ Enchaînement alternatif 4 :

A : L'apprenant décide de faire le TP.

Le scénario nominal démarre à l'étape « 2 » et enchaîne :

6. L'apprenant clique sur le lien « TP » ;
7. Le système répond en chargeant la page de services de TP qui contient deux liens, un qui permet d'afficher l'annuaire et l'autre permet l'accès à l'espace de travail ;
8. L'apprenant clique sur le lien « accéder à l'espace de travail » ;
9. Le système lui affiche la fenêtre de l'espace de travail.

- Gestion des cours et exercices

Titre : Gestion du contenu pédagogique.

Acteur : Acteur.

➤ Scénario nominal 4 :

1. Après identification, l'auteur atteint sa page d'accueil ;
2. Le système répond en affichant la page d'accueil. Sur la page figure le lien hypertexte « cours, exercices, évaluation » ;
3. L'auteur clique sur le lien « cours » ;
4. Le système interroge la base de données et retourne une page contenant la liste de tout les cours ;

➤ Enchaînement alternatif 1 :

A : l'auteur décide de consulter un cours.

Le scénario nominal démarre à l'étape 4 et enchaîne :

5. L'auteur clique sur le lien « consultation » ;
6. Le système interroge la base de données et retourne une page contenant une liste de tous les chapitres constituant le cours à consulter ;
7. L'auteur sélectionne un chapitre ;
8. Le système lance le téléchargement du cours en transférant ses différents chapitres au côté client (environnement de l'auteur) ;

➤ Enchaînement alternatif 2 :

A : l'auteur décide d'ajouter un chapitre au contenu du cours.

Le scénario nominal démarre à l'étape 4 et enchaîne :

5. L'auteur clique sur le lien « ajouter » ;
6. Le système interroge la base de données et retourne une page contenant une liste de tous les cours ;

7. L'auteur « sélectionne » un cours le système affiche un formulaire proposant le nom du chapitre et son contenu ;
8. L'auteur clique sur le bouton « envoyer » ;
9. Le système interroge la base de données et retourne une page contenant une liste de tous les chapitres y compris celui ajouté et propose d'appuyer sur le bouton « valider ».

➤ Enchaînement alternatif 2 :

A : l'auteur décide de supprimer un cours.

Le scénario nominal démarre à l'étape 4 et enchaîne :

5. L'auteur clique sur le lien « supprimer » ;
6. Le système interroge la base de données et retourne une page contenant une liste de tous les cours ;
7. L'auteur sélectionne le cours à supprimer en appuyant sur le bouton « confirmer » ;
8. Le système interroge la base de données et retourne une page confirmant la suppression avec succès.

- Gérer les préinscriptions des apprenants

Titre : gérer les pré inscriptions des apprenants.

Acteur : administrateur.

➤ Scénario nominal 5 :

1. Après identification ;
2. Le système charge l'espace administrateur avec le lien « gestion des utilisateurs » ;
3. L'administrateur clique sur le lien « apprenants » ;
4. L'administrateur clique sur le lien « préinscrits » ;
5. Le système affiche une liste à plusieurs entrées de tous les candidats ayant fait une préinscription et attendant une réponse ;
6. L'administrateur clique sur une entrée pour voir le profil ;

7. Le système affiche le profil du candidat concerné et propose deux boutons « accepter » et « refuser » ;

➤ Enchaînement alternatif 1 :

A : si le profil du candidat est correct.

Le scénario nominal démarre à l'étape « 6 » et enchaîne ;

7. L'administrateur clique sur le bouton « accepter » ;
8. Le système ajoute le candidat à la table des apprenants et le supprime de la table des candidats, génère et envoie un message à l'apprenant lui confirmant son inscription ;

Le scénario nominal démarre au point « 4 ».

V. Réalisation des cas d'utilisation

Cette étape consiste à représenter des objets d'interface et de contrôle en page client et serveur. Les icônes utilisées pour représenter ces objets sont définies par l'extension UML pour le web (Voir l'annexe UML).

Dans la suite de ce paragraphe, on représentera les diagrammes de réalisation associés aux cas d'utilisation analysés précédents.

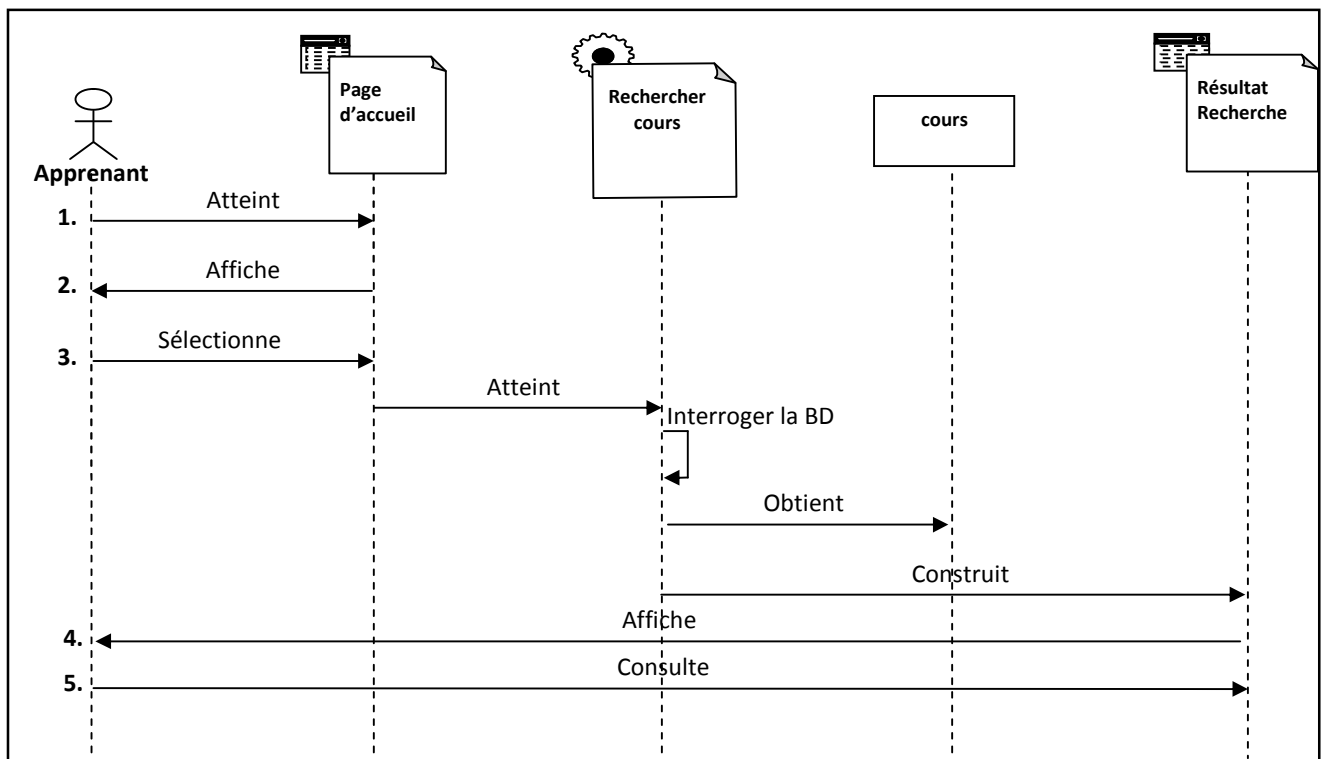
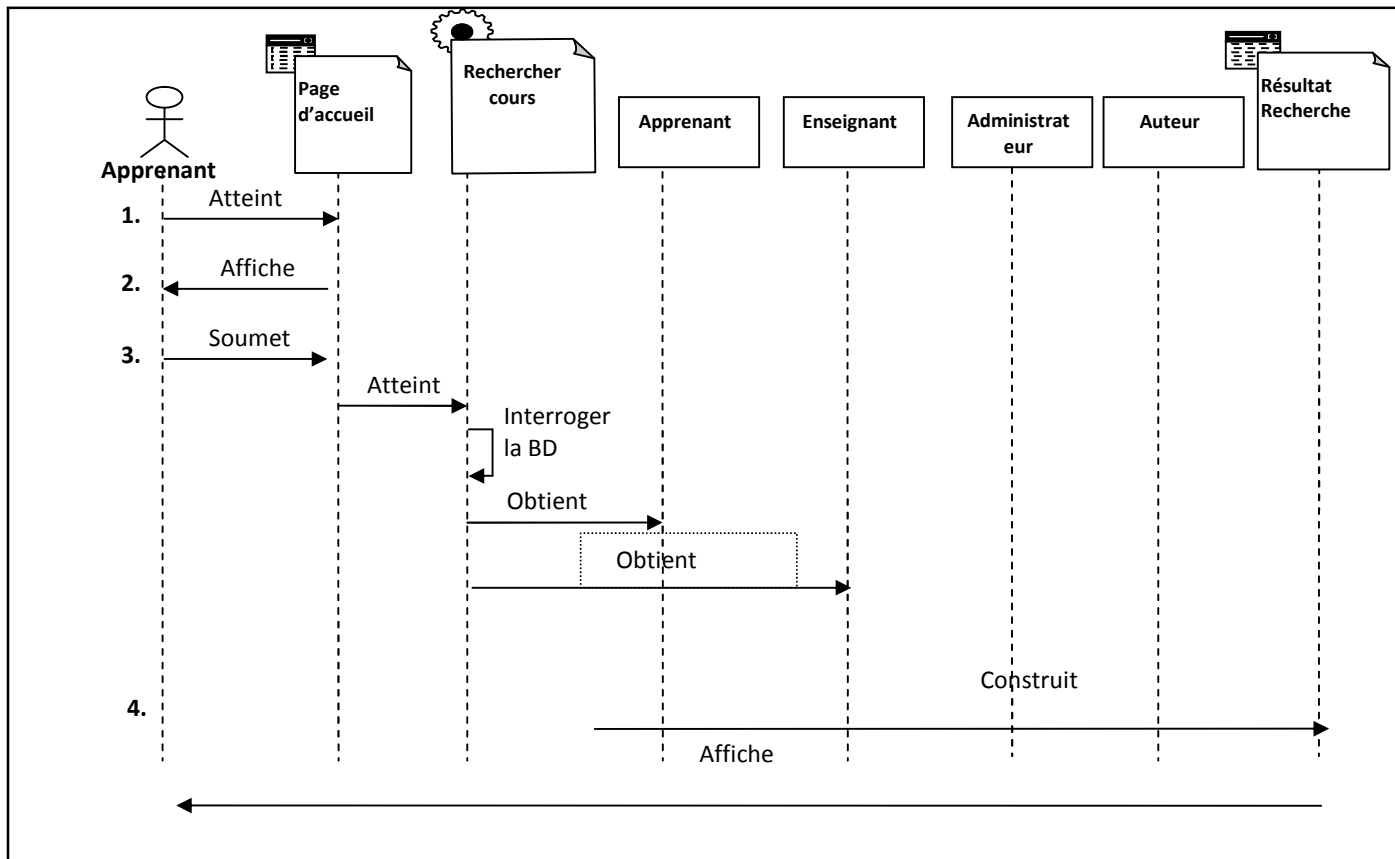


Figure III.12 : Diagramme de séquence de réalisation du cas d'utilisation « consulter cours ».

