REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

Mémoire

de fin d'études

En vue de l'obtention d'un diplôme de Master II

Options **:** **conduite de projet informatique**

Thème :

Conception et réalisation d'une plate forme e-Learning adaptable supportant des télé cours synchrone

Proposé et dirigé par :

M[r] KERBICH.M

Réalisé par :

M[elle] ZENINE Naima

2013/2014

Remerciements

Je tiens tout d'abord à remercier mon promoteur M. kerbiche d'avoir accepter de me prendre en charge pour la réalisation de ce travail.

Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Table Des Matières

Chapitre II : Fondation des services web-les protocoles internet

Chapitre III: Les technologies XML

Chapitre IV : Généralités sur e.learning

Chapitre V : analyse et conception

Chapitre VI : réalisation et mise en œuvre

Table des figures

introduction générale

Introduction générale

Les Technologies de l'Information et de la Communication, Internet en particulier, ont envahi de nombreux domaines tels que le commerce traditionnel (e-commerce) et les administrations (e-administration), Internet est en passe de devenir la clé de voûte d'une nouvelle forme d'enseignement : la e-formation. Si l'engouement pour ce nouveau concept est croissant, l'offre reste néanmoins cantonnée aux domaines où l'enseignement théorique prime sur l'enseignement pratique. En effet, les services actuels de formation à distance reposent essentiellement sur des télé-cours, télé-TDs, télé-projets,..., sans réelle activité pratique. Pourtant ce type d'enseignement est indispensable dans les disciplines scientifiques et techniques, mais la mise en place de travaux pratiques à distance (télé-TPs) se heurte à des difficultés techniques et organisationnelles. Les récents travaux de recherche en e-learning tentent de donner un nouvel élan à cette discipline en proposant des solutions et des architectures génériques favorisant l'adaptabilité, la réutilisation et l'interopérabilité.

Contexte de nos travaux

Globalement, notre contexte de travail s'intègre dans celui des EIAH (Environnements Informatiques pour l'Apprentissage Humain) et plus particulièrement dans celui de l'e-learning. Nous nous intéressons à offrir des environnements intégrant l'activité pratique au sein des dispositifs de formation à distance existant. Ces environnements doivent répondre aux exigences de tous les acteurs de la formation à distance notamment les concepteurs des contenus d'apprentissage en leur offrant les outils nécessaires pour une meilleure efficacité dans leur travail grâce à l'adaptabilité et à la réutilisation des éléments crées au préalable.

Problématique

Le retard pris par le développement d'une véritable recherche autour de la problématique des applications d'e-learning peut s'expliquer notamment par l'interdisciplinarité que nécessite une telle approche. Dans les systèmes e-learning, une dimension d'adaptabilité non négligeable vient s'ajouter aux problèmes classiques engendrés par la mise à distance. La question qui se pose alors est : comment offrir un environnement d' e-learning aux acteurs de

la formation à distance sans effectuer des mises à jour quand il s'agir d'accéder à des nouveaux sites ?

Objectifs du travail

Ce travail porte principalement sur l'étude d'un système ouvert d'édition (mis à la disposition de l'auteur) et d'exécution (par les apprenants et le formateur) de scénarios pédagogiques de d'e-learning. Ce système devra être réutilisable quelle que soit la matière scientifique concernée (physique, chimie, automatique, productique, ..) et adaptable en fonction des objectifs pédagogiques recherchés et des contraintes liées au dispositif technologique. Ce système devra s'intégrer de manière homogène dans un environnement pédagogique général comprenant notamment télé-Cours, télé-TDs et télé-Projets. Cet environnement doit favoriser le travail collaboratif en offrant les outils nécessaires pour ce genre d'activité.

De ces objectifs, Il apparaît clairement que la solution qui devra être proposée doit nécessairement se conformer aux standards actuels de l'e-learning.

Plan du mémoire

Le premier chapitre de ce mémoire présente l'interopérabilité des applications et les Web Services, l'évolution des services Web, leurs découvertes, leurs publications, leurs cycles de vie et les technologies développées aux tours d'eux. Une attention particulière est prêtée à l'adaptabilité et au pattern de protection des variations.

Le deuxième chapitre s'intéresse à l'aspect technique des services, dans lequel, nous avons présenté le protocole HTTP, le protocole SMTP, la transmission des messages,

Vu que la technologie des services web se focalise sur le langage XML, une présentation de ce langage est nécessaire. Pour cela, nous avons réservé le chapitre 3 à ce langage.

Le quatrième chapitre se focalise sur la présentation du domaine d'étude à savoir l'e-learning et en particulier l'e-learning synchrone.

Le cinquième chapitre décrit la mise en œuvre d'une solution proposée en utilisant le principe de protection des variations et en passant par l'étape d'analyse et conception. Pour communiquer des résultats convainquant, le chapitre VI présente une mise œuvre de cette solution en utilisant l'EDI DreamWeaver et MySQL.

Pour conclure nous dressons un bilan sur notre travail, sur les résultats obtenus et nous ouvrons des perspectives de recherche.

Chapitre I

l'interopérabilité des applications et les web services

1. Introduction

L'interopérabilité est le fait que plusieurs systèmes, qu'ils soient identiques ou radicalement différents, puissent communiquer sans ambiguïté et opérer ensemble. Compte tenu du fait que ces systèmes sont produits par des constructeurs divers, avec des méthodes variées, et qu'ils répondent à des besoins spécifiques, l'idée la plus simple consiste à définir une base explicite, une norme, que chaque élément va « implanter » dans son propre fonctionnement. Cette norme joue un double rôle : elle est d'abord un indicateur de la façon dont le dialogue entre les différents éléments doit s'opérer — et cristallise donc les besoins de ce dialogue ; elle est ensuite une passerelle de communication, qui va pouvoir éventuellement s'adapter aux besoins changeants des éléments. La norme est alors proche d'une interface.

La notion d'interface est essentielle pour aborder l'interopérabilité. On ne peut parler d'interopérabilité que lorsque les interfaces sont complètement définies, connues et librement utilisables. Les interfaces sont beaucoup moins complexes que les systèmes qui les utilisent. De plus, elles sont stables sur de longues durées car indépendantes des évolutions de ces systèmes.

1.1.Des enjeux cruciaux:

L'interopérabilité a de larges implications techniques. Elle peut aussi avoir une incidence sur l'organisation d'une entreprise ou d'un organisme, et pose des questions essentielles. Celles-ci ont trait par exemple aux données et à leur échange :
Est-ce que les gens (concepteurs de systèmes ou utilisateurs finaux) souhaitent partager leurs données ?Le cas échéant, dans quelle mesure et de quelle façon ?Comment définir une norme pour que l'interopérabilité visée soit à la fois la plus facilement accessible et la plus viable possible ?Comment faire pour qu'elle soit adaptée à des besoins complexes et parfois contradictoires ?

La standardisation constitue un élément de réponse pour certaines de ces questions. Économiquement, l'interopérabilité a des conséquences méconnues du grand public et parfois sous-estimées par les acteurs industriels. Si les produits de plusieurs concurrents ne sont pas interopérables (à cause de brevets exclusifs, de secrets de fabrications ou pour toute autre raison volontaire ou non), on peut aboutir à une situation monopolistique ou bien à un marché fragmenté.

Une telle configuration économique se fait au détriment du consommateur. L'informatique, notamment, présente différents cas : la position de Microsoft par rapport à ses concurrents sur le marché des systèmes d'exploitation illustre bien le premier cas. Les gouvernements peuvent essayer d'encourager les constructeurs à engager une démarche d'interopérabilité concertée, mais cela se heurte concrètement à des intérêts commerciaux déjà en place. De telles démarches peuvent aussi conduire à des accords semi-ouverts, semi fermés, c'est-à-dire excluant un ensemble d'acteurs économiques au profit d'un petit groupe. Enfin, l'interopérabilité peut renvoyer aux problématiques de la liberté (liberté d'utilisation, liberté de choix...).

1.2. Nécessité des normes :

L'interopérabilité nécessite que les communications obéissent à des normes, clairement établies et univoques (voir Normes et standards industriels). Ces documents techniques définissent souvent des exigences, parfois accompagnées de recommandations plus ou moins optionnelles. Si la norme est correctement écrite, deux systèmes qui satisfont auxexigences doivent dialoguer ensemble sans souci particulier. Ils peuvent ainsi évoluer librement sans risque de casser cette possibilité de communication, tant qu'ils respectent lanorme définissant leurs interfaces.

Par exemple, la norme peut définir des éléments comme :

- les formats des données échangées dans le contexte considéré, qui décrivent des séquences d'informations ou de commandes qu'un système doit envoyer, comment ses correspondants doivent y répondre (protocole de communication).
- les tensions et courants à utiliser ;
- les types de câbles à utiliser.

1.3.Approches de l'interopérabilité

On distingue les normes ouvertes, les formats fermés et les normes à distribution restreinte.

1.3.1. Les normes ouvertes et communautaires :

Certains groupes (souvent des consortiums ou des associations) ont un processus de rédaction des normes qui est collaboratif : sous certaines conditions, des individus ou des entreprises peuvent adhérer et participer à des groupes de travail qui élaborent la documentation technique qui constituera la norme. Cette norme est ensuite publiée parfois d'abord à l'état de brouillon ou draft, dont les essais d'implémentation permettront d'en trouver les failles et d'en corriger les défauts, puis de candidat à la publication et enfin de recommandation officielle ou de document d'information.

Cette publication est ouverte, tout un chacun a la possibilité d'étudier ces documents et de tenter de développer un système conforme à ces normes. De plus, le fait que la rédaction soit relativement ouverte à la communauté évite de voir des normes publiées qui ne satisfont qu'une minorité qui détiendrait un pouvoir de décision sur leur contenu.
Exemple d'organisme fonctionnant selon un processus ouvert : IEFT, W3C, ISOC, OASIS.
Exemple de normes ouvertes : XML, XHTML, PNG, OggVobis …

1.3.2. Les formats fermés et propriétaires :

À l'opposé, le frein majeur à une interopérabilité optimale est l'utilisation dans des matériels et logiciels de formats dont seuls leurs concepteurs auraient les clefs. Cette

fermeture est souvent volontaire car elle vise, dans le cas d'un format de fichier propriétaire, à s'assurer qu'un utilisateur n'utilisera pas un autre logiciel pour lire ses données.

À moins d'avoir obtenu les spécifications du format auprès du concepteur, il est nécessaire d'avoir recours à la rétro-ingénierie, pour en reconstituer les spécifications et pour pouvoir développer des outils compatibles. Des lois sont cependant à l'étude pour encadrer ce genre de pratique, comme la DMCA aux États-Unis, ou Directive sur le copyright(ou EUCD) dans l'Union Européenne. Certains *trusts* y voient un intérêt et font pression pour qu'elles soient adoptées.

- Exemple : Les messageries instantanées propriétaires comme ICQ, Yahoo! Messenger ou MSN Messenger dont les protocoles ne sont pas compatibles et maintenus non interopérables.

1.3.3. Les normes à distribution restreinte

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

Parfois, c'est une très petite assemblée de personnes, voire une seule, qui décide d'une norme. Elle peut être plus ou moins à l'écoute des suggestions, bien sûr, de ses utilisateurs. Des exemples courants sont RAR d'AlexanderRoshal(algorithme de compression), PDF d'Adobe Systems(Format de document pour l'impression), Java de Sun MicroSystems(langage de programmation), Flash de Macromedia (format d'animation pour le Web), etc. Dans certains pays ; comme les USA et le Japon, il est également possible de devoir réserver des dividendes pour utiliser des normes, lorsque celles-ci sont brevetables. GIF (format d'image) et MP3 (format de son) en sont des exemples.

1.4.**L'interopérabilité en informatique** :

L'interopérabilité en informatique est une capacité juridique offerte au citoyen d'utiliser l'informatique sans se soucier d'aspects techniques.

Cette capacité doit permettre à tout citoyen, sans préjudice, d'obliger par les ordres, qu'il donne à un ordinateur, par l'intermédiaire d'un programme, de toute nature, de se coordonner, de coopérer et d'être piloté par tout autre programme d'une autre nature quel que soit le lieu, le matériel et le langage utilisé. Il doit le faire si le service attendul'exige. Le service rendu doit

être de même valeur satisfaisante qu'il eut été fait par l'un ou l'autre des programmes, dès lors que le service correspond, sans obstacle contractuel ni obstacle technique.

Cela induit :

- une conception iso-fonctionnelle de l'ergonomie employée pour atteindre le résultat quel que soit l'outil employé, quelle que soit la langue.
- une conception compréhensible du fonctionnement des machines et des logiciels : c'est-à-dire imitant la nature de manière à laisser l'humain reconnaître son interopérabilité avec les autres systèmes.
- une conception qui n'exclut pas la nécessité pour un citoyen d'être formé à son utilisation comme il l'aurait été pour d'autres outils de même nature.
- un style de conception qui n'exclut jamais le fonctionnement avec les autres systèmes informatique pour un même résultat attendu.
- une présentation de résultat, dans son style propre, qui ne trahisse jamais le contenu renvoyé des autres logiciels faisant la preuve au mieux que celle-ci augmente la valeur de celui-ci sans le masquer, ni l'appauvrir.

L'interopérabilité s'applique couche après couche dans tout système informatisé par des programmes automatiques, mais aussi doit être pris en compte dans les lois sans diminuer ni réduire la liberté dans la limite imposée par les droits de l'homme.