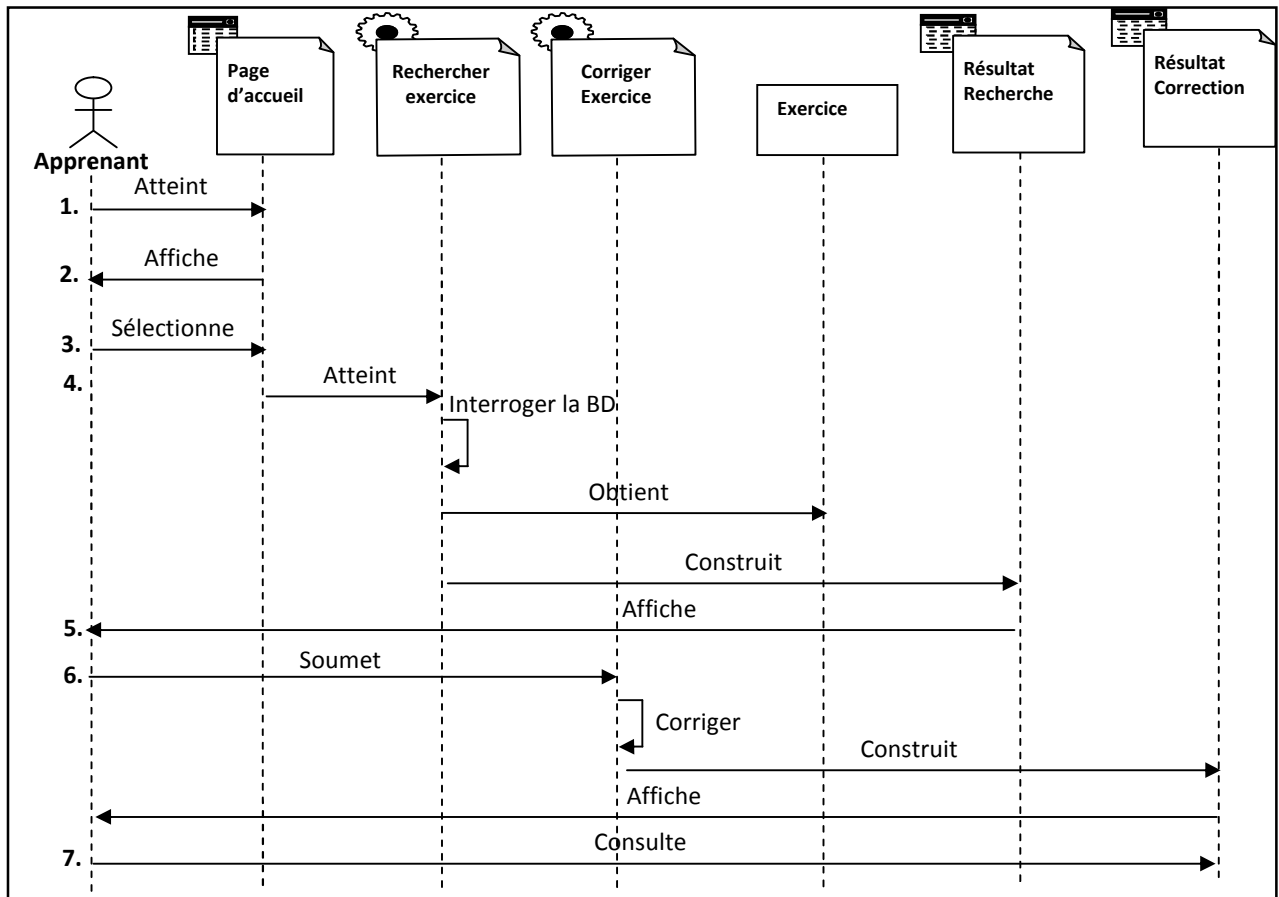


Figure III.13 : Diagramme de séquence de réalisation du cas d'utilisation « faire exercice ».

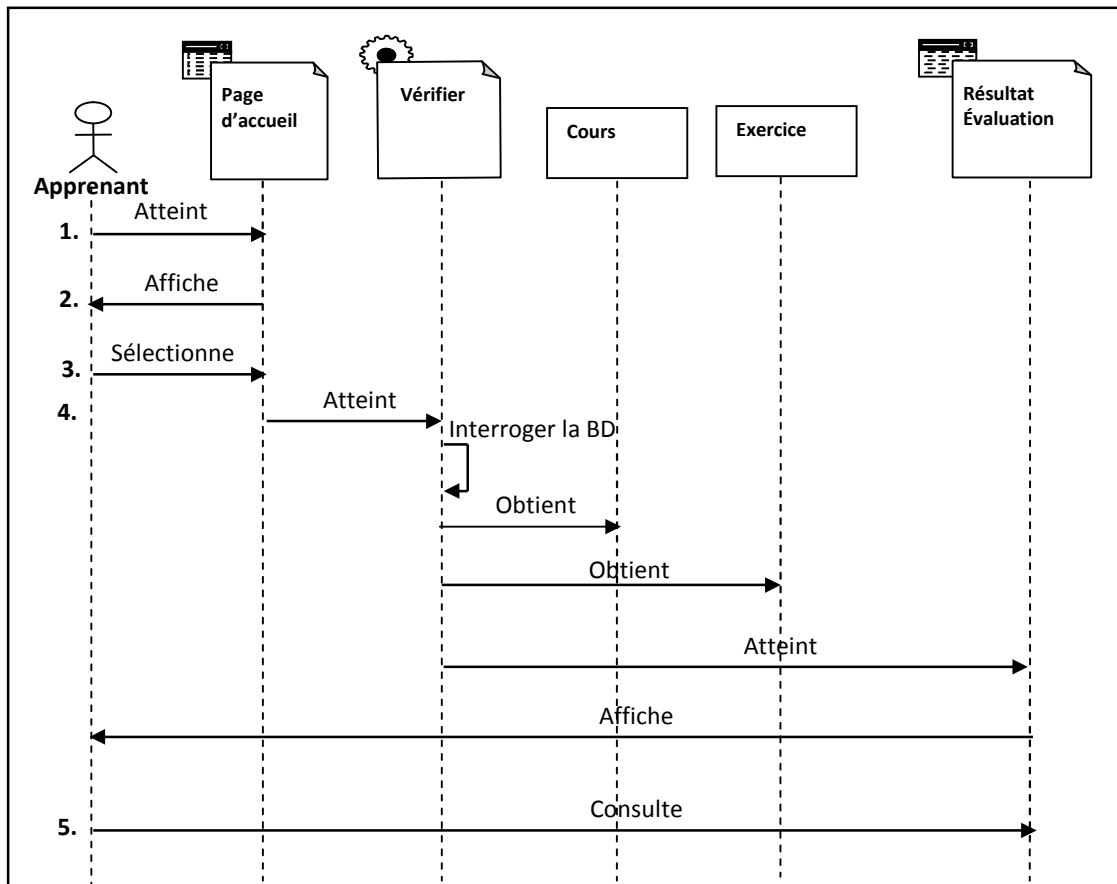


Figure III.14 : Diagramme de séquence de réalisation du cas d'utilisation « évaluation ».

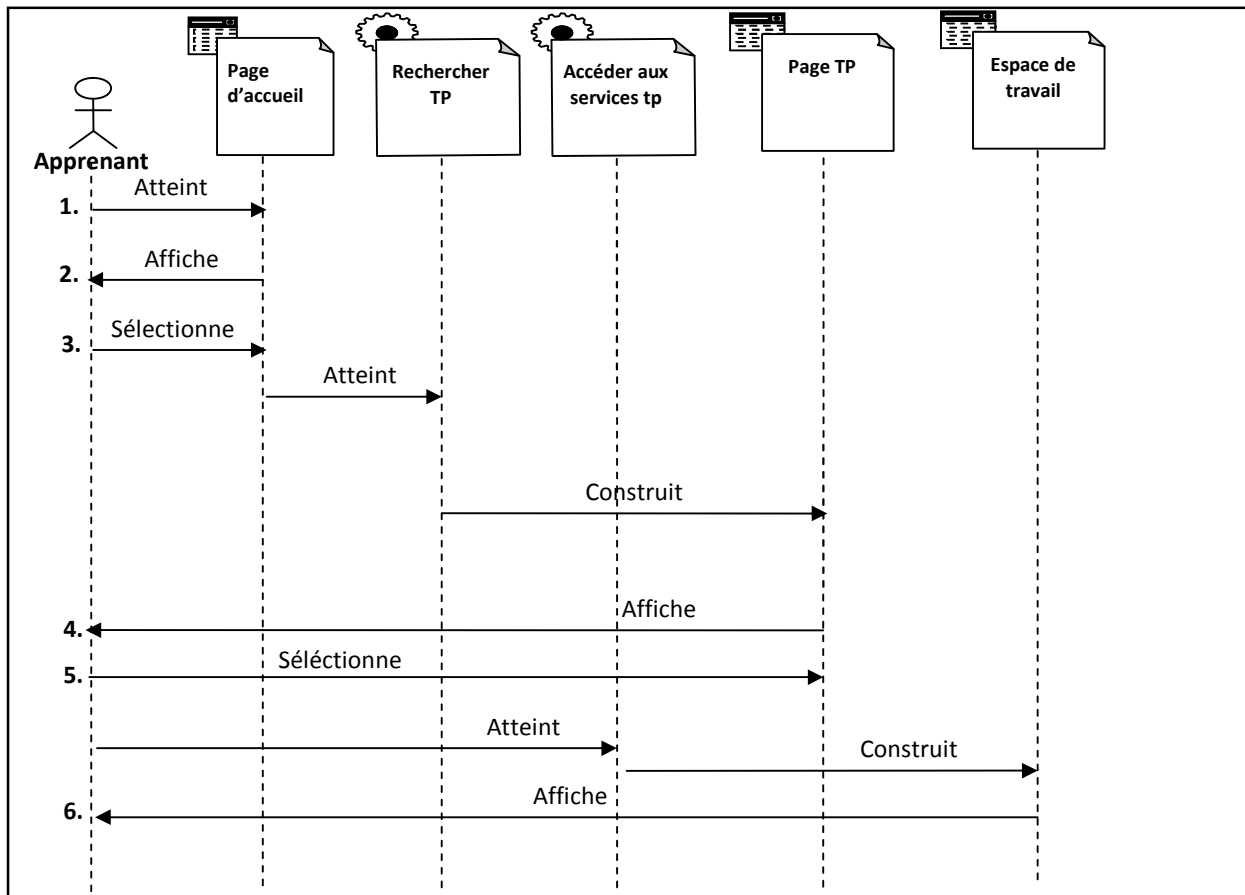


Figure III.15 : Diagramme de séquence de réalisation du cas d'utilisation « Faire TP ».

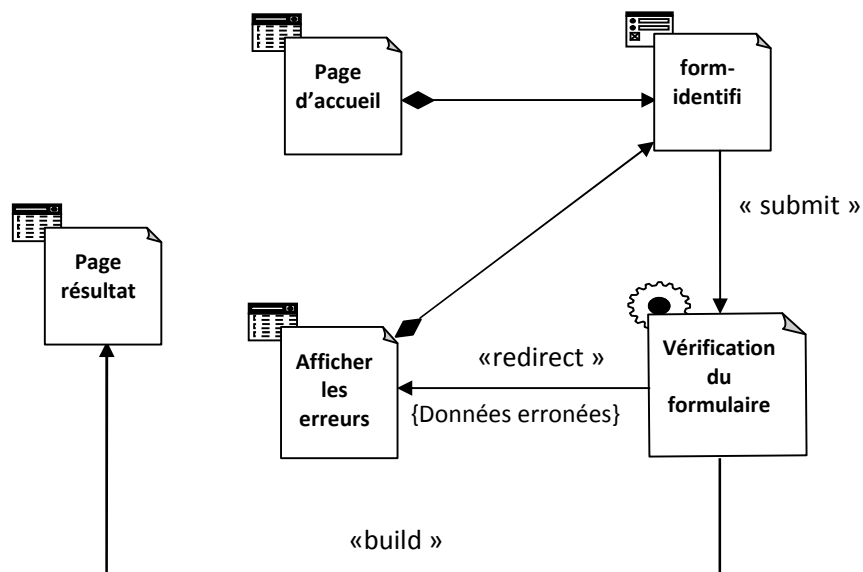


Figure III.16 : Diagramme de classe général de cas d'utilisation « identification ».

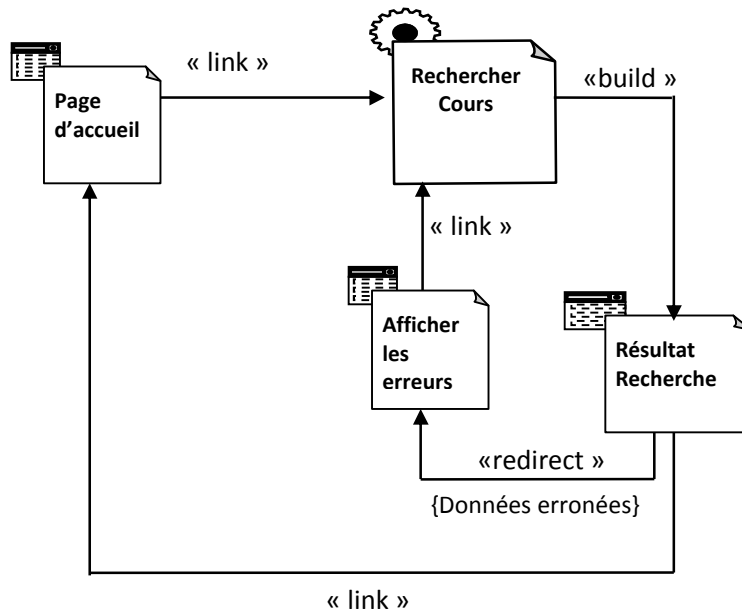


Figure III.17 : Diagramme de classe général de cas d'utilisation « consulter cours ».

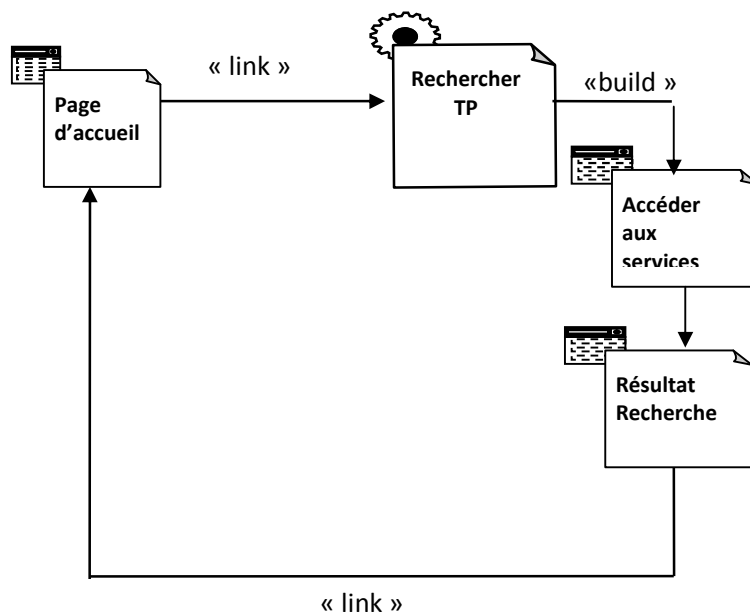


Figure III.18: Diagramme de classe général de cas d'utilisation « Faire TP ».

Figure III.19 : Diagramme de classe générale

b. le niveau logique de la base de données

Apprenant (**id_app**, nom_app, prénom_app, date_naiss_app, adresse_app, email_app, sexe_app, login_app, mdp_app).

Assister (**id_app, id_cours**).

Auteur (id_aut, nom_aut, prénom_aut, date_naiss_aut, adresse_aut, email_aut, sexe_aut, login_aut, mdp_aut, **id_cours***).

Chapitre (id_chap, intitulé_chap, nom_chap, contenu_chap, solution, id_exo*, id_eval*).

Chat (id_chat, message_chat, pseudo).

CompteRendu (id_cr, contenu_cr, id_app*).

Consulter (id_form, id_cr).

Cours (id_cours, intitulé_cours, nb_chapitre, id_chap*).

Evaluation (id_eval, num_eval, question_eval, propositions_eval, reponse_eval).

Exercice(id_exo, num_exo, contenu_exo, solution_exo).

Faire_eval(id_app, id_eval) .

Faire_exo(id_app, id_exo).

Formateur(id_form, nom_form, prénom_form, date_naiss_form, adresse_form, email_form, sexe_form, login_form, mdp_form).

Forum (id_forum, message_forum, login_p, titre, reponse, date, heure).

Groupe (id_grp, nom_grp, nb_inscrits, ens, id_app*).

Messagerie (id_messagerie, message_messagerie, emetteur, recepateur).

Participer (id_m, id_outil).

Préinscription_app (id_app, nom_app, prénom_app, date_naiss, adresse, sexe, login, mdp).

Préinscription_aut (id_aut, nom_aut, prénom_aut, date_naiss, adresse, sexe, login, mdp).

Préinscription_form (id_form, nom_form, prénom_form, date_naiss, adresse, sexe, login, mdp).

Suivre (id_form, id_grp).

V. Conclusion

Tout au long de ce chapitre, nous avons tiré au clair les données et traitements qui interviennent dans le système que nous voulons mettre en œuvre pour ainsi mettre en évidence les interactions entre ce dernier et ses acteurs.

A ce stade de notre étude, nous sommes en mesure de commencer la réalisation de notre application.

CHAPITRE IV

I. Introduction

Après avoir présenté dans le chapitre précédent la conception et le fonctionnement global de notre système, nous arrivons dans ce chapitre à la mise en œuvre de notre application.

Ce chapitre sera divisé en deux parties : la première partie sera consacrée à la description des outils de développement de notre application, la deuxième partie nous focaliserons sur la construction des composants spécifiques au web, et plus spécialement les pages web ainsi la présentation des interfaces essentielles de notre application.

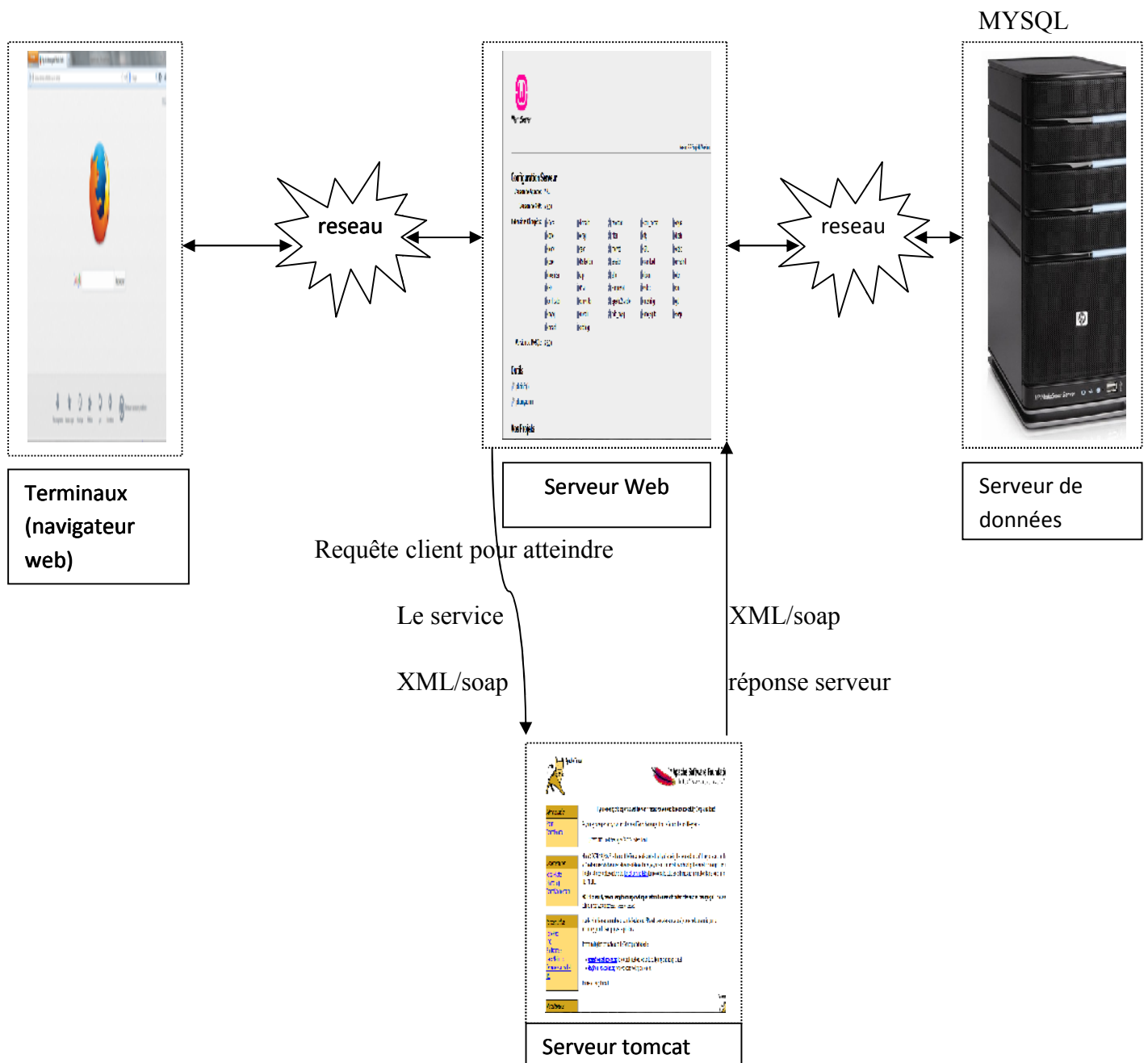


Figure IV.1 : architecture applicative générale

II. Outils technologiques de développement

II.1. Les langages utilisés :

II.1.1. Langage de programmation (Java)[11]

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparé en deux parties. D'une part, le programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution du programme Java. C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en ce langage puisse fonctionner.

Le langage choisi pour le développement de l'application est le langage JAVA qui répond au critère de portabilité maximale. En effet, ce langage, développé par « Sun Microsystems Inc », est pour les principales plates-formes du marché, qu'il s'agisse de Linux, Windows ou autres et est totalement gratuit. Java possède de nombreuses caractéristiques (orienté objet, robuste, fiable, multithread, rapide, extensible), mais le choix de ce langage a été motivé particulièrement par les caractéristiques qu'il présente :

- **Simple** : du fait que sa syntaxe soit basée sur celle de C++, mais dépouillée de tous les mécanismes complexes, redondants et inutiles.
- **Performant et rapide** : En effet, Java est d'une rapidité extraordinaire grâce à ses compilateurs spéciaux. Plus qu'un langage puissant, Java est une plateforme de développement comportant une bibliothèque de classes très riches et nombreux outils et interfaces de programmations applicatifs (API). Voir(FIGURE)
- **Interprète, portable et indépendant des architectures matérielles** : Cette caractéristique est un avantage primordial pour Java face à des applications transmises par un réseau et exécutées sur des machines hétérogènes. Un programme Java est successivement compilé pour fournir un code intermédiaire indépendant de la plate-forme d'exécution (le byte code) simple et rapide à traduire en langage machine.
- **Richesse** : Un des aspects importants de l'environnement JAVA est la richesse de ses bibliothèques de classes JAVA, accessibles via l'interface de programmation d'application (API) qui propose divers outils pour faciliter la programmation et manipuler des bases de données du texte, du son ou des images.