1.5.Service web :

Les **services web** (en anglais *web services*) représentent un mécanisme de communication entre applications distantes à travers le réseau internet indépendant de tout langage de programmation et de toute plate-forme d'exécution :

- utilisant le protocole HTTP comme moyen de transport. Ainsi, les communications s'effectuent sur un support universel, maîtrisé et généralement non filtré par lespare-feux ;
- employant une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées ;
- organisant les mécanismes d'appel et de réponse.

Grâce aux services web, les applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés.

Le premier bénéfice de ce découpage est la facilité de maintenance de l'application, ainsi que l'interopérabilité permettant de modifier facilement un composant (un service) pour le

remplacer par un autre, éventuellement développé par un tiers. Qui plus est, les services web permettent de réduire la complexité d'une application car le développeur peut se focaliser sur un service, indépendamment du reste de l'application.

Les services web facilitent non seulement les échanges entre les applications de l'entreprise mais surtout permettent une ouverture vers les autres entreprises. Les premiers fournisseurs de service web sont ainsi les fournisseursde services en ligne (météo, bourse, planification d'itinéraire, pages jaunes, etc.), mettant à disposition des développeurs des API (Application Programmable Interface) payantes ou non, permettant d'intégrer leurservice au sein d'applications tierces.

1.5.1. Les caractéristiques d'un service Web

La technologie des services Web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web.

Un service Web possède les caractéristiques suivantes :

- il est accessible via le réseau ;
- il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire ;
- il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (généralement HTTP, mais rien n'empêche d'utiliser d'autres protocoles de transfert tels : SMTP, FTP, BEEP...) ;
- l'intégration d'application en implémentant des services Web produit des systèmes faiblement couplés, le demandeur du service ne connaît pas forcément le fournisseur. Ce dernier peut disparaître sans perturber l'application cliente qui trouvera un autre fournisseur en cherchant dans l'annuaire.

1.5.2. Architecture d'un service Web

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Comme pour le **World Wide Web**, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

REST

REST (*Representational State Transfer*) est une architecture de services Web. Élaborée en l'an 2000 par*RoyFiedling*, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

XML-RPC

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications. Par exemple, un client Java peut parler de XML-RPC à un PerlServer !

SOAP

SOAP (*Simple object Access Protocol*) est un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes.
Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

WSDL

WSDL (*Web Services Description Language*) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

UDDI

UDDI (*Universal Description, Discovery and Integration*) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients.

Les informations qu'il contient peuvent être séparées en trois types :

- les pages blanches qui incluent l'adresse, le contact et les identifiants relatifs au service Web ;
- les pages jaunes qui identifient les secteurs d'affaires relatifs au service Web ;
- les pages vertes qui donnent les informations techniques.

1.5.3. Description en couche des services Web

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches sans être dépendante de façon excessive de la pile des protocoles. Cette structure est formée de quatre couches majeures :

Découverte de services	UDDI
Description de services	WSDL
Communication	SOAP
Transport	HTTP

Couches technologiques des services Web.

- Le transport de messages XML-RPC ou SOAP est assuré par le standard HTTP.
- SOAP ou XML-RPC prévoit la couche de communication basée sur XML pour accéder à des services Web.
- La description d'un service Web se fait en utilisant le langage WSDL. WSDL expose l'interface du service.
- La publication et la découverte des services Web sont assurées par le biais du référentiel UDDI. Un référentiel UDDI est un catalogue de services Web.

Couche transport

Cette couche est responsable du transport des messages XML échangés entre les applications. Actuellement, cette couche inclut HTTP, SMTP, FTP, et de nouveaux protocoles tels que BEEP.

Couche communication

Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité. Actuellement, deux styles architecturaux totalement différents sont utilisés pour ces échanges de données. Nous avons d'un côté l'architecture orientée opérations distribuées (protocoles RPC) basée sur XML et qui comprend XML-RPC et SOAP et de l'autre côté une architecture orientée ressources Web, REST (Representational State Transfer) qui se base uniquement sur le bon usage des principes du Web (en particulier, le protocole HTTP).

Couche description de service

Cette couche est responsable de la description de l'interface publique du service Web. Le langage utilisé pour décrire un service Web est WSDL qui est la notation standard basée sur XML pour construire la description de l'interface d'un service. Cette spécification définit une grammaire XML pour décrire les services Web comme des ensembles de points finaux de communication (ports) à travers lesquels on effectue l'échange de messages.

Couche publication de service

Cette couche est chargée de centraliser les services dans un registre commun, et de simplifier les fonctionnalités de recherche et de publication des services Web. Actuellement, la découverte des services est assurée par un annuaire UDDI (Universal Description, Discrovery, and Integration).

1.5.5.**Interopérabilité et site web** :

Le partage de l'information géographique est aujourd'hui une nécessité reconnue par tous et divers dispositifs mis à disposition des acteurs d'un même territoire la permettent : observatoires territoriaux et infrastructures de données spatiales mises en œuvre au niveau d'agglomérations, de départements, de régions ou au niveau national. Ces dispositifs s'appuient techniquement sur divers standards d'interopérabilité mis au point par l'OGC ('Web Services' géographiques et de métadonnées notamment) pour assurer à la fois l'interopérabilité de chaque plate-forme avec les dispositifs de ses adhérents / partenaires et l'interopérabilité des plates-formes entre elles.

Ils permettent notamment :

- De moissonner des métadonnées, soit par transfert périodique automatisé, soit par accès distant via des requêtes à un géorépertoire (interface CS-W)
- D'accéder uniformément à des données distribuées, exposées par des Services Web, pouvant comporter égalementdes fonctions de traitement, et donc sans jamais requérir de transférer ou télécharger les fichiers natifs (interfaces WMS, WFS, WCS, WPS)

On rappelle la définition publiée sur son site par le Forum OGC France (FOF). « L'interopérabilité est la capacité des systèmes à communiquer, à échanger des données, à « travailler » ensemble, sans que l'utilisateur ait besoin de connaître les caractéristiques spécifiques à chaque système. L'interopérabilité est basée sur l'utilisation de standards définissant les interfaces des composants des systèmes. Une application connaissant les interfaces standards est capable d'utiliser n'importe quel composant respectant ces standards. »

3 composants techniques sont donc concernés par cette problématique d'interopérabilité :

- Les données, car la plupart des besoins d'interopérabilité concernent la capacité d'accéder à des données distantes et/ou stockées dans des systèmes divers, et leur mise à disposition dans un schéma applicatif (ou modèle de données) unifié: elles doivent donc si possible être accessibles à travers des interfaces de service et des encodages standardisée (GML, SensorML, WaterML, O&M…),
- Les services d'accès aux données (en encodage natif et/ou standard) et les services offrant des traitements sur ces données,
- Enfin, les métadonnées, qui permettent dans un premier temps d'identifier des données et des services, dans un second temps de les sélectionner, dans un troisième temps d'intégrer les accès ou traitements à des processus informatisés.

Il semble important de rappeler que ces standards sont conçus pour plusieurs types d'usages, impliquant comme émetteurs ou récepteurs des humains ('Humana') et des systèmes informatiques ('Machine'). Aujourd'hui, pour de nombreux géomaticiens, les métadonnées sont surtout un moyen pour un acteur (personne ou structure organisationnelle) de faire connaître à d'autres acteurs l'existence de données et de services et ils sont tentés de se focaliser sur les interactions entre humains. Dans ce registre, les métadonnées et les outils

d'exploitation associés posent un certain nombre de problèmes (ergonomie, complexité sémantique, etc). On verra plus loin que les métadonnées doivent également être conçues comme un moyen pour les systèmes informatiques de dialoguer et d'échanger. Il s'agit ici véritablement d'interopérabilité et celle-ci pose des problèmes d'un autre ordre qui doivent également être abordés.

1.5.6. Cycle de vie pour le développement des services WEB

1.5.6.1. Activités de développement des services Web

Dans la pratique, le développement des services web consiste à réaliser un ensemble de tâches partant de la modélisation métier vers l'implémentation et le déploiement. L'aspect réutilisation est aussi très important. On peut citer deux processus pour développer les services web :

- Un processus agile de développement des services web ;
- Un processus global de développement des services Web.

1.5.6.2. Processus agile de construction des services Web

On peut envisager pour le développement des services Web de s'appuyer sur un processus agile dont les activités sont représentées sur la figure1.12 :

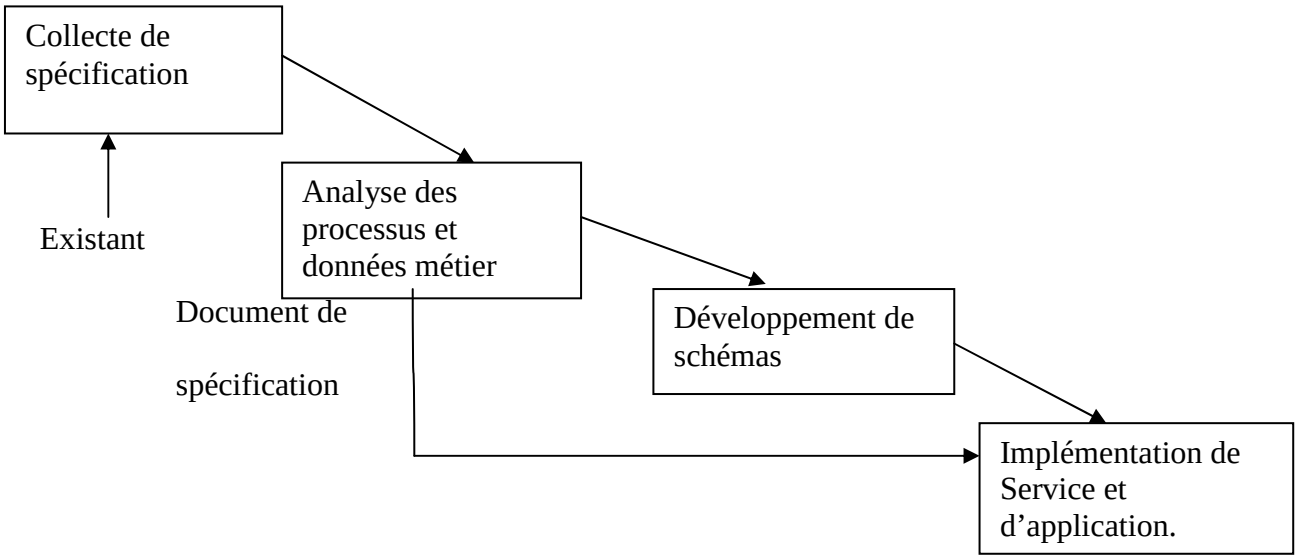


Figure 1.1. Activités de développement agile de services Web

L'analyse des processus et des données métiers permet de définir le périmètre et le but de l'étude, ce qui permet de mieux s'organiser. L'analyse peut être conjointement menée entre informaticiens et clients. Desmétamodèles, des patterns, etc., peuvent être utilisés pour compléter l'étude et conduire au développement de schémas.

Cette dernière activité est la conception et l'implémentation de services, où l'on pourra ainsi définir si l'on choisit par exemple des techniques de type EJB (Entreprise Java Beans), les SessionBeans issues des Workflow, les Entitybeans issues des classes des niveaux métiers pour construire les composants applicatifs et générer les fichiers techniques de description et de support des services web.

La prise en compte du besoin supporté par une approche axée sur l'intégration de composants permet de concevoir non plus un système d'information monolithique mais plutôt un système d'information modulaire où les composants communiquent par envoi de messages.

1.5.7. Etapes d'exécution des services Web

D'une manière simplifiée, les principales étapes d'exécution d'un service Web sont les suivantes :
- Découverte du service. Le demandeur de service lance la recherche d'un service correspondant à ses besoins sur un annuaire UDDI qui peut être public ou privé.
- Récupération des informations de description du service. Le demandeur de service récupère de l'annuaire UDDI la description de ce service au format WSDL.
- Connexion au service Web : la communication entre le composant demandeur de service et celui fournisseur de service est assurée en phase d'exploitation à travers des wrappers (listener et proxy) SOAP qui servent d'interface entre ces composants et les protocoles de communication de l'infrastructure de déploiement. Le proxy du composant demandeur de service émet une requête SOAP au composant fournisseur de service. Le protocole http véhicule le message SOAP jusqu'au listener du fournisseur de service.
- Le service Web renvoie sa réponse : Le service Web du fournisseur renvoie sa réponse au demandeur sous la forme d'un document XML via SOAP et http.

Dans la pratique, la synchronisation de traitements impliquant plusieurs services Web reste difficile. L'orchestration de processus étendus reste difficile à réaliser faute de fonctions transactionnelles au niveau de la couche de transport SOAP et d'outils adaptés pour effectuer le déploiement.

1.5.8. XML

Les technologies XML sont une composante fondamentale de l'informatique depuis une dizaine d'année. Elles se sont introduites dans toutes les disciplines fondamentales (structuration de données, bases de données, Web, middleware, etc.) ou sectorielles (math, chimie, musique, etc.) et on ne peut y échapper. On peut alors soit les subir, soit les comprendre et en tirer le meilleur parti, car XML est accompagné d'une fabuleuse boîte à outil disponible dans tous les langages et toutes les plateformes et qui offre des possibilités spectaculaires dans l'extraction, l'agrégation et le remodelage des données, et notamment dans les applications Web.