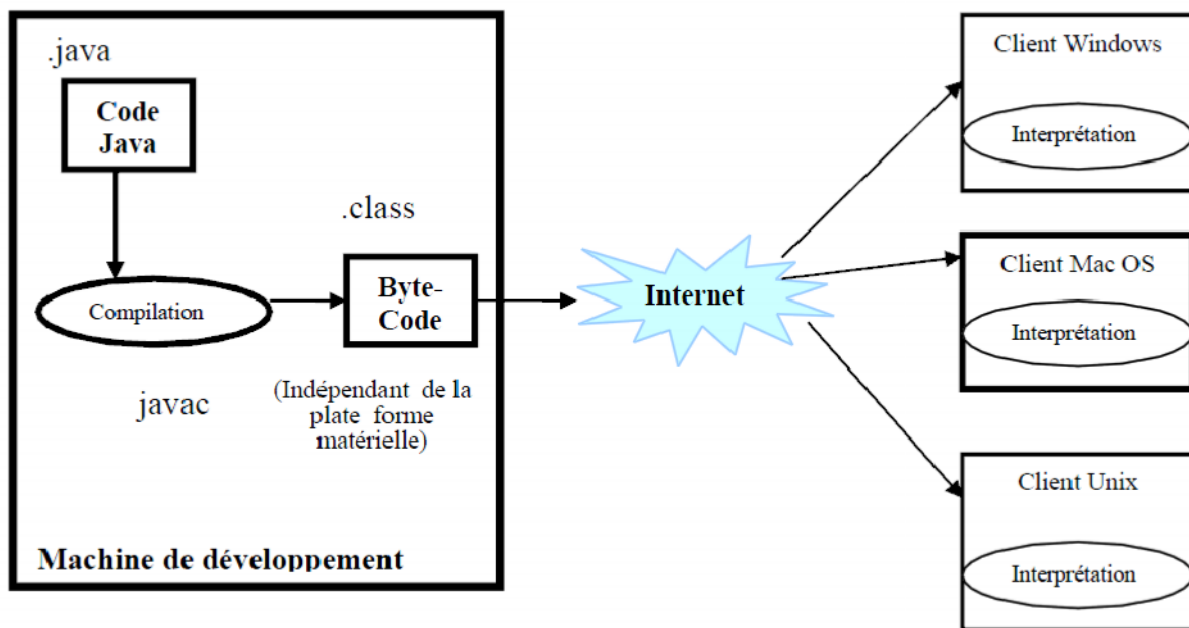


Figure IV.2 : Indépendance de Java des architectures matérielles.

II.1.2. Le langage PHP : (Personal Home Page) [12]

PHP est un langage de script coté serveur, incorporé au document HTML, mais exécuté par le serveur Web et non par le client. Conçu pour réaliser des pages dynamiques, le résultat du script est un document HTML standard, sans trace du script exécuté préalablement garantissant ainsi une compatibilité avec tous les navigateurs disponibles.

Les raisons qui nous ont amenées à choisir le langage PHP sont nombreuses. Nous citons les suivantes :

- Le PHP est rapide : compilé en tant que module Apache, les temps d'exécution sont plus courts, surtout sous Unix ;
- simplicité d'écriture de scripts ;
- Le PHP est multi plates formes : il fonctionne sous Windows, Unix....
- Le PHP gère très bien les requêtes SQL : On peut facilement écrire des programmes qui affichent des données extraites de bases SQL, ou qui stockent des données postées par un formulaire dans une table SQL. Le PHP sait communiquer avec presque tous les SGBD (Oracle, MySQL, DB2, SQL Server, Access) mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur les plates formes Unix, Linux, Windows ;

- Le PHP fournit une multitude de fonctions : couvrant presque tous les besoins pour un développeur de sites Internet : prise en charge de XML, génération de PDF (Portables Document Format), création d'images, compression/décompression, statistiques, cryptologie, génération d'email....
- gratuité et disponibilité du code source ;
- intégration au sein de nombreux serveurs Web (Apache, Microsoft IIS, ...) ;
- simplicité d'interfaçage avec des bases de données (de nombreux **SGBD** sont supportés, mais le plus utilisé avec ce langage est **MySQL**, un SGBD gratuit disponible sur les plates formes **Unix**, **Linux**, et **Windows**).

➤ **Fonctionnement de PHP**

Le serveur Web reconnaît de l'extension des fichiers, différente de celle des pages HTML simples, si le document appelé par le client comporte du code PHP.

1. Le serveur Web lance l'interpréteur PHP ;
2. L'interpréteur PHP traduit le document demandé et exécute le code source de la page ;
3. les commandes figurant dans la page interprétées et le résultat prend la forme d'une page HTML publiée à la place du code source dans le même document ;
4. la page modifiée est envoyée au client pour y être affichée par le navigateur ;

De cette façon, la page Web est créée dynamiquement, c'est-à-dire au moment même où le client est en dialogue avec le serveur.

II.1.3 Le langage HTML : (Hyper Text Markup Language) [24]

HTML est un langage de description de document, il utilise des marqueurs explicites (appelés tags ou balises) qui précisent la structure et la mise en forme du contenu du document. Ces marqueurs seront reconnus par les navigateurs, et interprétés comme des directives, afin de réaliser la présentation attendue sur le poste client. Il inclue des informations variées (textes, images, sons, animations) et permet d'établir des relations cohérentes entre ces informations grâce aux liens hypertextes.

II.1.4 Le langage JavaScript [24]

JavaScript est l'un des principaux langages utilisés pour la programmation de pages Web. Il est parfaitement adapté au développement pour l'Internet, car il s'agit d'un langage de scripts (interprété lors de l'exécution, plutôt qu'un langage compilé). Il n'exige aucune configuration spéciale des serveurs web, et il permet en quelques lignes de code de dynamiser une page.

II.1.5 Le langage de requête MySQL : (Structured Query Language)

C'est un langage standard (langage standardisé en 1992) qui permet de communiquer avec la plupart des SGBD en utilisant la même syntaxe. Toutefois, chaque SGBD possède sa propre interface permettant d'établir une connexion.

II.1.6. Le SGBD (MYSQL)

MYSQL est un gestionnaire de base de données libre. Il est très utilisé dans les projets libres et dans le milieu industriel.

MySQL est un SGBD relationnel développé dans un souci de performances élevées. Il est multi threads, multiutilisateurs.

II.1.7. Les Servlets [11]

Une servlet est un programme java qui utilise des modules supplémentaires figurant dans l'API java. Elle s'exécute dynamiquement sur le serveur Web et permet l'extension des fonctions de serveur. Les servlets permettent donc de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages Web dynamiques).

Une servlet s'exécute dans un conteneur de servlet utilisé pour établir le lien entre la servlet et le serveur Web, dans notre cas, on utilise le conteneur Tomcat version 6.0. Ainsi le programmeur n'a pas à se soucier des détails techniques tels que la connexion au réseau, la mise en forme de la réponse à la norme http.

II.2. L'environnement de développement et d'implémentation

Dans ce paragraphe, nous allons décrire l'environnement utilisé pour le développement et l'implémentation de notre application web.

II.2.1. Environnement de développement (Eclipse Galileo) [22]

Pour le développement des composants logiciels java de notre application, nous avons utilisé la plate-forme « Eclipse » qui offre un environnement de développement java très complet. Elle fournit toutes les API (Application Programming Interface) de base. Ces API sont structurées en package et contiennent des classes réutilisables pour construire des programmes plus complexes.

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

IBM est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le cœur de son outil Websphere Studio Workbench (WSW), lui-même à la base de la famille des derniers outils de développements informatiques.

Eclipse utilise énormément le concept de modules nommés « plug-in » dans son architecture.

D'ailleurs, hormis le noyau de la plate-forme nommé « Runtime », tout le reste de la plate-forme est développé sous forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

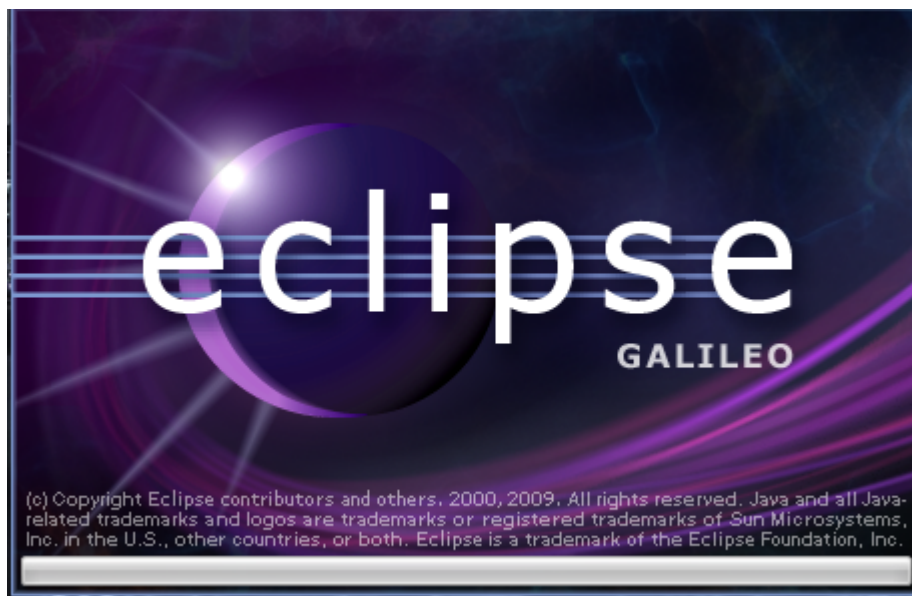


Figure IV.3 : Lancement d'éclipse GALILIO



Figure IV.4 : Accueil de l'IDE éclipse GALILIO

II.2.2. WampServer [26]

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP.

WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi qu'une administration pour les deux bases SQL PhpMyAdmin et SQLiteManager.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

La grande nouveauté de WampServer réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

II.2.2.1. Installer Wampserver

- Télécharger Wampserver sur le site wampserver.fr.malavida.com
- Double cliquer sur l'exécutable téléchargé.
- Sélectionner le répertoire d'installation et suivre la procédure.

II.2.2.2. Lancer wampserver :

On ne peut pas proprement parler du lancement de wampserver, il s'agit en fait de la mise en route du serveur apache et de serveur MySQL. A l'installation, un raccourci vers wampserver est créé dans le répertoire « Démarrer /programmes/wampserver ». Une fois wampserver lancé, une icône se place dans la barre des tâches à côté de l'horloge. Un clic droit permet d'accéder à différents menus :

- *Fichier log* : renvoie aux erreurs générées par Apache et MySQL ;
- *Configuration* : donne accès aux différentes configurations du wampserver;
- *Administration* : ouvre la page d'administration des alias et du répertoire des bases de données « PhpMyAdmin » ;
- *Web local* : ouvre le web local ;
- *Démarrer /Arrêter* : démarrer/arrêter Apache et MySQL
- *Redémarrer* : Redémarre Apache et MySQL
- *Quitter* : ferme Easy PHP.