Voici les principaux atouts de XML :

- La lisibilité : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML
- Autodescriptif et extensible
- Une structure arborescente : permettant de modéliser la majorité des problèmes informatiques
- Universalité et portabilité : les différents jeux de caractères sont pris en compte
- Déployable : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme HTTP
- Intégrabilité : un document XML est utilisable par toute application pourvue d'un parser (c'est-à-dire un logiciel permettant d'analyser un code XML)
- Extensibilité : un document XML doit pouvoir être utilisable dans tous les domaines d'applications

1.5.8.1 XML et l'intégration d'applications

L'intégration des applications d'entreprises (EAI, Entreprise Application Intégration) est un domaine privilégié pour exploiter l'universalité et la flexibilité apportées par les différentes normes XML :

- Au niveau des données, par la définition de formats de documents XML mais aussi dans la validation des messages avec les schémas XML ou dans la transformation de formats avec XSL.
- Au niveau de l'architecture technique, XML rend possible le découplage nécessaire à la communication entre applications hétérogènes.

En effet, L'EAI représente actuellement une combinaison de technologies mise en œuvre dans différents types de produits. Une solution EAI complète utilise les services de connectivité pouvant également être offerts par les produits de type middleware, les services de transformation de données pouvant également être offerts par les produits ETL (Extracting, Transforming and Loading data Tools) et les services de gestion de processus pouvant également être offerts par les produits de gestion de Workflows.

Intégration de processus Modélisation du workflow ou de processus
Intégration d'applications Translation et transformation de données Routage basé sur les règles.
Intégration de composants Serveurs d'applications
Intégration de données Outils d'extraction, transformation et chargement de données. Management de métadonnées
Intégration de plates-formes Messagerie, ORB et RPC

Figure1.2. Niveau d'intégration EAI dans le SI

Comme illustré sur la figure 1.9, nous pouvons identifier cinq niveaux d'intégration de composants d'un système d'information en utilisant les outils EAI :

Intégration au niveau de la plate-forme : ce niveau d'intégration assure la connectivité entre matériels, systèmes d'exploitation et plates formes applicatives hétérogènes. Les technologies qui offrent une intégration au niveau de la plate forme concernent la messagerie, l'ORB (Object Request Broker) et le RPC (RemoteProcedure Call). La messagerie fournit la connectivité asynchrone, le RPC offre la connectivité synchrone et les ORB couplés à la messagerie offrent les deux modes de connectivité.

Intégration de données : deux catégories de produits permettent actuellement d'assurer l'intégration de données de l'entreprise : les passerelles vers les bases de données et les outils d'extraction, transformation, déplacement et chargement de données couramment appelés en anglais ETML tools (Extracting, Transforming, Moving and Loading data). Les passerelles vers les bases de données assurent l'accès SQL aux sources de données hétérogènes. Ce sont des produits d'accès synchrones aux données qui requièrent de la part du développeur la connaissance du schéma de la base de données. Les outils ETML généralement batchs sont mieux adaptés aux chargements initiaux des entrepôts de données de l'entreprise ou aux transferts de masse batchs. Dans une architecture multi niveaux, les outils de cette catégorie extraient et chargent les données directement sans transiter par la logique de l'application. La plupart de ces outils sont originellement développés pour la construction des entrepôts de données (datawarhouse).

Intégration au niveaudes composants : ce niveaud'intégration facilite l'introduction de nouvelles fonctionnalités aux packages ERP (Entreprise Ressource Planning), aux applications clients/serveurs et aux autres applications existantes. Les serveurs d'applications initialement focalisés sur la gestion de transactions, sont des évolutions naturelles de moniteur TP (Transaction Processing) et offrent actuellement un ensemble de services en complément à la gestion de transactions comportant le LoadBalancing, la tolérance aux pannes, le pooling de connexion, la gestion des états et de sessions, la sécurité et l'accès aux sources de données relationnelles et non relationnelles.

Intégration au niveau des applications : ce niveau fournit un cadre général de mise en œuvre d'un ensemble de technologie assurant l'intégration d'applications. La logique de connexion à une application est définie à travers un adaptateur d'application dédié. A ce niveau, le message Broker gère la transformation et la translation de données, le routage basé sur les règles et la connectivité aux applications via les adaptateurs d'application dédiés (les adaptateurs SAP par exemple).

Intégration au niveau processus : l'intégration au niveau processusest le plus haut niveau d'abstraction et d'adaptabilité pour une solution EAI. Pour s'adapter aux processus complexes inter entreprises, les offres d'intégration B2B combinent les outils d'intégration EAI classiques -adaptateur et courtier de messages- à un moteur de workflowqui pilote le courrier de messages selon des scénarios établies par les architectes métiers. Cela est facilité par l'utilisation des outils à interface graphique de modélisation des processus métiers. Un workflow est capable de piloter à la fois des applications et des hommes, des procédures des protocoles de transport inclus s'exécutent sur une infrastructure sécurisée. Le moteur de

workflow assurant le BPM (Business Process Management) est un composant de l'infrastructure d'exécution chargé de piloter le courtier de message.

Doté de sa fonction BPM, l'EAI semble être une bonne technologie pour :
- Supporter la vision processus dans la conception et l'urbanisation des SI ;
- Répondre aux besoins de la flexibilité et de la capacité d'adaptation ;
- Fournir une solution de robustesse et transactionnelle minimale face aux difficultés actuelles de disposer de transactions réparties ACID.

1.6. Adaptabilité

exprime la capacité du logiciel à communiquer et à utiliser les ressources d'autres logiciels comme, par exemple, les documents créés par une certaine application sans qu'il subit des mises à jours.

Patron de conception protection des variations [GRASP]

Problème : comment concevoir des objets, des sous-systèmes ou des systèmes de façons que les variations ou l'instabilité de ces éléments n'aient pas d'impact (adaptabilité) indésirable sur d'autres éléments ?

Solution : identifier les points de variation ou d'instabilité prévisibles. Affecter les responsabilités pour créer une interface stable autour d'eux.

Nous employons ici le terme interface au sens le plus large de vue d'accès, et non au sens littérale d'une interface Java.

1. 7. **Conclusion**

Dans ce chapitre, nous avons présenté un panorama sur l'interopérabilité et l'adaptabilité, pour expliquer au mieux ces aspects sur le plan technique nous avons décidé de consacrer au niveau du chapitre suivant la présentation la technologie des services Web.

Chapitre II

fondation
des services web
et les protocoles internet

2. Introduction :

Un **service web** (ou **service de la toile**) est un programme informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. Le protocole de communication est défini dans le cadre de la norme SOAP dans la signature du service exposé (WSDL). Actuellement, le protocole de transport est essentiellement HTTP(S).

Le concept a été précisé et mis en œuvre dans le cadre de *Web Services Activity*, au W3C, particulièrement avec le protocole SOAP. Associé avec les Échanges de Données Informatisés (EDI), le consortium ebXML l'a utilisé pour automatiser des échanges entre entreprises. Cependant le concept s'enrichit avec l'approfondissement des notions de ressource et d'état, dans le cadre du modèle REST, et l'approfondissement de la notion de service, avec le modèle SOA.

Dans sa présentation la plus générale, un service web se concrétise par un agent, réalisé selon une technologie informatique précise, par un fournisseur du service. Un demandeur, à l'aide d'un agent de requête, utilise ce service. Fournisseur et demandeur partagent une même sémantique du service web, tandis qu'agent et agent de requête partagent une même description du service pour coordonner les messages qu'ils échangent

Il existe plusieurs technologies derrière le terme services web :

- Les services web de type *Representational state transfer* (REST) exposent entièrement ces fonctionnalités comme un ensemble de ressources (URI) identifiables et accessibles par la syntaxe et la sémantique du protocole HTTP. Les Services Web de type REST sont donc basés sur l'architecture du web et ses standards de base : HTTP et URI .

2.1. URI, URL, URN :

Le Web est une formidable mine d'informations, de documents, de programmes et de services, bref,
de ressources en tout genre. Il était primordial de définir un mécanisme permettant aux utilisateurs et aux programmes de nommer et de localiser ces ressources. C'est l'objectif des URI (UniformResource Identifier) définis par un standard Internet (*Draft Standard*) proposé par l'IETF sous laréférence RFC2396 (août 1998).
Ce mécanisme d'identification et de localisation est utilisé non seulement par les protocoles de basedu Web que sont HTTP, FTP ou Telnet, mais aussi par la plupart des technologies récentes telles queles espaces de noms (*namespaces*) XML, SMIL, ou SVG. Les enjeux sont suffisamment importantspour qu'un groupe de travail du W3C, en coordination avec l'IETF,

2.1 .1 . Définition URI(Uniform Ressource Identifier) :

Un **URI**, de l'anglais ***Uniform Resource Identifier***, soit littéralement *identifiant uniforme de ressource*, est une courte chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource Web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web . La norme était précédemment connue sous le terme *UDI*.

Les URI sont la technologie de base du World Wide Web car tous les hyperliens du Web sont exprimés sous forme d'URI.

Le principe de URI est d'identifier une ressource de manière permanente, même si la ressource est déplacée ou supprimée.

Les URI sont classés en trois groupes :
• ceux qui permettent de localiser des ressources sur un réseau, appelés URL (Uniform Resource Locator) ;
• ceux qui permettent d'identifier et de nommer des ressources de manière unique et persistante,appelés URN (Uniform Resource Name) ;

• et ceux qui permettent à la fois de localiser et d'identifier une ressource.

2.1.2. **Application deURI**:

Bien que les URI soient très largement utilisés dans le monde informatique, avec surtout les URL sur Internet, on en retrouve d'autres applications dans le monde réel. Ainsi le code ISBN, qui est l'identifiant unique d'un livre, et permet de retrouver celui-ci depuis n'importe quelle librairie ou bibliothèque, dans le monde entier. On peut considérer également lescodes-barres comme une métaphore d'URI, dans le monde physique : un code-barres ne localise pas un produit mais l'identifie (bien qu'il identifie *l'ensemble des exemplaires* d'un produit, pas chaque exemplaire individuellement, ce qui est le travail du numéro de série, lequel n'est pas systématique mais réservé aux produits onéreux).

2.1. 3. Relation avec URL et URN :

Un **URI** peut être de type « *locator* » ou « *name* » ou les deux.

Un **Uniform Resource Locator (URL)** est un URI qui, outre le fait qu'il identifie une ressource sur un réseau, fournit les moyens d'agir sur une ressource ou d'obtenir une représentation de la ressource en décrivant son mode d'accès primaire ou « emplacement » réseau. Par exemple, l'URL *http://www.wikipedia.org/* est un URI qui identifie une ressource (page d'accueil Wikipédia) et implique qu'une représentation de cette ressource (une page HTML en caractères encodés) peut être obtenue via le protocole HTTP depuis un réseau hôte appelé www.wikipedia.org.

Un **Uniform Resource Name (URN)** est un URI qui identifie une ressource par son nom dans un espace de noms. Un URN peut être employé pour parler d'une ressource sans que cela préjuge de son emplacement ou de la manière de la référencer. Par exemple, l'URN *urn:isbn:0-395-36341-1* est un URI qui, étant un numéro de l'*International Standard Book Number* (ISBN), permet de faire référence à un livre, mais ne suggère ni où, ni comment en obtenir une copie réelle.

Le point de vue actuel du groupe de travail qui supervise les URI est que les termes *URL*et *URN* sont des aspects dépendant du contexte des URI, et que l'on a rarement besoin de faire la distinction entre les deux. Dans les publications techniques, spécialement les normes érigées par l'IETF et le W3C, le terme *URL* n'a pas été reconnu pendant longtemps, parce qu'il était rarement nécessaire de faire une distinction entre les URL et les URI. Cependant, dans des contextes non

techniques et dans les logiciels du World Wide Web, le terme *URL* reste omniprésent. De plus, le terme *adresse web*, qui n'a pas de définition formelle, est souvent employé dans des publications non techniques comme synonyme d'URL ou URI, bien qu'il ne se réfère généralement qu'aux protocoles 'HTTP' et 'HTTPS'.

2.1.4 .syntaxe de URI :

Un URI n'utilise qu'un jeu restreint de caractères (chiffres, lettres et quelques symboles) car il doit pouvoir être utilisé tout aussi bien avec des moyens de communication informatisés que non informatisés(papier, etc.). Il est constitué :

• de caractères réservés (« ; », « / », « ? », « : », « @ », « & », « = », « + », « $ », « , ») qui serventde délimiteurs ;

• de chaînes de caractères codés en ASCII US ou à l'aide de séquences d'échappement commençantpar le signe « % » (par exemple : « %2D » qui est le caractère « - »).

2.1.5.URN (Uniform ressource Identifiers):

Les URN sont des *Uniform Resource Identifiers* (URI) et en respectent donc les règles syntaxiques. Les URN ont la syntaxe suivante :

```
urn:NID:NSS
```

- urn est la méthode d'URI des URN.
- *NID* (*Namespace Identifier*) est un identificateur d'espace de noms.
- *NSS* (*Namespace specific String*) est la partie spécifique à l'espace de noms identifié par le *NID*. L'interprétation syntaxique de cette partie dépend de l'espace de noms.

L'usage de minuscules ou de majuscules ne fait pas de différence pour l'écriture de la méthode URN ni pour le *NID*. Il peut en revanche faire une différence pour le *NSS*.

Le *NID* définit un espace de noms. L'Internet Assigned Numbers Authority (IANA) tient un registre des *NID* officiellement enregistrés. Le RFC 3406 donne la marche à suivre pour procéder à un tel enregistrement.

2.1.5.1 . URN et URL :

Lors de la conception du World Wide Web, les *Uniform Resource Locators* (URL) ont été inventées et utilisées pour l'identification des ressources. Mais une URL identifie en fait l'emplacement d'une ressource, plutôt que la ressource elle-même. Ainsi, lorsqu'une ressource est déplacée, par exemple mise sur un autre serveur Web, toutes les URL l'identifiant sont rendues obsolètes. Ce problème est à la base de la plupart des hyperliens « cassés » du Web.

Pour remédier à ce problème, le concept d'URN a été avancé. Par opposition aux URL, les URN identifient les ressources elles-mêmes, indépendamment de leur emplacement. Ce concept nécessite toutefois un mécanisme capable de trouver l'emplacement d'une ressource – par exemple son URL, du moins si elle est accessible sur le réseau – à partir de son URN. Un tel mécanisme repose typiquement sur un répertoire de correspondances.

Dans la pratique, les URN ne sont guère utilisés. Les problèmes de localisation de ressource sont généralement résolus avec unmoteur de recherche. On peut noter la fonctionnalité de « document en cache » qui conserve un certain temps une version du document référencé, indépendamment de son accessibilité à son URL originale.

2.2.MIME (Multipurpose Internet Mail Extensions):

Multipurpose Internet Mail Extensions (*MIME*) ou **Extensions multifonctions du courrier Internet**est un standard internet qui étend le format de données des courriels pour supporter des textes en différents codage de caractères autres que l'ASCII, des contenus non textuels, des contenus multiples, et des informations d'en-tête en d'autres codages que l'ASCII. Les courriels étant généralement envoyés via le protocole SMTP au format MIME, ces courriels sont souvent appelés courriels *SMTP/MIME*.

À l'origine, SMTP avait été prévu pour ne transférer que des fichiers textes (codés en ASCII). Avec l'apparition du multimédia et l'utilisation croissante des applications bureautiques, le besoin s'est fait sentir d'échanger, en plus des fichiers textes, des

fichiers binaires (format des applications bureautiques, images, sons, fichiers compressés).

Les types de contenus définis par le standard MIME peuvent être utilisés à d'autres fins que l'envoi de courriels, dans les protocoles de communication comme le HTTP pour le World Wide Web.

MIME est initialement spécifié dans cinq RFC : RFC 2045, RFC 2046, RFC 2047, RFC 2048, RFC 2049. La RFC 2045est complétée par la RFC 2077. La RFC 2048, maintenant obsolète, est remplacée par les RFC 4288 et RFC 4289.

Le protocole de base de transmission de courriels, SMTP, ne supporte que les caractères ASCII 7-bits. Cela limite les courriels aux messages qui n'incluent que ces caractères, soit un petit nombre de langages comme l'anglais. Les autres langages basés sur l'alphabet latin incluant des diacritiques ne sont pas supportés par l'ASCII 7-bits, ces messages ne peuvent donc pas être correctement représentés dans des courriels basiques.

MIME définit des mécanismes pour l'envoi d'autres sortes d'informations dont des textes dans des langages autres que l'anglais utilisant des codages de caractères autres que l'ASCII et des données binaires comme des fichiers contenant des images, des sons, des films ou des programmes informatiques. MIME est également un composant fondamental des protocoles de communications comme HTTP, qui requièrent l'envoi de données dans le même contexte que l'envoi de courriels, même si ces données ne sont pas des courriels. L'intégration ou l'extraction des données au format MIME est généralement automatiquement effectuée par le client de messagerie ou par le serveur de messagerie électronique quand le courriel est envoyé ou reçu.

Le format de base des courriels est défini dans la RFC 2822 qui est une mise àjour de la RFC 822. Ces standards spécifient le format des en-têtes et du corps des courriels contenant du texte, ainsi que les règles d'en-têtes générales comme « To: », « Subject: », « From: » ou « Date: ». MIME définit un ensemble d'attributs additionnels d'en-têtes de courriels pour le type de contenu du message et son codage. Le codage étant la façon de traduire en ASCII 7-bits, les données 8 bits du message. MIME définit également des règles spécifiques pour encoder des

caractères non ASCII dans les en-têtes de messages, pour, par exemple, autoriser des caractères accentués dans le sujet d'un courriel.

MIME est extensible. Sa définition inclut une méthode pour enregistrer de nouveaux types de contenus ou d'autres valeurs d'attributs.

Un des autres buts explicites de la définition de MIME est de ne pas avoir à changer les serveurs de messagerie électronique préexistants, et de permettre le fonctionnement correct des courriels de base avec les clients préexistants. Ce but est réalisé par le fait que les attributs de messages MIME sont optionnels et que le comportement par défaut est la création d'un message textuel sans MIME qui peut être interprété correctement par un client.

2.2.1. Description d'un message MIME :

MIME introduit des lignes d'en-têtes dans les messages :
• une ligne « MIME-Version: 1.0- » ;
• une ligne « Content-Type » qui précise le format du message ;
• une ligne « Content-Transfer-Encoding » qui précise l'encodage de ce type de contenu.
Deux encodages fondamentaux sont utilisés par les messages MIME :
• Quoted-Printable (ou QP), qui permet de coder n'importe quel jeu de caractères sur 7 bits (parsouci de compatibilité) ;
• Base64, qui permet de coder n'importe quel fichier binaire.
Ces deux encodages ne sont pas obligatoires mais fortement recommandés.

1. HTTP 1.1 (Hyper Text Transfer Protocol) :

- HTTP (HyperText Transfer Protocol) est un standard Internet (Draft Standard) proposé par l'IETF
- sous la référence RFC2616 (dernière RFC datée de juin 1999) et utilisé sur le Web depuis 1990.
- HTTP est un protocole application générique (couche 7, voir en annexe la section « Le modèle deréférence OSI de l'ISO »), qui permet de transférer des messages au format MIME entre un client et un serveur. Il est largement utilisé par de nombreux types de clients (PC, PDA, CD-Rom, etc.), desmoyens

de transport variés (depuis les réseaux sans fil jusqu'aux liaisons optiques transocéaniques)

- et sur des architectures plus ou moins complexes composées de passerelles, de hiérarchies de caches.

2.3.1. **Présentation générale** :

Le protocole HTTP utilise un jeu de *requêtes*/*réponses* entre un client, qui initie le dialogue, et unserveur. La communication peut être directe entre les deux acteurs mais elle peut également faireintervenir trois types d'intermédiaires, que voici :

• un *proxy*, c'est-à-dire un agent qui transfère les messages vers le serveur après en avoir réécrit toutou partie du contenu ;

• une *passerelle*, c'est-à-dire un agent qui agit comme une surcouche pour un serveur sous-jacentutilisant un autre protocole ; cet agent se charge de traduire les messages pour permettre leur transfertvers ce serveur tiers ;

• un *tunnel*, c'est-à-dire un relais qui se charge de transmettre le message entre deux points deconnexion sans modification du message (à travers un intermédiaire tel qu'un pare-feu).

En dehors des tunnels, tous les autres intermédiaires peuvent implémenter des fonctions de cache : ils'agit de garder localement une copie de la réponse tant que celle-ci est valide et de retourner cetteréponse au client sans interroger à nouveau le serveur (ce qui amène un gain de performance et de
trafic réseau).

La connexion établie par le protocole HTTP 1.0 est par défaut volatile : le client ouvre une connexionavec le serveur, envoie une requête et se met en attente de la réponse ; le serveur reçoit la requête, latraite, envoie la réponse et ferme la connexion. Pour garder la connexion ouverte au-delà du traitementde la requête/réponse courante, le client doit explicitement demander le maintien de la connexion(Keep-Alive). À son tour, le serveur doit expliciter sa capacité à maintenir la connexion dans laréponse. En HTTP 1.1, la connexion est persistante par défaut et il faut que le client ou le serveur laferment explicitement.

En HTTP 1.0, le protocole est synchrone et *half-duplex* : sur une connexion, même persistante, leclient envoie une requête et se met en attente de la réponse avant d'envoyer la requête suivante. EnHTTP 1.1, les requêtes peuvent être acheminées en séquence (*pipelining*) sur une connexion sansattendre les réponses respectives. Le serveur doit acheminer les réponses dans le même ordre desrequêtes : la corrélation requête/réponse est maintenue, dans une connexion, par correspondance denuméro d'ordre.

Le protocole HTTP est à l'origine sans état, c'est-à-dire qu'il est incapable de traiter une succession de réponses/requêtes issues du même client comme un dialogue de session : le client envoie une requête, le serveur y répond, mais il n'y a pas de moyen pour communiquer entre client et serveur des

informations sur le contexte du dialogue (l'état de la session). Ce fonctionnement s'est vite avéré problématique pour les sites Internet qui ont souvent besoin de suivre les actions d'un utilisateur au sein d'une session (par exemple, pour une prise de commande). C'est pour cette raison qu'un mécanismede gestion d'état a été ajouté au protocole dans la RFC2965 (Proposed Standard), il introduitde nouveaux en-têtes (*cookies*) qui permettent d'échanger des données d'état tout au long des échanges entre un client et un serveur.

Enfin, HTTP utilise en général TCP/IP sur le port TCP 80 (par défaut), mais en théorie tout autre protocole peut être utilisé.

2.3.2. **Description d'un message http** :

Un message HTTP est soit la requête d'un client, soit la réponse d'un serveur (ou d'un intermédiaire).

Une requête HTTP 1.1 est composée de :

• une ligne de requête qui précise la méthode utilisée, l'URI auquel s'applique cette méthode et laversion du protocole HTTP utilisé ;

• zéro ou plusieurs champs d'en-têtes du type « champ:valeur ». Les en-têtes sont de type *général*,*requête* ou *entité*

• une ligne vide ;

• un corps de message MIME optionnel : la présence ou non de ce contenu dépend de la commandeutilisée. La forme de ce corps de message dépend du type et de l'encodage utilisé (champsContent-Type et Content-Encoding).

Une réponse HTTP 1.1 est composée de :

• une ligne de statut qui précise la version du protocole HTTP utilisé (en général HTTP 1.0), un coderéponse numérique et une description textuelle ;

• zéro ou plusieurs champs d'en-têtes du type « champ:valeur ». Les en-têtes sont de type *général*,réponse ou entité ;

• une ligne vide ;

• un corps de message MIME optionnel : la présence ou non de ce contenu dépend de la commandeutilisée. La forme de ce corps de message dépend du type et de l'encodage utilisé (champsContent-Type et Content-Encoding).

L'indication de la version de protocole utilisée est très importante car elle permet la prise en compted'intermédiaires sur le réseau qui ne prennent pas tous en charge les mêmes versions : la version 1.1assure une compatibilité ascendante avec la version 1.0.

2.3.3.Les en-têtes supplémentaires supportés par HTTP 1.1 :

Connection

Cet en-tête peut être envoyé par le client ou le serveur et contient une liste de noms spécifiant les options à utiliser avec la connexion actuelle. Si une option possède des paramètres ceux-ci sont spécifiés par l'en-tête portant le même nom que l'option (**Keep-Alive** par exemple, pour spécifier le nombre maximum de requêtes par connexion). Le nom **close** est réservé pour spécifier que la connexion doit être fermée après traitement de la requête en cours.

Accept

Cet en-tête liste les types MIME de contenu acceptés par le client. Le caractère étoile * peut servir à spécifier tous les types / sous-types.

Accept-Charset

Spécifie les encodages de caractères acceptés.

Accept-Language

Spécifie les langues acceptées.

L'ordre de préférence de chaque option (type, encodage ou langue) est spécifié par le paramètre optionnel **q** contenant une valeur décimale entre 0 (*inacceptable*) et 1 (*acceptable*) inclus (3 décimales maximum après la virgule), valant 1 par défaut.

Le support des connexions persistantes doit également fonctionner dans les cas où la taille de la ressource n'est pas connue d'avance (ressource générée dynamiquement par le serveur, flux externe au serveur, …).

Pour cela, l'encodage de transfert nommé **chunked** permet de transmettre la ressource par morceaux consécutifs en précédant chacun par une ligne de texte donnant la taille de celui-ci en hexadécimal. Le transfert se termine alors par un morceau de taille nulle, où des en-têtes finaux peuvent être envoyés.

Les en-têtes supplémentaires liés à cet encodage de transfert sont :

Transfer-Encoding

Spécifie l'encodage de transfert. La seule valeur définie par la spécification RFC 2616 est **chunked**.

Trailer

Liste tous les en-têtes figurant après le dernier morceau transféré.

TE

Envoyé par le client pour spécifier les encodages de contenu supportés (**Content-Encoding**, ne pas confondre avec **Transfer-Encoding** car **chunked** est obligatoirement supporté par les clients et serveurs implémentant le standard HTTP/1.1), et spécifie si le client supporte l'en-tête **Trailer** en ajoutant **trailers** à la liste.

2.4. **SMTP (Simple Mail Transfer Protocol**) :

Simple Mail Transfer Protocol (**SMTP**, littéralement « protocole simple de transfert de courrier ») est un protocole de communication utilisé pour transférer le courrier électronique (courriel) vers les serveurs de messagerie électronique.

SMTP est un protocole assez simple (comme son nom l'indique). On commence par spécifier l'expéditeur du message, puis le ou les destinataires d'un message, puis, en général après avoir vérifié leur existence, le corps du message est transféré. Il est

possible de tester un serveur SMTP en utilisant la commande telnet sur le port 25 d'un serveur distant.

Le SMTP commence à être largement utilisé au début des années 1980. Il est alors un complément à l'UUCP, celui-ci étant plus adapté pour le transfert de courriers électroniques entre des machines dont l'interconnexion est intermittente. Le SMTP, de son côté, fonctionne mieux lorsque les machines qui envoient et reçoivent les messages sont interconnectées en permanence.