II.2.2.3 Utiliser le répertoire www ou des alias :

Pour que les pages PHP soient interprétées, il est impératif de placer les fichiers dans le répertoire www ou dans un alias créé. Pour visualiser les pages, il suffit alors d'ouvrir le « *web local* » ou d'accéder aux alias via la page d'administration.

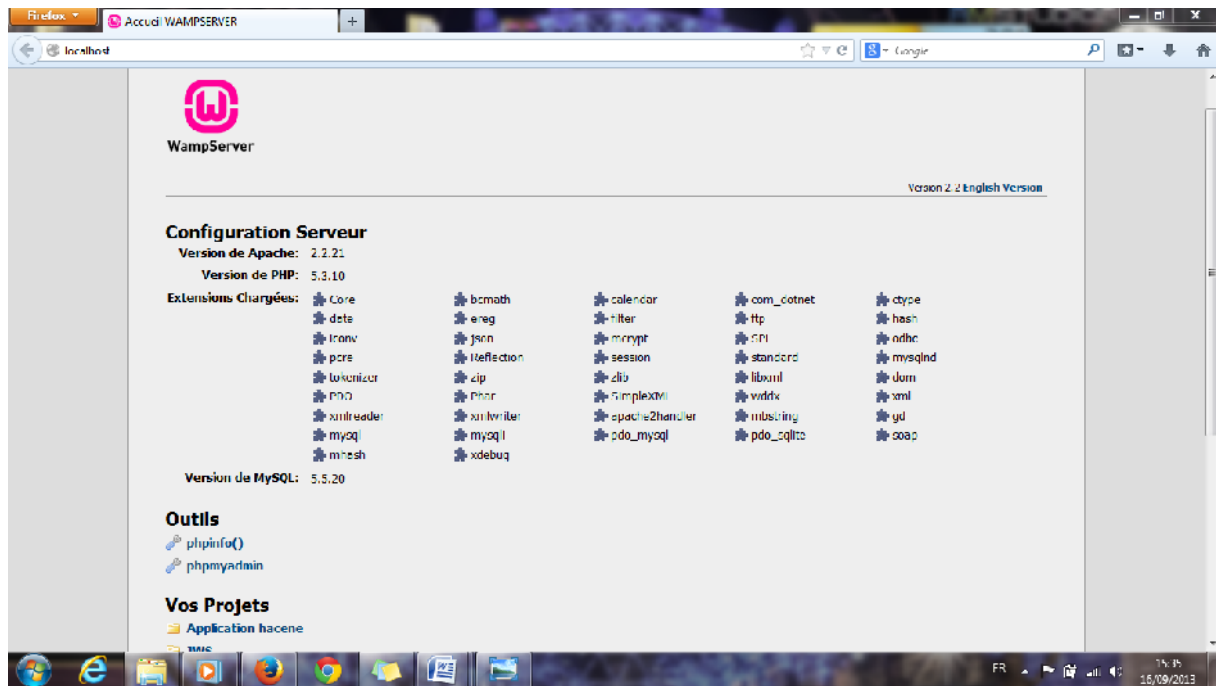


Figure IV.5 : Interface de WampServer

II.2.2.4. PhpMyAdmin

PhpMyAdmin est un utilitaire rendant plus conviviale l'administration de base de données MySQL. Il consiste en un ensemble de scripts PHP permettant d'administrer des bases de données MySQL en passant par un navigateur web.

➤ Les fonctions de PhpMyAdmin :

- Création et suppression de base de données ;
- Création, modification, copie et suppression de table ;
- Edition, ajout et suppression de champ ;
- Exécution des commandes SQL ;
- Création d'index ;
- Chargement de fichier dans des tables.

➤ Utilisation de PhpMyAdmin :

Pour accéder à PhpMyAdmin, il faut d'abord vérifier qu'Easyphp.exe est lancé et que le serveur fonctionne, après on peut accéder à partir

de « l'administration ». Pour ouvrir l'administration il suffit de faire un clic droit sur l'icône et sélectionner "administration". Une page Web apparaît, au milieu de celle-ci il y a un bouton PhpMyAdmin avec un simple clique là-dessus, la page d'accueil de PhpMyAdmin s'affiche dans la fenêtre du navigateur, accompagné d'un champ de sélection de base de données présente sur l'hôte MySQL par défaut, comme l'illustre la figure suivante :

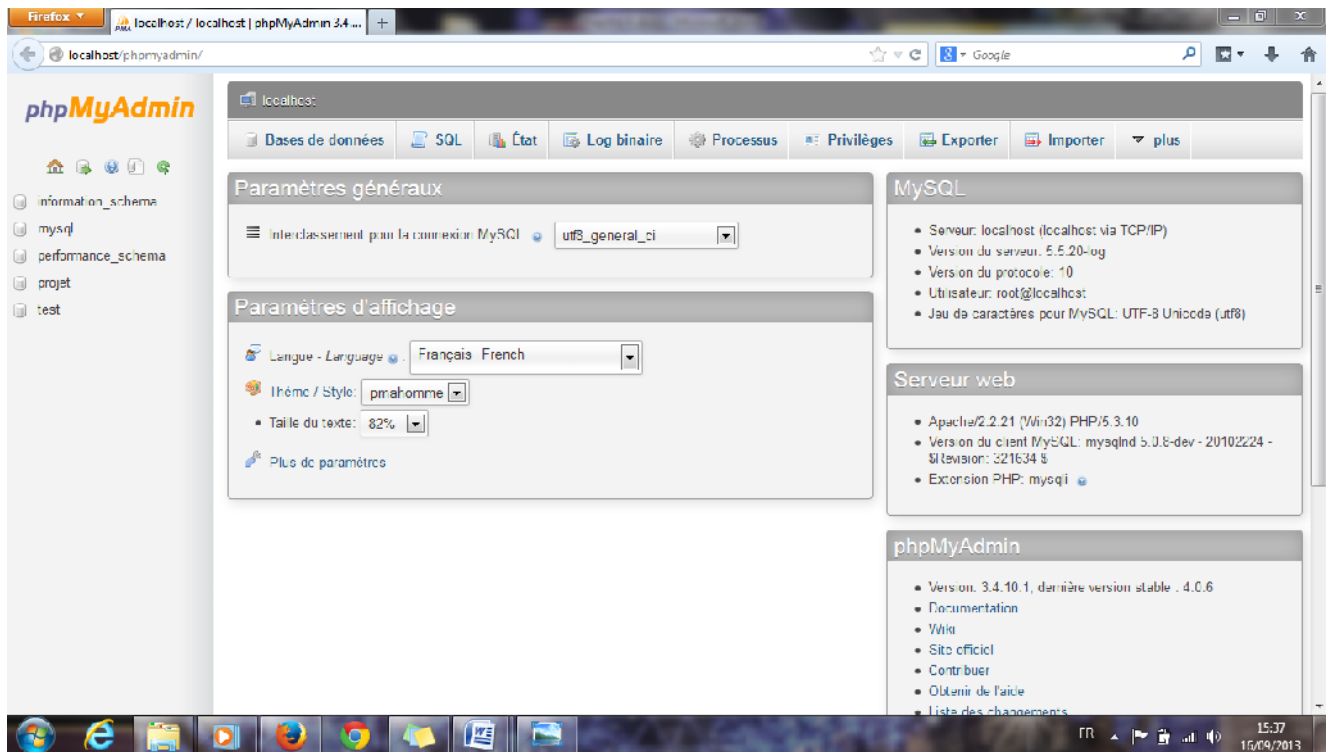


Figure IV.6 : Administration de MySQL à partir de PhpMyAdmin

III. les serveurs

III.1. le serveur [12]

Un serveur Web est un logiciel permettant à des clients d'accéder à des pages Web, c'est-à-dire des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant.

Un serveur Web est donc un logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole.

Les principaux serveurs Web sont: Apache, Microsoft IIS (Internet Information Server), Microsoft PWS (Personal Web Server).

Afin d'exécuter et de tester notre application durant la partie réalisation, nous avons opté pour l'utilisation d'Apache WebServer.

L'ancêtre Apache est le serveur libre développé par le National Center for Supercomputing Applications de l'université de l'Illinois. L'évolution de ce serveur s'est arrêtée lorsque le responsable a quitté le NCSA en 1994, les utilisateurs ont continué à corriger les bugs et à créer des extensions qu'ils distribuaient sous forme de "patches" (bouts de programmes ajoutés par les utilisateurs des NCSA pour étendre les fonctionnalités d'Apache) d'où le nom "a patchee Server" la version 1.0 de Apache a été disponible le 1 Décembre 1995.

Notre choix s'est porté sur le serveur Apache pour les raisons suivantes :

- ✓ Apache est aujourd'hui le serveur le plus répandu sur Internet;
- ✓ Il s'agit à la base d'une application fonctionnant sous les systèmes d'exploitation de la famille Unix, mais il a désormais été porté sur les différents autres systèmes dont Windows ;
- ✓ C'est un logiciel libre ;
- ✓ Il est extensible, modulaire et configurable ;
- ✓ Un niveau élevé de performance des exigences matérielles modestes ;
- ✓ C'est un serveur gratuit (peut être téléchargé à partir du site du groupe Apache à l'adresse '<http://www.apache.org>') ;
- ✓ Robuste et sécurisé ;

III.2. Serveur MySQL

C'est un véritable serveur de bases de données SQL multiutilisateurs et multitraitements, Cela permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes.

MYSQL est basé sur une bibliothèque de gestion de donnée éprouvée depuis de nombreuses années et faisant appel à des index d'arbres binaires. Grâce à cela, le cœur du système peut afficher une performance remarquable, tout particulièrement dans les accès indexés.

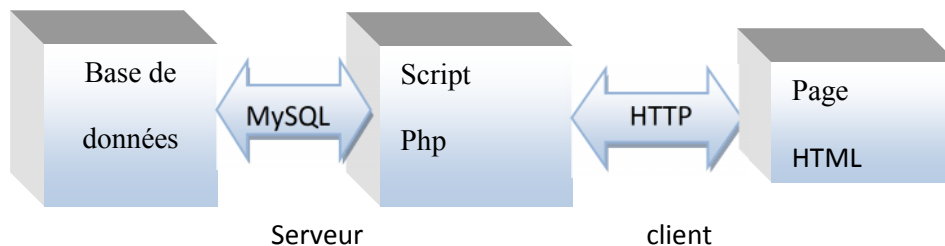


Figure IV.7: Serveur MYSQL.

III.3 Fonctionnalités de MySQL

La liste suivante décrit quelques fonctionnalités importantes de MYSQL :

- **Multitraitement :** MySQL est multitraitement en utilisant les threads du noyau. Il peut utiliser plusieurs CPU
- **Langues :** Le serveur peut fournir au client les messages d'erreurs en plusieurs langues.
- **Langages:** Les applications de bases de données MySQL peuvent être écrites en C, C++, Eiffel, JAVA, PERL, PHP, PYTHON et TCL.
- **Multi plateformes :** Prise en charge de plus de 20 plates-formes de système d'exploitation Windows, UNIX et LINUX.
- **Tables :** MySQL stocke chaque table sous forme de fichier distinct dans le répertoire de la base de données. La taille maximale d'une table comprise entre 4Go et la taille maximale de fichier acceptée par le système d'exploitation. Le mélange des tables de différentes bases de données est supporté dans une même requête.