Le logiciel Sendmail est l'un des premiers, sinon le premier serveur de messagerie électronique à utiliser SMTP. Depuis, la plupart des clients de messagerie peuvent utiliser SMTP pour envoyer les messages. Certains nouveaux serveurs sont apparus, comme Postfix, Qmail de Daniel J. Bernstein, Exim etExchange de Microsoft.

Comme le protocole utilisait du texte en ASCII (7 bits), il ne fonctionnait pas pour l'envoi de n'importe quels octets dans des fichiers binaires. Pour pallier ce problème, des standards comme MIME ont été développés pour permettre le codage des fichiers binaires au travers de SMTP. Aujourd'hui, la plupart des serveurs SMTP acceptent le MIME sur 8 bits, ce qui permet de transférer des fichiers binaires presque aussi facilement que du texte simple.

SMTP utilise TCP pour le transfert des données.

SMTP ne permet pas de récupérer à distance des courriels arrivés dans une boîte aux lettres sur un serveur. Les standards Post Office Protocol (POP) et IMAP ont été créés dans ce but.

2.4.1 . Transmission d'un message :

Le transfert de messages entre serveurs de messagerie électronique se fait généralement sur le port 25 qui est le port standard enregistré auprès de l'IANA. Les serveurs utilisent les *enregistrements MX* des *serveurs DNS* pour acheminer le courrier.

Les *clients de messagerie* utilisaient aussi le port 25 (smtp) pour soumettre des messages en utilisant le protocole SMTP. Mais la nécessité de mieux contrôler les envois des clients, en particulier par l'authentification, a conduit à l'attribution du port 587 (submission)Les administrateurs de serveur peuvent choisir si les clients utilisent

le port TCP 25 (SMTP) ou le port 587 (soumission), tel que formalisé dans la RFC 6409[4] (RFC 2476 précédemment), pour relayer le courrier sortant vers un serveur de messagerie. Les spécifications et de nombreux serveurs supportent les deux. Bien que certains serveurs prennent en charge le port 465 (historique) pour le SMTP sécurisé en violation des spécifications, il est préférable d'utiliser les ports standard et les commandes ESMTP (Extended SMTP) standard selon la RFC 3207, si une session sécurisée doit être utilisée entre le client et le serveur.

2.4.2 .Description du message

SMTP transporte un objet message composé :

• d'une *enveloppe*, elle-même composée d'une série de champs, dont l'adresse de l'émetteur, une ou

plusieurs adresses de destinataires et des données d'extension ;

• d'un *contenu*, composé d'un en-tête et d'un corps de message MIME.

La spécification SMTP ne décrit pas la syntaxe du message qui fait l'objet d'une RFC indépendante

(RFC2822), associée bien sûr à celle de MIME.

2.4.3 .Commandes SMTP

Le transfert des messages est réalisé lors d'un dialogue entre deux serveurs SMTP : respectivementle client qui émet le message (qui peut être la source du message ou un relais) et le serveur qui leréceptionne (qui peut être le destinataire du message ou un relais).

Ce dialogue s'établit dans le cadre d'une session. Le client envoie ensuite une séquence de commandesqui attendent systématiquement une réponse du serveur pour notifier du succès ou de l'échec de lacommande (par exemple : 250 OK).

Une séquence type est la suivante1 :

1. La commande EHLO (ou HELO) initie le dialogue, identifie le client et lui permet de connaître lesextensions SMTP supportées par le serveur.
2. La commande MAIL FROM débute une transaction d'envoi de message en précisant l'adresse del'émetteur.
3. La commande RCPT TO identifie l'(les)adresse(s) des destinataires.

4. La commande DATA envoie, ligne à ligne, le contenu du message. La fin de la transmission et doncde la transaction est indiquée par une ligne contenant uniquement un caractère « . ». Seuls lesmessages transmis dans le cadre d'une transaction valide et complète seront traités par le serveur.

5. La commande QUIT termine la session.

D'autres commandes sont définies par SMTP :

• RESET, qui arrête la transaction en cours ;

• VERIFY, qui vérifie l'argument comme étant une adresse de messagerie ;

• EXPAND, qui vérifie que l'argument est une liste de messagerie et retourne le contenu de cette liste ;

• HELP, qui demande une information ;

• NOOP, qui n'a aucune action si ce n'est une réponse du serveur.

Enfin, des extensions SMTP sont possibles : il s'agit de commandes commençant par un caractère« X » et qui dépendent des logiciels serveurs SMTP utilisés. La commande EHLO permet de connaîtreces extensions.

2.5.**Les protocoles TLS et SSL** :

Transport Layer Security (**TLS**), et son prédécesseur **Secure Sockets Layer** (**SSL**), sont desprotocoles de sécurisation des échanges sur Internet, développés à l'origine par Netscape (SSL version 2 et SSL version 3). Il a été renommé en Transport Layer Security (TLS) par l'IETF à la suite du rachat du brevet de Netscape par l'IETF en 2001. Le groupe de travail correspondant à l'IETF a permis la création des RFC 2246 pour le TLS et RFC 4347 pour son équivalent en mode datagramme, leDTLS. Depuis son rapatriement par l'IETF, le protocole TLS a vécu deux révisions subséquentes : TLSv1.1 décrite dans la RFC 4346 et publiée en 2006 et TLSv1.2, décrite par la RFC 5246 et publiée en 2008.

Malgré le peu de différences entre SSL version 3 et TLS version 1 (qui correspond à la version 3.1 du mécanisme SSL), les deux protocoles ne sont pas interopérables. TLS a tout de même mis en place un mécanisme de compatibilité ascendante avec SSL. En outre, TLS diffère de SSL pour la génération desclés symétriques. Cette génération est plus sécurisée dans TLS que dans SSLv3 dans la mesure où aucune

étape de l'algorithme ne repose uniquement sur MD5 pour lequel sont apparues des faiblesses en cryptanalyse.

Par abus de langage, on parle de SSL pour désigner indifféremment SSL ou TLS.

TLS fonctionne suivant un mode client-serveur. Il fournit les objectifs de sécurité suivants :

- l'authentification du serveur ;
- la confidentialité des données échangées (ou session chiffrée) ;
- l'intégrité des données échangées ;
- de manière optionnelle, l'authentification ou l'authentification forte du client avec l'utilisation d'un certificat numérique ;
- la spontanéité, c'est-à-dire qu'un client peut se connecter de façon transparente à un serveur auquel il se connecte pour la première fois ;
- la transparence, qui a contribué certainement à sa popularité : les protocoles de la couche d'application n'ont pas à être modifiés pour utiliser une connexion sécurisée par TLS. Par exemple, le protocole HTTP est identique, que l'on se connecte à un schème http ou https.

2.5.1. **Introduction à la sécurité** :

Les techniques mises en oeuvre pour assurer la sécurité des échanges ont pour objectif :

• pour l'émetteur, de crypter ses données et pour le récepteur, de les décrypter ;

• de contrôler l'intégrité des informations reçues ;

• d'authentifier l'émetteur du message ;

• d'obliger l'émetteur à reconnaître l'émission des informations grâce à un mécanisme de nonrépudiation

Le chiffrement/déchiffrement de l'information est réalisé à l'aide d'un algorithme (ou *cipher* enanglais) : l'intérêt de cette technique est que la sûreté du chiffrement ne repose pas sur la méthode decalcul utilisée, qui est connue et publiée, mais sur

l'utilisation de chiffres, appelés *clés*, qui permettentà l'algorithme de générer un document crypté, puis de le décrypter. Plus ces clés sont grandes (en nombre de bits), plus il est difficile, voire « impossible », de décrypter un document si l'on ne connaîtpas les chiffres qui ont permis sa génération.

Deux types de clés sont utilisés :

• Les clés *symétriques* : comme leur nom l'indique, la même clé est utilisée pour crypter et décrypterles données. Cette technique est rapide et fournit en outre un moyen d'authentification. Elle estnéanmoins un peu sensible, puisque toute la sécurité repose sur la connaissance d'une clé quel'émetteur et le récepteur doivent garder secrète.

• Les clés *publiques* (ou asymétriques) : le principe consiste à crypter les données avec une clépublique, c'est-à-dire connue de tout le monde, et de les décrypter avec une clé privée connueuniquement par le récepteur. Les deux clés publique/privée fonctionnent par paire et dans les deuxsens puisque, à l'inverse, la clé publique peut décrypter ce qui a été généré à l'aide de la clé privée.

Cette technique est plus lourde et n'offre pas de mécanisme d'authentification de l'émetteur. Enrevanche, elle permet d'authentifier le récepteur qui peut crypter sa signature à l'aide de sa clé privée.

Le contrôle d'intégrité d'un message est réalisé à partir d'une *signature électronique*, elle-mêmecryptée, et qui est le résultat d'une fonction de hachage.

Une fonction de hachage appliquée à des données fournit un chiffre unique : la moindre altération deces données modifie obligatoirement le résultat du hachage.

Le résultat d'une fonction de hachage ne permet pas de déterminer les données qui ont permis sagénération.

Un *certificat* est un document électronique qui permet d'identifier une entité, c'est-à-dire un utilisateur,une entreprise, un serveur, etc. Il est associé à la clé publique de l'entité qu'il identifie. Il est égalementlié à une *autorité de certification* (CA ou Certification Authority) : il s'agit d'une application serveurqui peut être spécifique à une entreprise, ou gérée par une organisation tierce (telle que Verisign : *http://www.verisign.com*). Le rôle de cette autorité est de générer le certificat et de proposer à l'utilisateur quile reçoit une fonction de validation.

Les certificats, associés aux signatures électroniques, sont utilisés pour authentifier à la fois le clientet le serveur dans le cadre d'une connexion SSL, mais aussi des e-mails (S/MIME), du code Java ouJavaScript, etc.

2.5.2. **Présentation générale :**

SSL et TLS sont des protocoles qui se situent à un niveau intermédiaire, entre la couche transport etla couche application. Cette position permet donc a priori à toutes les applications qui s'appuient surTCP/IP (HTTP, SMTP, Telnet, etc.) d'utiliser les fonctionnalités de SSL/TLS, soit :

• permettre à un serveur de s'authentifier auprès d'un client, c'est-à-dire au client de vérifier l'identitédu serveur ; • optionnellement, permettre au client de s'authentifier auprès du serveur, c'est-à-dire au serveur devérifier l'identité du client ;

• permettre au client et au serveur de sélectionner un algorithme de chiffrement ;

• permettre aux deux machines d'établir une connexion cryptée pour assurer un haut niveau deconfidentialité.

Ces protocoles sont décomposés en deux sous-protocoles :

• le protocole d'enregistrement (*record protocol*), qui définit le format de transmission des données ;

• le protocole de négociation (*handshake protocol*), qui s'appuie sur le protocole d'enregistrement,et qui est chargé d'établir la connexion entre le client et le serveur (authentification, sélection de laméthode de chiffrement, etc.).

2.5.3.***Les méthodes de chiffrement (cipher)***

Tous les échanges de données réalisés dans le cadre d'une connexion SSL/TLS, y compris les messagesinitiaux gérés par le protocole de négociation, sont cryptés. Différents algorithmes mathématiques,classés en suites de chiffrement, sont pris en charge par SSL/TLS. Ces algorithmes sont très nombreuxmais les principaux sont les suivants :

• DES (Data Encryption Standard), un algorithme de chiffrement à base de clés symétriques créé parIBM en 1977 et utilisé par le gouvernement américain avant AES (FIPS 46-3, ANSI X3.92 et X3.106) ;

• Triple-DES, l'algorithme DES appliqué trois fois ;

• AES (Advanced Encryption Standard) est le nom du projet lancé en 1997 par le NIST pour trouverun remplaçant à DES. En octobre 2000, l'algorithme de chiffrement à base de clés symétriques*Rijndael*, créé par Joan Daemen et Vincent Rijmen, a été retenu et est devenu un standard fédéralaméricain (FIPS 197) ;

• IDEA (International Data Encryption Algorithm), un algorithme de chiffrement à base de cléssymétriques, créé par Xuejia Lai et James Massey et considéré comme très efficace (brevetinternational détenu par la société Ascom-Tech) ;

• MD5 (Message Digest), un algorithme d'empreinte numérique développé en 1991 par le professeurRonald Rivest du MIT (RFC1321) ;

• RC2 et RC4 (Ron's Code), qui sont des algorithmes de chiffrement développés par le professeurRonald Rivest du MIT pour la société RSA Security (brevet international) ;

• RSA, qui est un algorithme à base de clé publique développé en 1977 par Rivest, Shamir etAdleman. Il est utilisé à la fois pour le chiffrement et l'authentification. (brevet US n°4405829,tombé dans le domaine public depuis septembre 2000) ;

• SHA-1 (Secure Hash Algorithm), un algorithme d'empreinte numérique développé en 1994 etutilisé par le gouvernement américain ;

• SKIPJACK, un algorithme à base de clé symétrique implémenté dans des systèmes matérielscompatibles FORTEZZA (tels que la puce Clipper) et utilisé par le gouvernement américain

2.5.4.*Le protocole de négociation (handshake)*

Le protocole de négociation correspond à un échange de messages réalisé entre le serveur et le clientlors de l'ouverture d'une session SSL/TLS. Cet échange est primordial puisqu'il permet au serveurde s'identifier grâce aux techniques de clés privées (l'identification du client est optionnelle), de fixerles paramètres de la session et de définir les clés symétriques qui seront utilisées pour le chiffrement/
déchiffrement. Il peut se résumer ainsi :

• Le client envoie au serveur sa version SSL/TLS, ses paramètres de chiffrement et toutes les donnéesnécessaires à l'établissement de la session.