- **Système de droits flexibles et sécurisé** : Système de droits flexible et sécurisé de mots de passe, et qui autorise une vérification faite sur l'hôte : le serveur détermine l'identité du client grâce à l'hôte depuis lequel il se connecte et le nom d'utilisateur qu'il spécifie. Puis, l'associer avec les droits d'utilisation des commandes selectes, insert, update et delete sur cette base (le serveur vérifie chaque requête émise pour voir si elle est autorisée).

III.4. Tomcat

Tomcat est le conteneur web le plus utilisé, il est open source gratuit et multi plateformes (écrit en Java).

Tomcat possède deux modes de fonctionnement :

- Autonome (*standard*) :

Tomcat est aussi un serveur Web, il est capable de servir des pages HTML et d'exécuter des Servlets et des JSP.

- Collaboratif (*in-process* et *out-of-process*) :

Tomcat peut s'installer comme une extension d'un serveur Web (Apache, Microsoft IIS ou Netscape NetServer), ce qui permet des meilleures performances pour le service des pages HTML.

III.5. Le module Tomcat

Le conteneur de servlets choisi est le moteur Tomcat, (ou moteur d'exécution), il est utilisé par Sun Microsystems pour l'implémentation de référence des servlets et des pages JSP. C'est en fait un serveur HTTP et un serveur de servlets/JSP tout-en-un écrit en java, ce qui rend cet outil très pratique pour tester un site internet qui gère des pages statiques et des pages dynamiques avec des servlets et des pages JSP.

Tomcat peut aussi être intégré au serveur WebApache ou au serveur Web Microsoft IIS. [44]

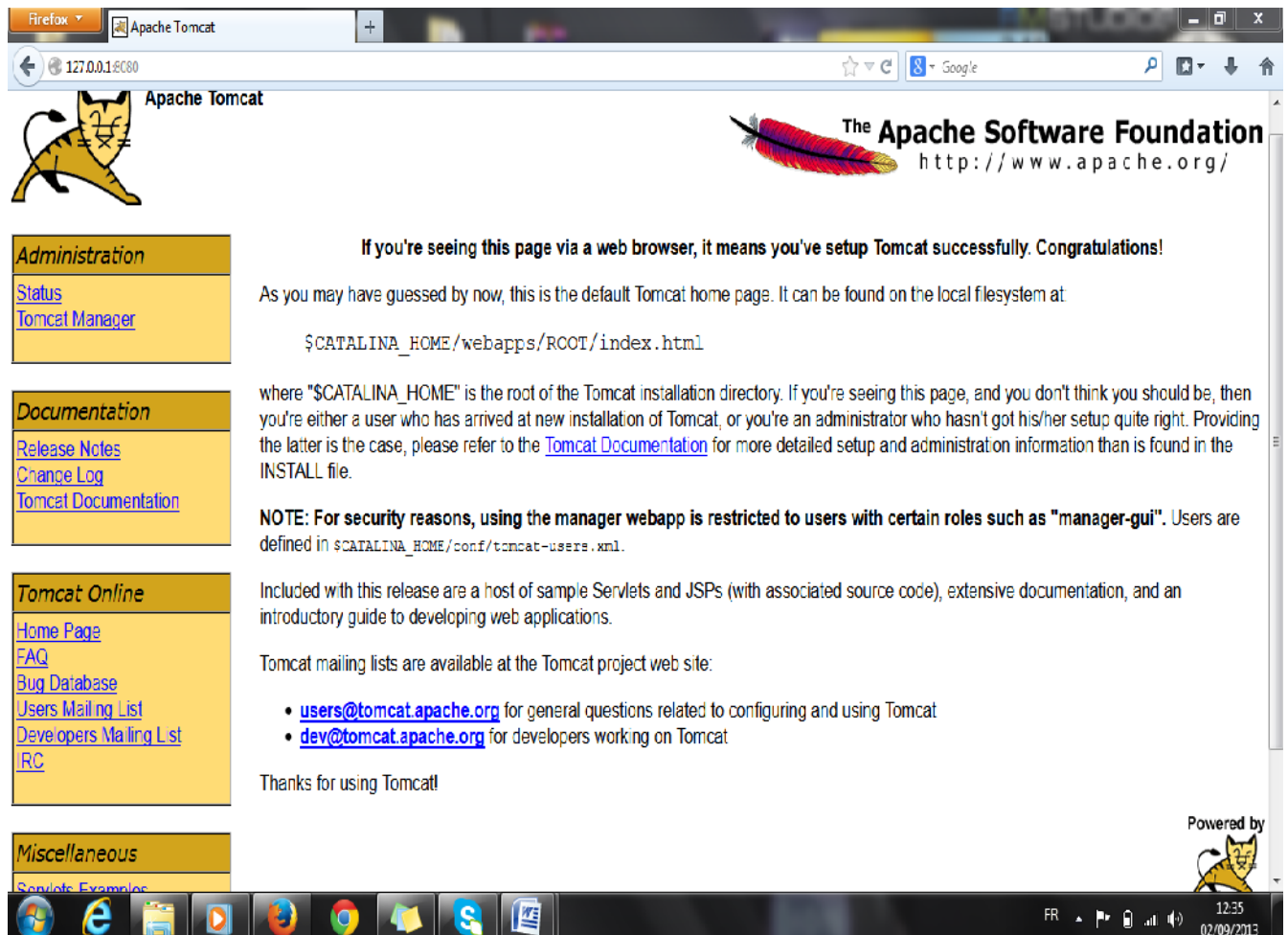


Figure IV.8 : Interface d'Apache Tomcat

IV. Navigateurs :

Un **navigateur Web** est un **logiciel** conçu pour consulter le **World Wide Web**. Techniquement, c'est au minimum un **client HTTP**. Le terme *navigateur web* (ou *navigateur Internet*) est inspiré de **Netscape Navigator**. D'autres **termes** sont utilisées notamment *browser*, en **anglais**.

Parmi les navigateurs les plus confrontés:

✓ Internet explorer



Windows Internet Explorer, ou plus simplement Internet Explorer, est le **navigateur web** de **Microsoft**, installé par défaut avec **Windows**. Depuis qu'il a détrôné **Netscape Navigator** à la fin des **années 1990**, c'est le navigateur Web le plus utilisé au monde. Aujourd'hui, son principal concurrent est **Mozilla Firefox**.

Les versions antérieures à la version 7 sont nommées « Microsoft Internet Explorer », tandis qu'à partir de la version 7, le logiciel est nommé « Windows Internet Explorer ».

✓ Mozilla Firefox



Mozilla Firefox est un [navigateur Web](#) gratuit, développé et distribué par la [Mozilla Foundation](#) aidée de centaines de bénévoles grâce aux méthodes de développement du [logiciel libre/Open Source](#) et à la [liberté du code source](#).

Firefox est à l'origine un [programme dérivé](#) du logiciel [Mozilla](#) , mais reprenant uniquement les fonctions de navigation de ce dernier. Ce logiciel multiplateforme est compatible avec diverses versions de [Microsoft Windows](#), [Mac OS X](#) et [GNU/Linux](#). Il a été porté sur d'autres systèmes d'exploitation, ce qui est rendu possible par la mise à disposition de son code source.

Le développement de notre application nécessite un navigateur tel que : *Mozilla Firefox*, *internet explorer* ou autres.

V. Présentation graphique de l'application

V.1. Page d'accueil de notre plate forme e-learning

C'est la première page qui s'affiche lorsqu'un utilisateur saisit l'adresse : « <http://127.0.0.1/plate-forme/> » dans son navigateur localement.

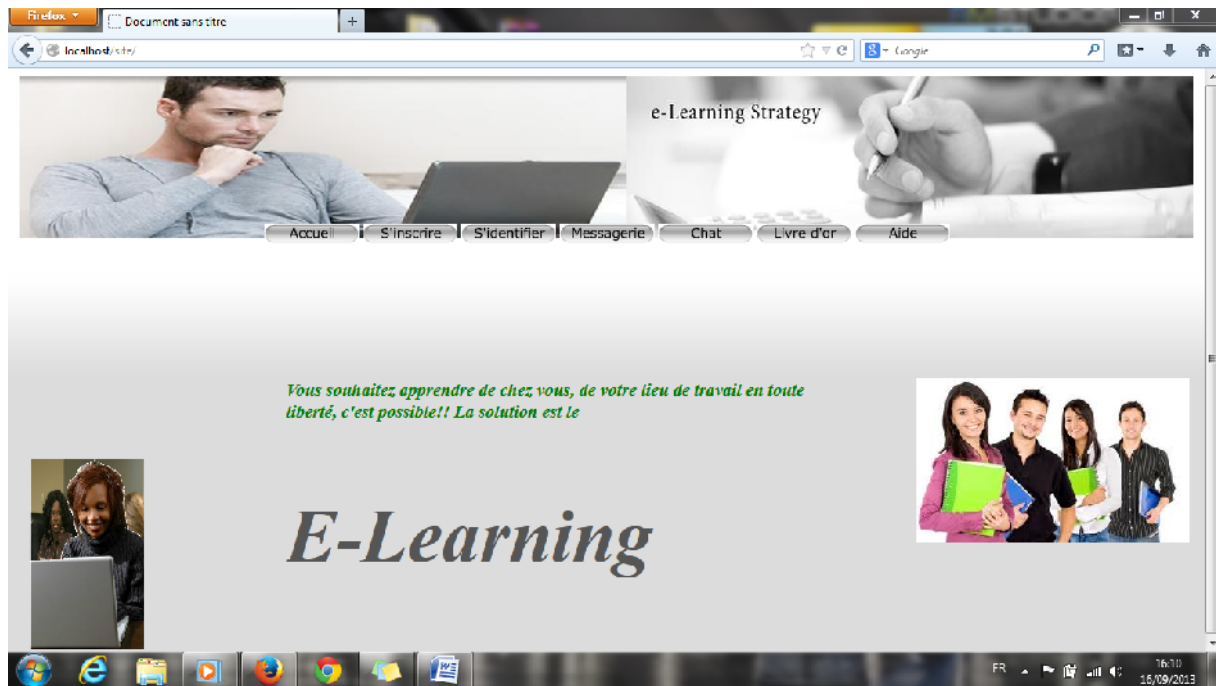


Figure IV.9: page d'accueil de notre plate forme

V.2. Page d'inscription

Si l'utilisateur clique sur «s'inscrire», un choix d'adhésion (apprenant, formateur, auteur) lui sera proposé et qui s'affiche dans le cadre principal de la page d'accueil.