• En retour, le serveur envoie sa version SSL/TLS et ses paramètres de chiffrement ainsi que soncertificat. Si cela est nécessaire, il demande au client son certificat pour pouvoir l'authentifier.

• Le client procède aux tests d'authentification : date de validité, contrôle de l'autorité de certification,contrôle de la clé publique, etc. La validité de ces tests détermine si la session peut se poursuivreou non.

• Le client crée la clé préliminaire (*premaster*), la crypte à l'aide de la clé publique du serveur et lalui transmet. Si le serveur demande l'identification du client, le client lui envoie aussi son certificat.

• Le serveur décrypte les données transmises par le client, notamment la clé préliminaire, à l'aide desa clé privée. Si cela est nécessaire, le serveur procède aux tests d'authentification : date de validité,contrôle de l'autorité de certification, contrôle de la clé publique, etc. La validité de ces testsdétermine si la session peut se poursuivre ou non.

• Le client et le serveur calculent l'un et l'autre une clé principale (*master*) à l'aide de la clé préliminaire.

Cette clé principale est utilisée pour calculer les deux clés de sessions symétriques : il y a une clépour chaque sens de transmission, qui est utilisée à la fois pour crypter et décrypter les donnéestransférées.

• Les deux parties s'informent mutuellement de la fin des opérations et mettent fin au protocole denégociation.

2.6.conclusion

Dans ce chapitre nous avons présenté l'aspect technologie des services web et les protocoles internet , Dans le chapitre suivant nous évoquerons les technologies XML

Chapitre III

les technologies XML

3.Introduction :

L'***Extensible MarkupLanguage*** (**XML**, « langage de balisage extensible » en français) est un langage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG… Elle est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche…) entre systèmes d'informations hétérogènes (interopérabilité). Avec ses outils et langages associés, une application XML respecte généralement certains principes :

la structure d'un document XML est définie et validable par un schéma, un document XML est entièrement transformable dans un autre document XML.

L'objectif initial de XML est expliqué au début de la spécification du 10 février 1998, la phrase est toujours d'actualité : « Son but est de permettre au SGML générique d'être transmis, reçu et traité sur le web de la même manière que l'est HTML aujourd'hui. »**(fr)**SGML est un langage de balisage, employé dans les industries de la documentation et de l'édition. En adoptant cette syntaxe pourHTML, Tim Berners-Lee confrontait une technologie complexe à de plus en plus d'utilisateurs. L'objectif d'XML était de définir un langage aussi générique, mais plus simple : « *XML has been designed for ease of implementation* »**(en)**, « XML a été conçu pour une facilité de mise en œuvre »**(fr)**.

À la lumière des années passées, cette spécification a rempli l'objectif qu'elle se fixait, XML a été largement suivi et favorise l'interopérabilité. La disponibilité d'une syntaxe standard et d'outils de manipulation réduit

3.1.Rappel des règles de base :

Les règles syntaxiques d'XML sont simples mais strictes, ainsi, un programme qui traite un documentXML doit s'arrêter à la première erreur.
On dit d'un document XML qu'il est *bien formé* s'il respecte les règles syntaxiques imposées et ainsirésumées :

• Un document XML doit commencer par une ligne de déclaration ne serait-ce que pour préciser laversion d'XML. Exemple :

<? xml version="1.0"?>

• Les éléments qui composent un document XML doivent être encadrés par une balise ouvrante etune balise fermante. Exemple :

<para> Ceci est un paragraphe </para>

• Les noms de balises sont sensibles à la casse des caractères. Exemple :

<Para> Ceci est correct </Para>

<para> Ceci est incorrect </Para>

• Tous les éléments doivent être correctement encadrés entre eux. Exemple :

<para><texte> Ceci est correct </texte></para>

<para><texte> Ceci est incorrect </para></texte>

• Un document XML possède toujours une racine qui est définie par la première balise rencontréedans le traitement. Tous les éléments du document sont encadrés par cette racine. Exemple :

<racine>

<element1>

<souselement1> Sous élément du premier élément </souselement1>

</element1>

<element2> Second élément </element2>

</racine>

• Les éléments peuvent être dotés d'attributs. Exemple :

<Para monattribut="mavaleur"> Élément avec comme attribut monattribut de valeur mavaleur</Para>

• Les valeurs des attributs doivent toujours être encadrées par des quotes (simples ou doubles).

Exemple :

<Para date="maintenant"> Ceci est correct </Para>

<Para date=maintenant> Ceci est incorrect </Para>

•Les commentaires sont définis par la balise <!-- et -->.

On dit d'un document qu'il est *valide* s'il respecte une certaine description : ces descriptions sontétablies par des DTD (Document Type Definition) ou des schémas (documents XML décrivantd'autres documents XML), internes ou externes.

3.2.Un document XML

Le corps d'un document XML est constitué d'un ou plusieurs éléments délimités par des balisesouvrantes et fermantes. Ces éléments sont organisés entre eux dans une structure arborescente.

Dans l'exemple suivant :

<debut>

<element1> Premier element </element1>

<element2> Second element </element2>

</debut>

• debut est la racine du document, ;

• element1 et element2 sont les fils de début ;

• debut est le père d'element1 et element2 ;

• element1 et element2 sont frères.

Les éléments possèdent un contenu délimité par les balises. Ce contenu peut être :

• simple s'il s'agit de texte uniquement ;

• mixte si l'élément possède à la fois un contenu simple et d'autres éléments ;

• vide : dans ce cas, la balise ouvrante est aussi fermante. Exemple :
<saut_de_page/>, qui est équivalent
à <saut_de_page></saut_de_page>.

Remarque

Tout le contenu d'un élément, c'est-à-dire tout ce qui est entre la balise ouvrante et fermante, est analysé par lesprogrammes à la recherche d'autres éléments. Il existe cependant un moyen d'indiquer que le contenu est simpleet ne nécessite pas d'analyse en utilisant une section CDATA. Exemple :

<texte><![CDATA[Ceci est un <contenu> simple]]></texte>

Le nommage des types d'éléments (c'est-à-dire des balises) doit suivre les règles suivantes :

• Le nom peut contenir des lettres, des chiffres ou tout autre caractère autorisé (voir énumérationUnicode dans la spécification : *http://www.w3c.org/TR/2000/REC-xml-20001006 - NT-Name*).

• Le nom ne doit pas commencer par un chiffre ni un caractère de ponctuation.
• Le nom ne doit pas commencer par xml (quelle que soit la casse).
• Le nom ne doit pas contenir d'espaces.

Mis à part ces quelques restrictions, seules les règles de bons sens prévalent pour nommer deséléments (le nom doit être clair, précis et concis).

Un document XML est extensible, ce qui signifie qu'il est possible d'ajouter des éléments XML sansque cela remette en cause le traitement du document, pourvu que les éléments existants demeurentinchangés (auquel cas, il ne s'agit plus d'une extension).

Les éléments XML peuvent posséder des attributs. qui permettent d'ajouter des informationssupplémentaires aux éléments. Exemple :

<image type="JPEG"> mon image.jpg </image>

Il n'y a pas de règle pour déterminer précisément si une information doit être traitée en tant qu'attributou en tant qu'élément. Cependant, en général, les attributs sont utilisés en tant que métadonnées, pourqualifier le contenu des éléments. Ce qui est certain, c'est que l'usage des attributs est beaucoupmoins souple que celui des éléments : ils sont difficilement extensibles, leur contenu est simple, etc.

3.3.**XML namespaces**

Les espaces de noms XML ou *namespaces* sont une extension de la recommandation XML qui a étépubliée en janvier 1999 par le W3C. À l'origine de cette extension, il y a la volonté d'introduire lamodularité dans les documents XML et de permettre la réutilisation de tout ou partie des documents

3.4. L'attribut xmlns ou xmlns:

Un espace de noms XML identifie une collection de noms qui sont utilisés dans un document XMLpar les éléments et les attributs.

La déclaration d'un espace de noms XML est réalisée à l'aide de l'attribut réservé xmlns ou d'un attributspécifique précédé du préfixe xmlns:. La *valeur* de cet attribut, une référence d'URI, est le *nom* del'espace de noms. Par exemple :

<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">

est équivalent à :

<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">

et déclare l'espace de noms *http://schemas.xmlsoap.org/soap/envelope/*.
Dans le premier exemple, xmlns: non seulement déclare un espace de noms, mais aussi définit un *préfixe*.
Tout élément ou attribut dont le nom appartient à l'espace de noms doit être précédé de ce préfixe. Parexemple :

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
soap-env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<soap-env:Body>
</soap-env:Body>
</soap-env:Envelope>
```

existants. Pour atteindre cet objectif, il est nécessaire de doter XML d'un mécanisme permettantd'éviter toute ambiguïté de nommage (problèmes de collisions de noms d'éléments ou d'attributs).

Attributs et espaces de noms

Il faut noter, dans l'exemple précédent, qu'encodingStyle est un attribut de l'espace de noms *http://schemas*
.xmlsoap.org/soap/envelope/.
Bien évidemment, il est possible de définir plusieurs espaces de noms dans un même document ainsiqu'un espace de noms par défaut : les règles de portée sont assez logiques et s'appuient sur la hiérarchiedu document. Par exemple :

```
<?xml version="1.0"?>
<!-- définition de l'espace www.eyrolles.com préfixé par pref -->
<pref:livre xmlns:pref="http://www.eyrolles.com/">
<!-- l'espace de noms par défaut est HTML -->
<table xmlns="http://www.w3.org/TR/REC-html40">
<tr>
<td><!-- plus d'espace de noms -->
<titre xmlns="">Ceci est le titre</titre>
<!-- retour à l'espace par défaut HTML -->
</td>
<td>
<!-- espace de noms www.eyrolles.com -->
<pref:sujet>les services Web</pref:sujet>
```

```
<!-- retour à l'espace par défaut HTML -->
</td>
</tr>
</table>
</pref:livre>
```

3.5. XLINK :

XML *Linking Language*(XLink) version 1.0 est une recommandation du W3C publiée en juin 2001.
L'objectif de cette spécification est de fournir un mécanisme permettant de créer et de décrire, dansdes documents XML, des liens entre des ressources. Ces liens peuvent être unidirectionnels ou desstructures plus complexes.
Cetterecommandation est complémentaire à Xpointer, laquelle fournit unmécanisme de définition d'adresses.

Un peu de vocabulaire

Un lien (*link*) est une relation explicite entre des ressources ou des parties de ressources. Il est renduexplicite par un élément liant (*linking element*), qui est l'élément du document XML qui décrit celien. Une utilisation courante des liens est celle des *hyperliens*, qui sont des liens destinés à êtreprésentés à un utilisateur humain.
Une *ressource* est définie comme étant une unité d'information. Elle est identifiée par un URI. Lorsqu'unlien associe un ensemble de ressources, on dit que ces ressources participent (*participate*) au lien.
Un lien *simple* implique une paire de ressources : la ressource de départ (*starting resource*) et laressource d'arrivée (*ending resource*). Un lien *étendu* implique quant à lui un nombre quelconque deressources. L'information concernant la manière de traverser une paire de ressources, c'est-à-dire ladirection et le comportement, est appelée un *arc*.
Une *ressource locale* est un élément XML qui participe à un lien en ayant un parent ou en étant lui-mêmeun élément liant. Une ressource qui participe à un lien en étant adressée par un URI est considéréecomme une ressource distante (*remote*) même si

elle se trouve dans le même document que l'élémentliant. L'élément HTML A est un lien simple dont la ressource de départ est une ressource locale et
dont la ressource d'arrivée est distante.
Un arc qui a une ressource de départ locale et une ressource d'arrivée distante est hors ligne (*outbound*),c'est-à-dire qu'il quitte l'élément liant. L'élément HTML A possède donc un arc hors ligne.
Si la ressource d'arrivée d'un arc est locale mais sa ressource de départ distante, alors l'arc est enligne (*inbound*). Si aucune des deux ressources n'est locale, il s'agit d'un arc tiers (*third-party*).
Enfin, les documents qui contiennent des collections de liens en ligne et/ou de liens tiers se nommentbases de liens (*linkbase*).
Ces derniers concepts sont plus complexes mais néanmoins très puissants : Xlink permet de créer desliens depuis des ressources sans qu'aucune action ne soit nécessaire sur cette ressource (distante),
c'est-à-dire sans qu'il ne soit nécessaire de modifier le document contenant cette ressource. C'est lesens des liens *en ligne*. Il est même possible de décrire totalement un lien dans un fichier externe(base de liens) sans qu'aucune des ressources participantes ne fasse partie de ce fichier. C'est le sens
des liens *tiers*. Ce principe est très utile lorsque le nombre de ressources est élevé ou lorsque cesressources sont en lecture seule, ou inaccessibles, et de manière générale lorsqu'il est trop coûteux demodifier ces ressources et plus intéressant de gérer les liens indépendamment.

3.5.1. *La syntaxe Xlink*

L'usage d'Xlink nécessite dans un premier temps la définition de l'espace de noms *http://www.w3.org/1999/xlink*. En général, le préfixe utilisé est xlink. Par exemple :
<monlien xmlns:xlink=" http://www.w3.org/1999/xlink "> … </monlien>
Cet espace de noms identifie un ensemble d'attributs qui permet de définir les liens Xlink. Le principalattribut est type, qui permet de définir le type de lien. Il y a, en effet
• des liens simples (type="simple") qui mettent en relation de façon unidirectionnelle une ressourcelocale et une ressource éloignée, ce sont typiquement les liens HTML A ou IMG ;

• des liens étendus (type="extended") qui utilisent pleinement les fonctionnalités de Xlink enpermettant la création de liens multidirectionnels, de liens en ligne ou tiers. Cet attribut permet également de qualifier d'autres éléments qui servent à la déclaration des liens étendus,comme :

• des ressources locales (type="resource") ;

• des ressources éloignées (type="locator") ;

• des règles de traversée (type="arc") ;

• des titres (type="title").

3.6. XML Base

XML Base 1.0 est une recommandation du W3C publiée en juin 2001. Son objectif est de définirdans un document XML un chemin de base permettant d'interpréter de façon relative tous les URIcontenus dans le document et implémentés en XLink. L'objectif de cette spécification est équivalentà celui d'HTML Base.

L'attribut xml:base

Le principe de fonctionnement de XML Base est simple. Il consiste à ajouter un attribut xml:base àn'importe quel noeud d'un document XML. La valeur de cet attribut est un URI de base utilisé partous les liens Xlink exprimés dans le noeud. Par exemple :

```
<? XML version="1.0" encoding="ISO-8859-1" ?>
<carnet_adresses xml:base="http://www.monSite.com/commun/"
xmlns:xlink="http://www.w3.org/1999/xlink">
<!-- http://www.monSite.com/commun/identite.xml -->
<titre>Mon carnet
<link xlink:type="simple" xlink:href="identite.xml">personnel</link>
</titre>
<liste xml:base="/adresses/">
<adresse id="1">
<nom>
<!-- http://www.monSite.com/adresses/dupont.xml -->
<link xlink:type="simple" xlink:href="dupont.xml">Dupont</link>
</nom>
```

<prenom>Bernard</prenom>
<cp>75014</cp>

</adresse>
</liste>
</carnet_adresses>

3.7.XPath

XML *Path Language* 1.0 (XPath) est une recommandation du W3C publiée en novembre 1999.
L'objectif de cette spécification est de fournir un mécanisme permettant d'adresser des parties dedocument XML, mécanisme utilisé à la fois par XSLT et XPointer. Le nom de cette technologie vient dufait qu'elle utilise des chemins (*path*) équivalents aux URL pour naviguer dans la structure hiérarchiquedes documents XML.
Par exemple, pour le document XML suivant :

```
<? XML version="1.0" encoding="ISO-8859-1" ?>
<carnet_adresses>
<adresse id="1">
<nom>Dupont</nom>
<prenom>Bernard</prenom>
<cp>75014</cp>
</adresse>
<adresse id="2">
<nom>Durand</nom>
<prenom>Paul</prenom>
<cp>06220</cp>
</adresse>
<adresse id="3">
<nom>Vincent</nom>
<prenom>Pierre</prenom>
<cp>21017</cp>
</adresse>
</carnet_adresses>
```

3.8.Les expressions XPath

Les documents XML peuvent être représentés comme des arbres de noeuds appartenant à un des sept
types suivants :
• le type racine, qui est utilisé pour la racine du document ;
• le type élément, qui est utilisé pour les éléments d'un document. Le nom d'un élément peut êtreexprimé en précisant un espace de noms ;
• le type texte, qui est utilisé pour les valeurs d'éléments données caractères (y compris <![CDATA[…]]>) ;
• le type attribut, qui est utilisé pour les attributs d'un élément ;
• le type espace de noms, qui est utilisé pour les attributs ou éléments affectés par la déclaration d'unespace de noms (attribut xmlns ou préfixé xmlns:) ;
• le type instruction, qui est utilisé pour les instructions XML <? … ?> (mis à part l'instruction dedéclaration XML en en-tête qui ne possède pas de noeud) ;
• le type commentaire, qui est utilisé pour les commentaires <!-- … -->.
Une expression XPath permet d'obtenir un ensemble de noeuds, de n'importe quel type, du documentXML. Cette expression se traduit sous la forme d'un chemin qui peut être :
• absolu s'il commence par un « / », le chemin identifie alors de manière constante un ensemble denoeuds à partir de la racine du document ;
• relatif lorsqu'il ne commence pas par un « / », le résultat dépend alors de l'endroit où l'expressionest appliquée lors de la navigation dans l'arbre.

3.9.XML Schema

XML Schema 1.0 est une recommandation du W3C qui a été publiée en mai 2001. L'objectif de cettespécification est de fournir un mécanisme de description et de validation des documents XML, équivalentà la DTD mais plus expressif, extensible et surtout utilisant lui-même une syntaxe XML et les espaces de noms.

Globalement, XML Schema permet de créer un modèle de document (un schéma) qui définit :

• les éléments et les attributs qui peuvent apparaître dans le document ;
• l'ordre et l'occurrence des éléments fils ;
• si un élément est vide ou s'il a un contenu ;
• les types de données des éléments et des attributs ;
• les valeurs par défaut des éléments et des attributs.

3.9.1.Description d'un schéma XML

Un schéma XML est d'abord un document XML 1.0 bien formé et valide au regard de ses spécifications,c'est-à-dire soit en utilisant le schéma des schémas, soit la DTD des schémas (respectivement dansles annexes A et G de la recommandation du W3C).
Le vocabulaire (noms d'attributs et d'éléments) est défini dans deux espaces de noms :
• *http://www.w3.org/2001/XMLSchema* : cet espace de noms définit la plus grande partie du vocabulaire
d'XML Schema. Pour simplifier la suite de ce chapitre, on utilisera systématiquement le préfixe
xsd: pour référencer cet espace de noms.
• *http://www.w3.org/2001/XMLSchema-instance* : cet espace de noms définit des attributs qui peuvent êtreutilisés dans tout document XML (type, null schemaLocation et noNamespaceSchemaLocation). Poursimplifier la suite de ce chapitre, on utilisera systématiquement le préfixe xsi: pour référencer cet
espace de noms.Le schéma est un modèle qui peut être appliqué à des instances de documents XML. En général, unmodèle est fait pour être réutilisé et il est préférable de le confier à un document à part plutôt que dele voir défini dans le corps du document XML auquel il s'applique. Le lien entre le document XML
et le schéma est réalisé à partir des attributs xsi:schemaLocation ou xsi:noNamespaceSchemaLocationqui permettent de référencer l'URL du schéma. Ce lien implique qu'un travail de validation devraêtre effectué sur l'instance de document par un programme adéquat (voir la section « Les analyseurs
syntaxiques XML »).
Un schéma XML peut être réparti dans plusieurs documents. Deux mécanismes d'inclusion sont fournis :

• <xsd:include>, qui permet d'inclure un schéma et d'utiliser telles quelles les définitions et les
déclarations de ce schéma ;
• <xsd:redefine>, qui permet non seulement d'inclure un schéma mais aussi de redéfinir les types dece schéma par restriction ou extension.
Dans les deux cas, l'attribut schemaLocation permet de spécifier l'URL du document à inclure. Deplus, ce schéma externe doit posséder le même espace de noms cible que le schéma dans lequel il estinclus.
Ce mécanisme d'inclusion est très important puisqu'il permet la modularité : au fur et à mesurequ'XML Schema prend de l'essor, des bibliothèques de schémas se créent et peuvent être réutiliséespar les développeurs et les concepteurs.
En tant que document XML, un schéma possède une racine qui est obligatoirement l'élément <schema>.
Par exemple :

```
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.eyrolles.com"
…
</xsd:schema>
```

Cet élément possède plusieurs attributs spécifiques :
• targetNamespace, qui permet de spécifier l'espace de noms cible des éléments et des attributs définis
par ce schéma ;
• attributeFormDefault="qualified" ou "unqualified", qui permet de préciser si les noms des attributsdoivent être qualifiés ou pas à l'aide du nom d'espace de noms cible ;
• elementFormDefault="qualified" ou "unqualified", qui permet de préciser si les noms des élémentsdoivent être qualifiés ou pas à l'aide du nom d'espace de noms cible, etc.
Un schéma XML est constitué d'un ensemble de composants :
• de définition, qui servent à créer de nouveaux types, simples ou complexes, par dérivation de typesexistants ;
• de déclaration, qui permettent de spécifier les noms et les types de contenus des éléments et des attributs qui pourront être utilisés dans les instances de document XML.

3.9.2.Les composants de déclaration

Les composants de déclaration permettent de définir :
• des éléments grâce à l'élément <xsd:element> ;
• des attributs grâce à l'élément <xsd:attribute> ;
• des notations grâce à l'élément <xsd:notation>.
Les attributs sont identifiés par leur nom (attribut name) et ne peuvent être que de types simples. Ladéclaration comporte d'autres informations utiles :
• default indique une valeur par défaut ;
• fixed indique une valeur fixe et par défaut ;
• use indique si l'attribut est optionnel ("optional"), interdit ("prohibited") ou obligatoire ("required").
Par exemple :

```
<xsd:attributename="pays" type="xsd:NMTOKEN" fixed="FR"/>
```

Les éléments sont identifiés par leur nom (attribut name) et peuvent être de n'importe quel type, simpleou complexe. La déclaration comporte d'autres informations utiles :
• default indique une valeur par défaut ;
• fixed indique une valeur fixe et par défaut ;
• maxOccurs précise le nombre d'occurrences maximal de l'élément ;
• minOccurs précise le nombre d'occurrences minimal de l'élément (0 signifie optionnel, sinonobligatoire).
Par exemple :

```
<xsd:element name="codePostal" minOccur="1">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:pattern value="\d{5}"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

Un attribut particulier, substitutionGroup, permet d'utiliser un mécanisme intéressant de substitution :

le principe est de définir un groupe d'éléments pouvant se substituer à un élément global appeléélément de tête. La contrainte est que les groupes de substitution doivent être du même type ou d'untype dérivé de l'élément de tête.
Dans l'exemple suivant, l'élément codePostalUS peut être substitué à l'élément de tête codePostalpartout où celui-ci est utilisé :

```
<xsd:element name="codePostal" type="xsd:string"/>
<xsd:element name="codePostalUS" type="xsd:string"
substitutionGroup="codePostal">
```

Si les déclarations sont effectuées directement à l'intérieur de <xsd:schema>, elles sont *globales*,sinon (si elles sont déclarées à l'intérieur de types complexes), elles sont *locales*. Ce qui est intéressantavec les déclarations globales, c'est qu'elles peuvent être réutilisées par d'autres déclarations à l'aide
de l'attribut ref. Par exemple :

```
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.eyrolles.com"
<xsd:element name="codePostal">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:pattern value="\d{5}"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="adresse">
<xsd:element name="nom" type="xsd:string">
<xsd:element name="prenom" type="xsd:string">
<xsd:element name="adresse" type="xsd:string">
<xsd:element name="CP" ref="codePostal" minOccur="1">
</xsd:element>
</xsd:schema>
```

3.9.3.Les composants de définition

Les composants de définition permettent de définir des types qui peuvent être réutilisés dans d'autrescomposants. Les définitions de types sont hiérarchisées dans

la mesure où toute définition de type estune extension ou une restriction d'une autre définition de type. La définition du type *ur-type*, dont le
nom est anyType, est la racine de cette hiérarchie.
XML Schema permet de définir deux catégories de types, à savoir simples et complexes.

Définition de types simples

Les types simples s'appliquent aux valeurs d'attributs et au contenu textuel d'éléments (qui n'ont pasd'éléments enfant). Un type simple est obligatoirement une restriction d'un type de base simple. XMLSchéma fournit un certain nombre de types prédéfinis qui sont utilisés pour créer les types utilisateur

3.9.4.Les définitions complémentaires

XML Schema permet de définir une contrainte d'unicité pour des valeurs d'attributs ou d'éléments.
Ce mécanisme s'appuie sur l'élément <xsd:unique> et les éléments fils suivants :
• <xsd:selector>, dont l'attribut xpath permet de spécifier un chemin XPath définissant le périmètreauquel s'applique la contrainte ;
• <xsd:field>, dont l'attribut xpath permet de spécifier un chemin XPath définissant les éléments oules attributs dont la valeur doit être unique au sein du périmètre précisé par <xsd:selector>.
Reprenons l'exemple XML suivant :

```
<? XML version="1.0" encoding="ISO-8859-1" ?>
<carnet_adresses>
<adresse id="1">
<nom>Dupont</nom>
<prenom>Bernard</prenom>
<cp>75014</cp>
</adresse>
<adresse id="2">
<nom>Durand</nom>
<prenom>Paul</prenom>
<cp>06220</cp>
</adresse>
```

```
<adresse id="3">
<nom>Vincent</nom>
<prenom>Pierre</prenom>
<cp>21017</cp>
</adresse>
</carnet_adresses>
```

3.10.L’interface DOM

L’objectif du Document Object Model (DOM) est de fournir une interface de programmation (API)standardisée, indépendante de la plate-forme et des langages, pour accéder et mettre à jour le contenuet la structure de documents HTML et XML. Une telle interface permet par exemple à un programme
de script de manipuler une page HTML indépendamment du navigateur.
Il existe plusieurs recommandations :

• DOM Level 1 : recommandation d’octobre 1998 qui se concentre sur les modèles de documentXML et HTML, et fournit des fonctionnalités de navigation et de manipulation Une seconde édition est en cours d’élaboration