Figure IV.10 : Interface d'inscription a la plate-forme

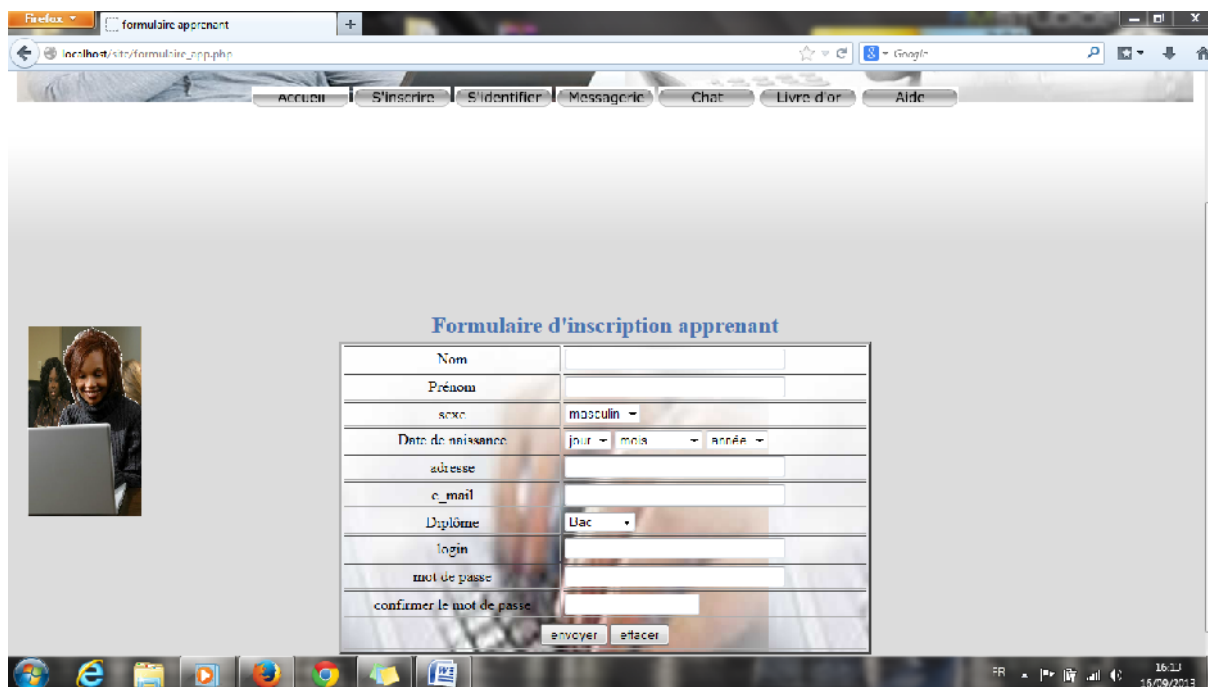


Figure IV.11 : Formulaire d'inscription Apprenant.

V.3. espace apprenant

L'espace qui permet aux apprenants d'utiliser les fonctionnalités de la plate forme qui lui sont offertes et accéder au service web en tapant sur le bouton TP.

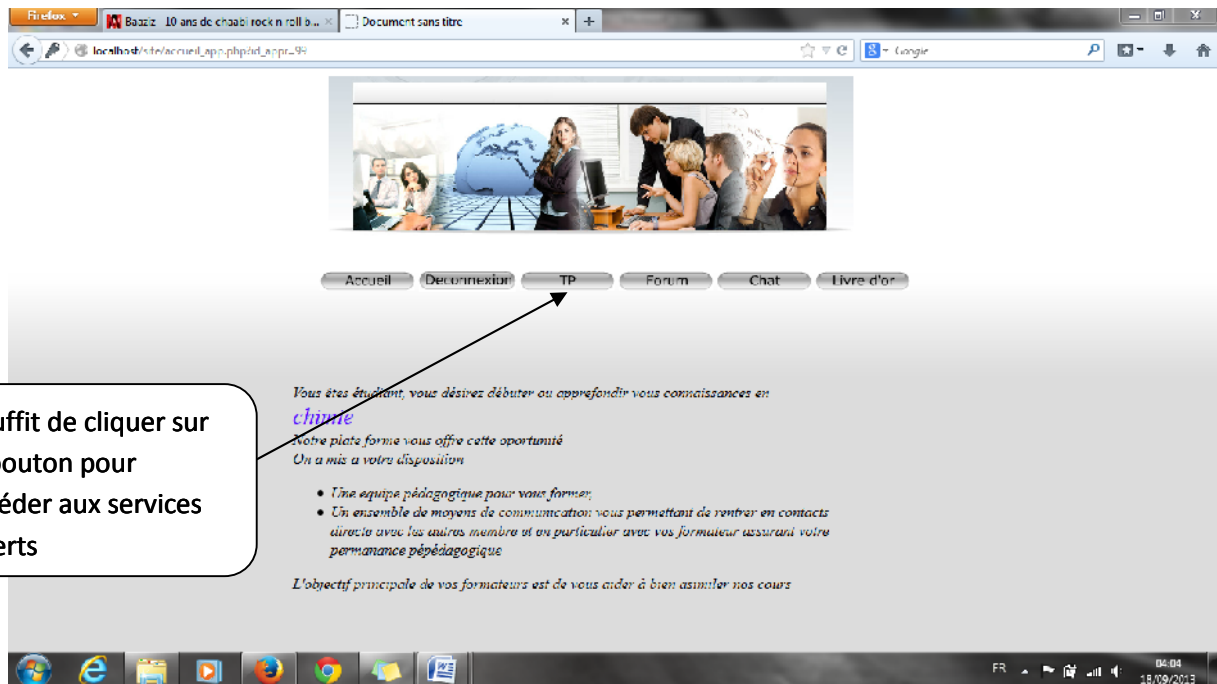


Figure IV.12 : Interface espace Apprenant

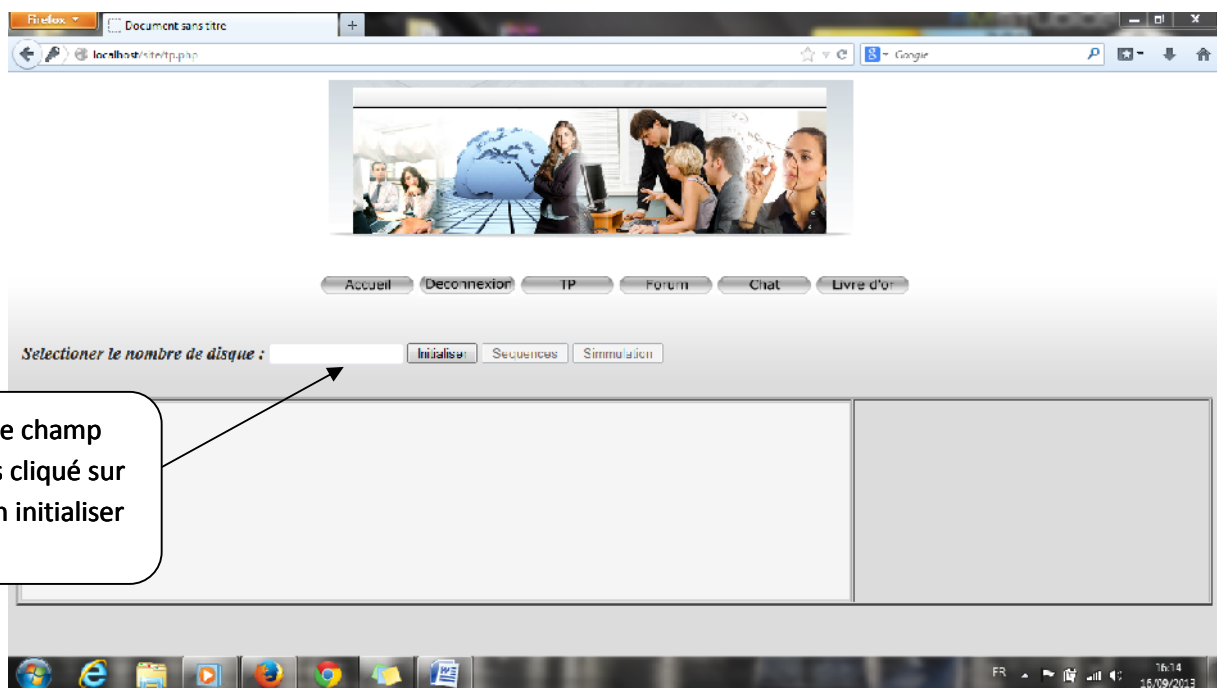


Figure V.13 : Interface Télé-TP

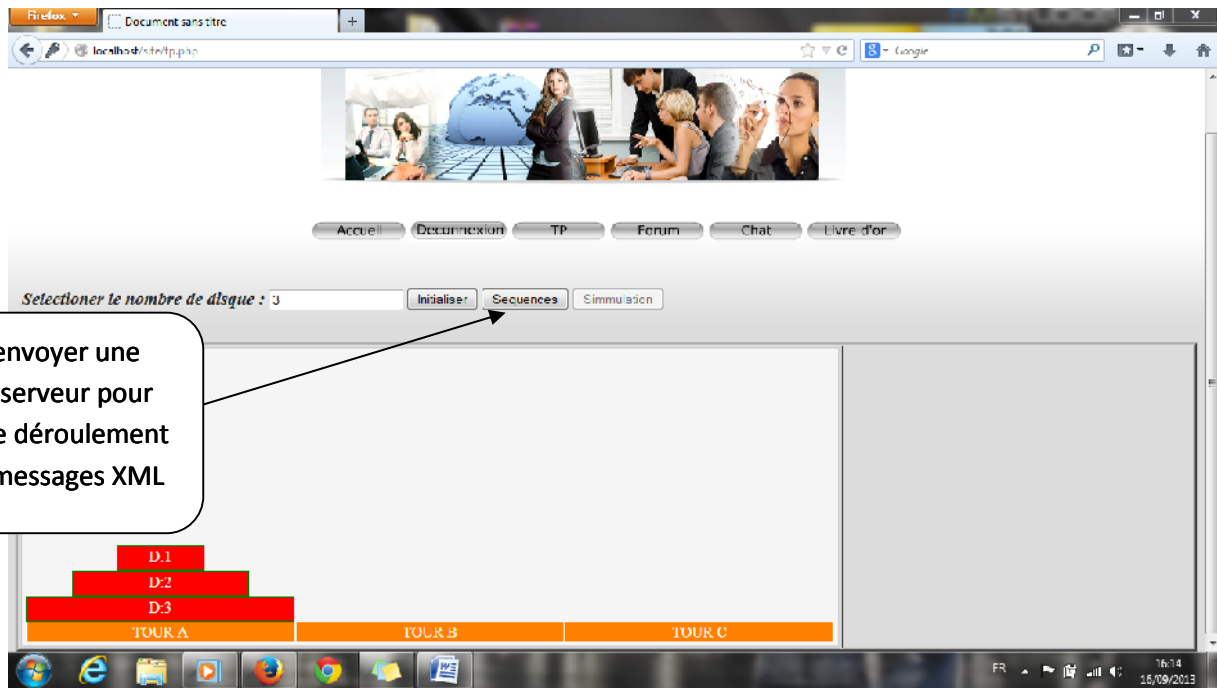


Figure IV.14: Résultats retourné après l'initialisation

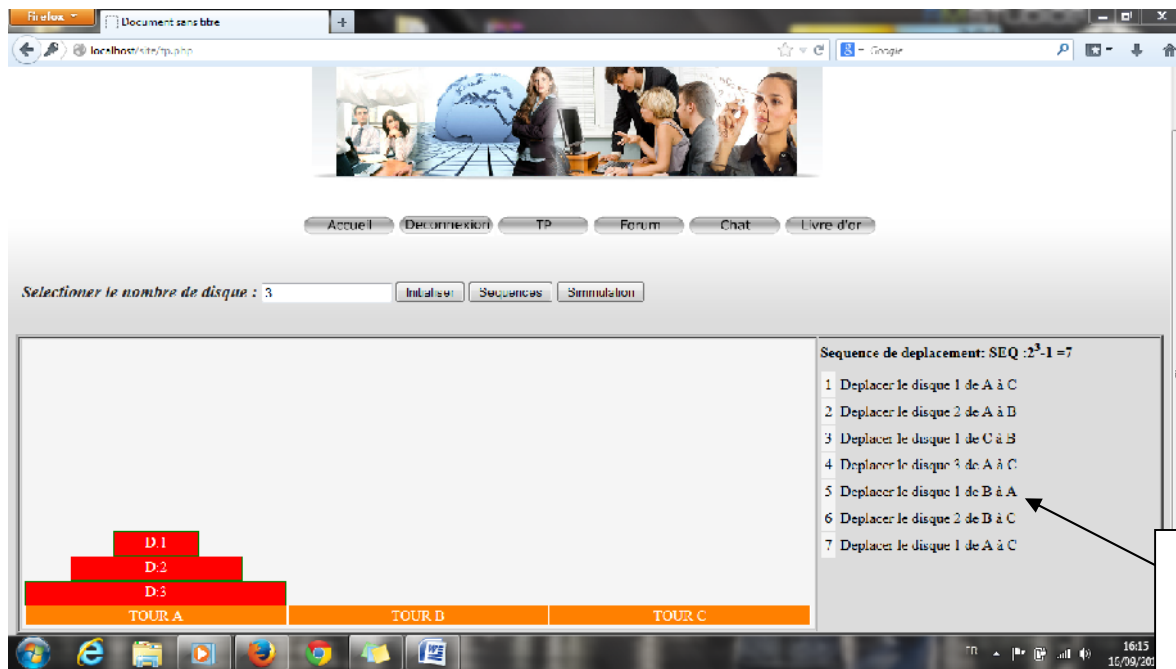


Figure IV.15 : Interface qui montre les séquences de déplacement

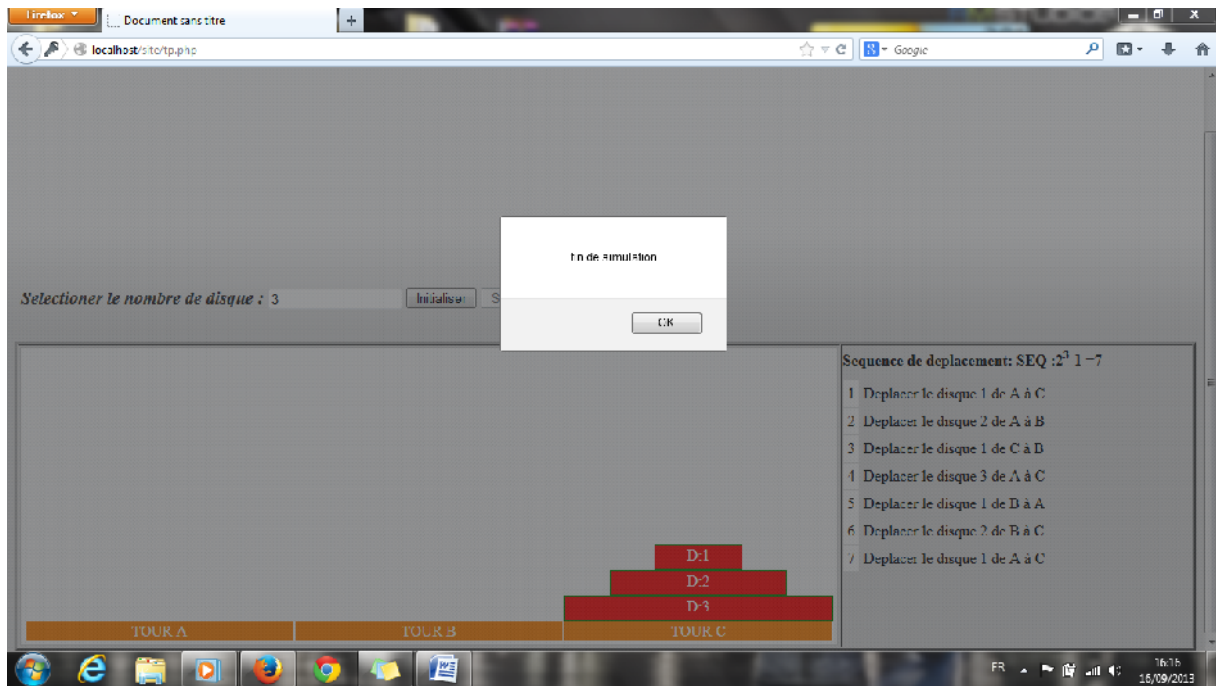


Figure IV.16 : Fin de simulation

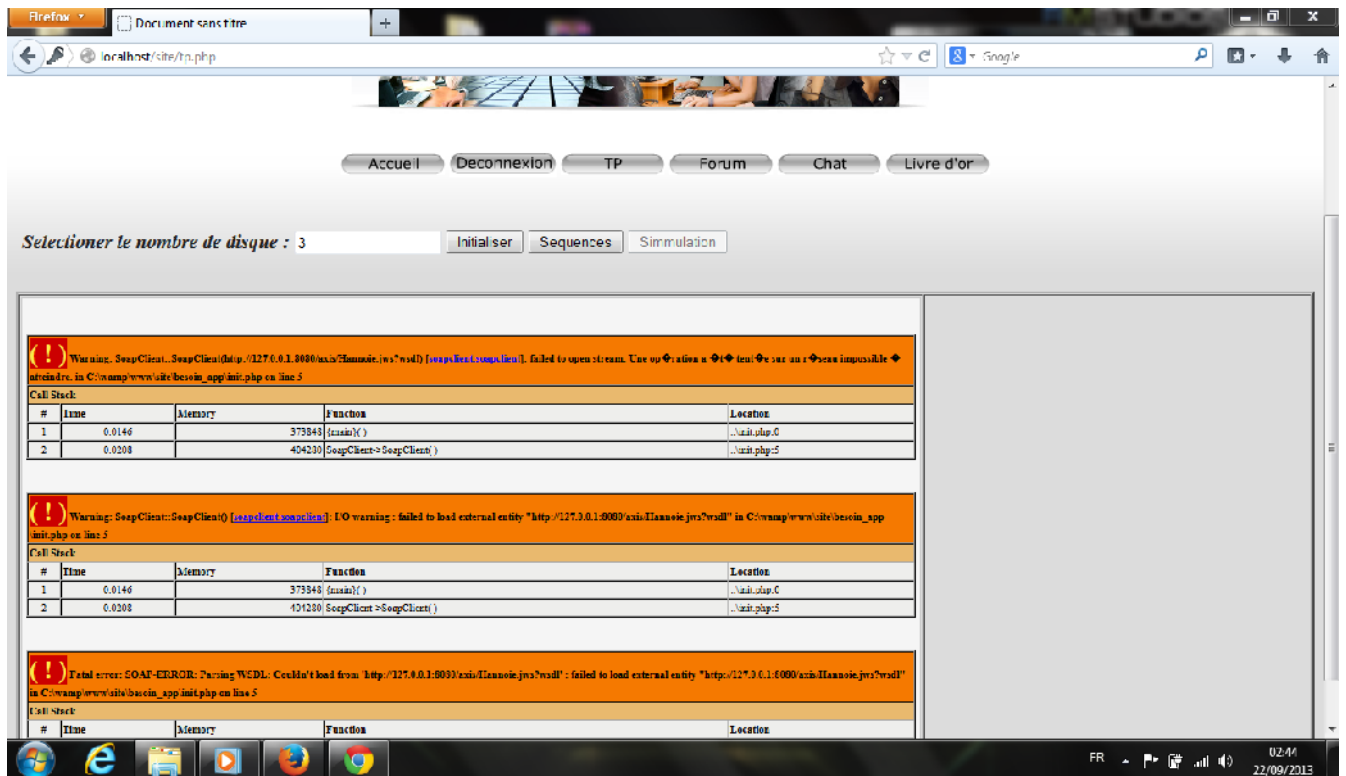


Figure IV.17: Message d'erreur lorsque serveur tomcat est hors service

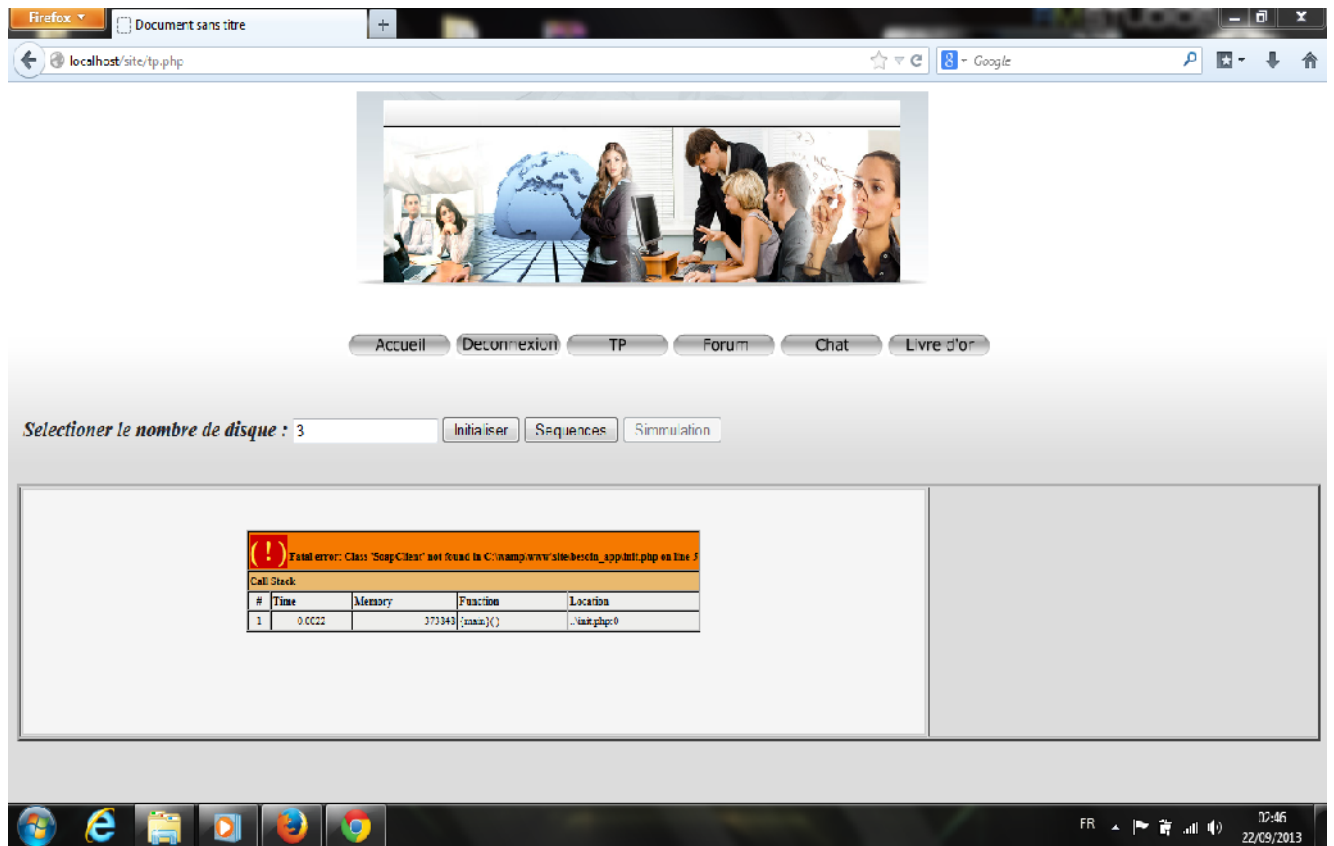


Figure IV.18 : Erreur générée lorsque le protocole soap ne fonctionne pas

VI. Conclusion

Ce dernier chapitre a été dédié à la présentation de notre application, ainsi, nous avons évoqué les différents outils qui ont servis à la réalisation de ce projet, l'architecture adoptée et pour finir, vous avez pris connaissance de quelques exemples de cas d'utilisation via des interfaces graphiques de notre application.