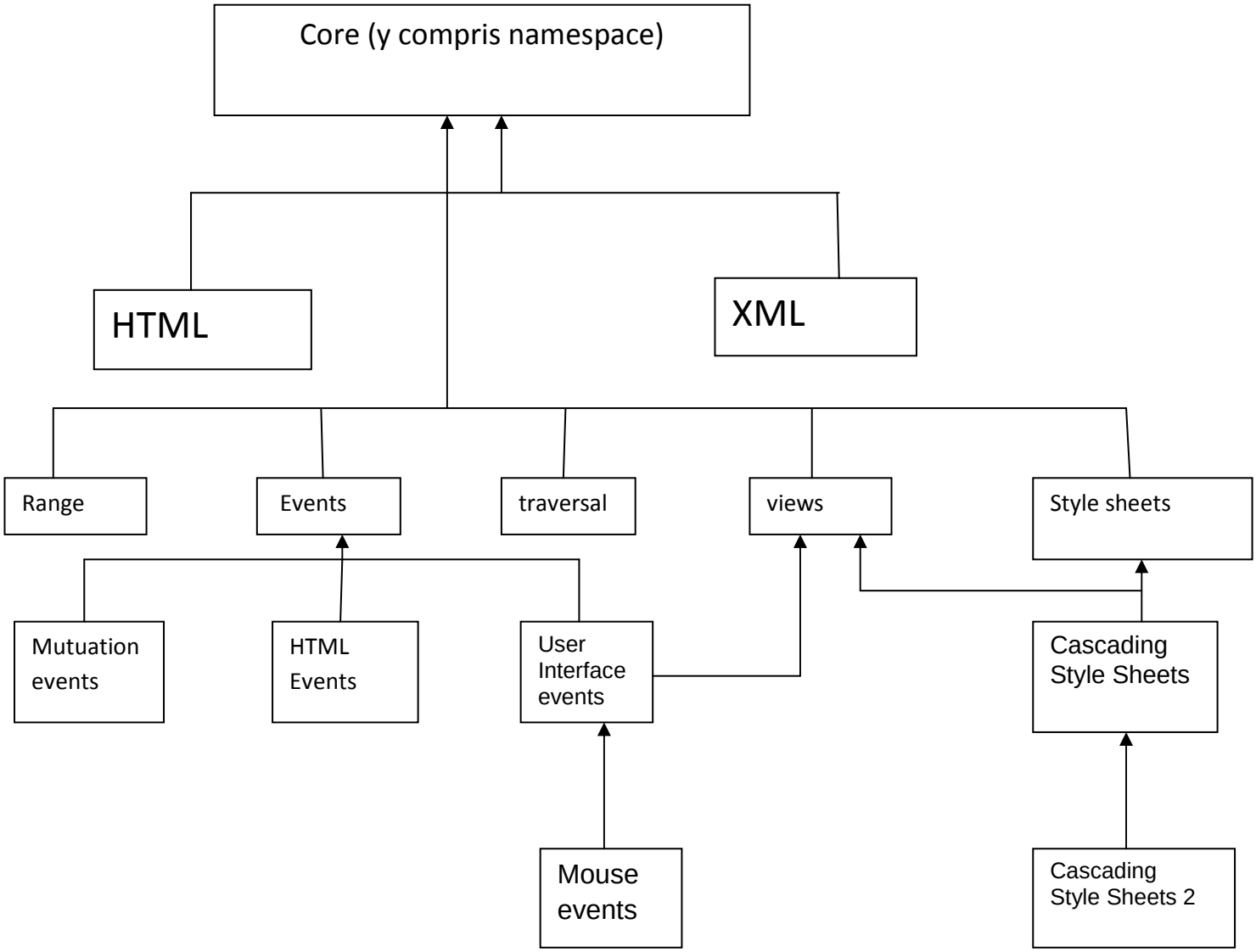


Figure 3.1:***Organisation de DOM Level 2.***

• DOM Level 2 : recommandation de novembre 2000 qui s'appuie pleinement sur DOM Level 1.

Les changements concernent l'ajout d'un modèle objet pour les feuilles de style, d'un modèle événementielet la prise en charge des espaces de noms. Elle est découpée en cinq sous-recommandations :

Core, *Views*, *Style*, *Events* et *Traversal-Range* (voir figure 6-3). Une sixième partie est en coursd'acceptation : HTML (statut *Candidate Recommandation* en juin 2002).

• DOM Level 3 : recommandation en cours d'élaboration (statut *Draft*) qui s'appuie pleinement surDOM Level 2. Les changements concernent l'ajout des modèles de contenu (DTD et Schémas XML),
des fonctionnalités de validation, de chargement et de sauvegarde des documents.

Il appartient aux éditeurs d'implémenter ou non ces spécifications DOM, en particulier dans lesnavigateurs Internet :

• Mozilla 1.0 implémente DOM1, DOM2 (des erreurs sont en cours de correction) et une petitepartie de DOM3.

• Microsoft Internet Explorer 6 implémente DOM1.

Implémentations

L'objectif premier de cette recommandation est de permettre l'écriture de programmes génériques qui fonctionnentavec les logiciels de différents éditeurs. En ce qui concerne les navigateurs, les dernières versions implémentent correctement DOM1, mais il faut compter avec les anciennes versions qui sont toujours utilisées sur Internet et quine respectent pas pleinement les standards (Netscape 4.x, IE 5+, etc.).

Dans la suite de ce chapitre, nous nous intéresserons uniquement au noyau de DOM utilisé pour lesdocuments XML.

3.10.1.Le noyau DOM2 XML

DOM permet de structurer les documents XML sous forme d'une arborescence de noeuds. Il fournitles méthodes qui permettent de naviguer au sein de cet arbre et les interfaces qui permettent demanipuler les noeuds en fonction de leur type : élément, attribut, texte, etc.

La spécification du noyau DOM définit :

• les interfaces et les objets permettant de représenter et manipuler un document ;

• la sémantique de ces interfaces et objets ;

• les relations et collaborations entre ces interfaces et ces objets.

L'interface DocumentFragment

DocumentFragment est un objet Document « allégé » utilisé pour représenter des portions d'arborescencesavec une implémentation légère. Cet objet est de type Node.Cet objet n'a ni attribut, ni méthode.

3.11.Les analyseurs syntaxiques XML

Les analyseurs syntaxiques (*parsers*) XML sont des programmes capables de parcourir et d'évaluerdes documents XML. L'objectif est bien évidemment de fournir au développeur un moyen de manipulerles documents XML par le biais d'une interface de programmation (API). Deux types d'API
sont fournis par les analyseurs syntaxiques XML :

• Une API conforme à la spécification DOM (1 ou 2, et à l'avenir 3) du W3C : cette API offre lesfonctions de navigation et de manipulation de la structure arborescente du document XML.

• Une API événementielle appelée SAX pour Simple API for XML (*http://www.saxproject.org*) : cette APIa été créée au départ pour Java mais elle est maintenant disponible dans la plupart des langages.
C'est un standard de facto.

Les différences fondamentales entre ces deux API sont :

• la première s'appuie sur un structure arborescente en mémoire et permet une vision globale dudocument, alors que la seconde se contente de déclencher des événements (« début document »,
« début élément paragraphe », « contenu », « fin élément paragraphe », etc.) au fur et à mesure duchargement. La consommation mémoire de SAX est donc bien inférieure

• la première nécessite un chargement préalable et total du document pour permettre sa manipulationalors que la seconde permet de débuter le traitement immédiatement.

Les analyseurs syntaxiques XML se distinguent aussi en ce qui concerne la fonction de validation desdocuments XML : l'implémentation de XML Schema 1.0 est très difficile, au point que beaucoupd'analyseurs syntaxiques ne sont compatibles qu'à 99 % avec la recommandation du W3C

3.**12.Conclusion**

Dans ce chapitre nous avons présenté le language fondamentale pour assurer l'interropérabilité et cela parceque les composants des services web (SOAP , WSDL..) sont écrits en XML. Pour mener à bien notre travail ,une présentation des domaines d'études est prémordiale pour le prochain chapitre sur les généralités de e.learning

Chapitre IV

Généralités sur e.learning

4.1. Introduction

Les enseignants et les formateurs ont intégré l'outil informatique à leur pratique ; une méthode de formation qui permet théoriquement de s'affranchir de la présence physique d'un enseignant à proximité d'où EAD, TICE, EAO, FAD, e-Learning, e-formation, téléformation….. Qui désigne un nouveau phénomène dans le domaine de l'enseignement et de la formation.

Dans ce chapitre nous allons présenter l'E-LEARNING d'une manière générale. Premièrement, nous nous intéresserons à l'historique de l'E-LEARNING et son évolution au cours de ces dernières années, puis, nous mettrons la lumière sur ses aspects à travers un panorama des principales plates formes développées, et nous mettrons finalement en évidence les points forts et les faiblesses de l'E-LEARNING

4.2.E-Learning

4.2.1.Définition

Terme anglophone pour e-formation. Le « e » : abréviation de « électronique » et maintenant « en ligne ».Préfixe indiquant l'utilisation des NTIC avec le terme qui le suit, avec le développement de l'Internet et des réseaux, l'aspect électronique correspond surtout à celui « en ligne ».

Le e-Learning est un processus d'apprentissage à distance s'appuyant sur des ressources multimédias, il consiste a utiliser les ressources de l'informatiques et de l'Internet pour acquérir, à distance des connaissances .Les supports multimédias utilisés peuvent combiner du texte, du graphisme en deux ou trois dimensions, du son, de l'image, de l'animation et même de la vidéo.

4.2.2. Historique et évolution d'e-Learning

La formation à distance n'est pas un phénomène récent puisqu'elle existe depuis plusieurs années, au début, l'enseignement a distance consistait en la formation par correspondance pour les adultes n'ayant pas pu achever leur enseignement secondaire où supérieur.

Au fil des années, cette méthode a connu une évolution marquée, depuis le papier acheminé par poste ou par fax, ensuite la diffusion hertzienne via la radio et les émissions spécialisées des chaines de télévision arrivant ainsi à l'enseignement assisté par ordinateur (EAO)

4.2.2.1. Enseignement assisté par ordinateur

Les premiers systèmes d'enseignement assisté par ordinateur sont apparus dans les années 1970 ayant comme objectif l'apprentissage en tant que transfert de connaissances, l'EAO est un terme qui désigne l'application des techniques informatiques dans le domaine de l'éducation, ces systèmes utilisent l'ordinateur comme support de transfert de connaissances, l'ordinateur alors est un outil pédagogique qui permet l'apprentissage d'un domaine particulier pour les apprenants.

Une multitude de programmes éducatifs où didacticiels (logiciels destiné à l'enseignement assisté par ordinateur) furent développés, spécialisé dans une ou plusieurs matières comme :

- **Le didacticiel de test** : est un logiciel a but purement évaluatif ou diagnostic. L'apprenant doit répondre à un questionnaire séquentiel, ouvert ou à choix multiple, dans un domaine particulier.
- **Le didacticiel informatif** : est un logiciel sous forme d'une suite de pages-écrans, qui sert à proposer et à présenter des connaissances dans un domaine précis sans chercher à vérifier leurs acquisitions par l'apprenant.
- **Le didacticiel d'entraînement ou drill** :est un logiciel utiliser pour entrainer de automatismes, c'est-à-dire une série de tâches à accomplir séquentiellement pour résoudre un problème, ces tâches sont répétitives et on entraine les calculs et des formules pour les sciences, la conjugaison,……

- **Le didacticiel de simulation** : Est un logiciel basé sur le principe pédagogique qui consiste à familiariser l'apprenant avec un modèle, il propose des activités de résolution de problèmes aux apprenants qui leur permettent de découvrir ou d'approfondir les notions présentées. A travers des questionnaires appropriés ce type de didacticiel focalise l'attention de l'apprenant sue certain phénomènes. Des tests intermédiaires et finaux assurent que l'apprenant à bien acquis les compétences nécessaires.

- **Le tutoriel :**

  Utilise toutes les formes précédentes, c'est la forme la plus ambitieuse d'EAO, tant dans son but que dans les moyens mis en œuvre.

- **Le pédagogiciel** : est un didacticiel 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.

4.2.2.2. Insuffisances des systèmes d'EAO

Les systèmes d'EAO traditionnels restent très insuffisants à cause de leur non prise en compte des capacités et de la progression individuelle des apprenants, ainsi que de leur faible capacité d'expliquer les erreurs commises par ces derniers.

Afin d'améliorer la qualité et les possibilités d'adaptation, l'EAO a fait appel à des domaines tels que l'apprentissage, la pédagogie, la psychopédagogie, la psychologie cognitive et l'intelligence artificielle (IA). Un nouveau type de systèmes connus sous le sigle d'EIAO (enseignement intelligemment assisté par ordinateur) est alors apparu.

4.2.2.3.Enseignement intelligemment assisté par ordinateur(EIAO)

L'application des techniques de raisonnement offertes par l'Intelligence Artificielle (IA) et les Systèmes Experts (SE), dans le système éducatif, ont permis des innovations en introduisant un niveau d'interaction plus élevé entre l'apprenant et le système. C'est ce qui a donné naissance aux systèmes d'Enseignement Intelligemment Assisté par Ordinateur (EIAO), il s'agit de faire bénéficier l'EAO des nouvelles techniques informatiques, les recherches effectuées afin d'adapter l'apprentissage au niveau de l'apprenant et par rapport à son niveau de connaissance ont donné lieu à une nouvelle génération de systèmes appelés : les tutoriels intelligents (STI)

Les STI sont des systèmes d'enseignement informatiques qui possèdent un contenu qui assure 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 s'adapter à l'élève en situation d'apprentissage, le modèle général des STI repose sur une triple expertise : celle du domaine à enseigner (expert du domaine), celle de l'enseignement (module pédagogique) et celle des compétences et connaissances de l'élève (modèle de l'élève), comme le montre la figure suivante :

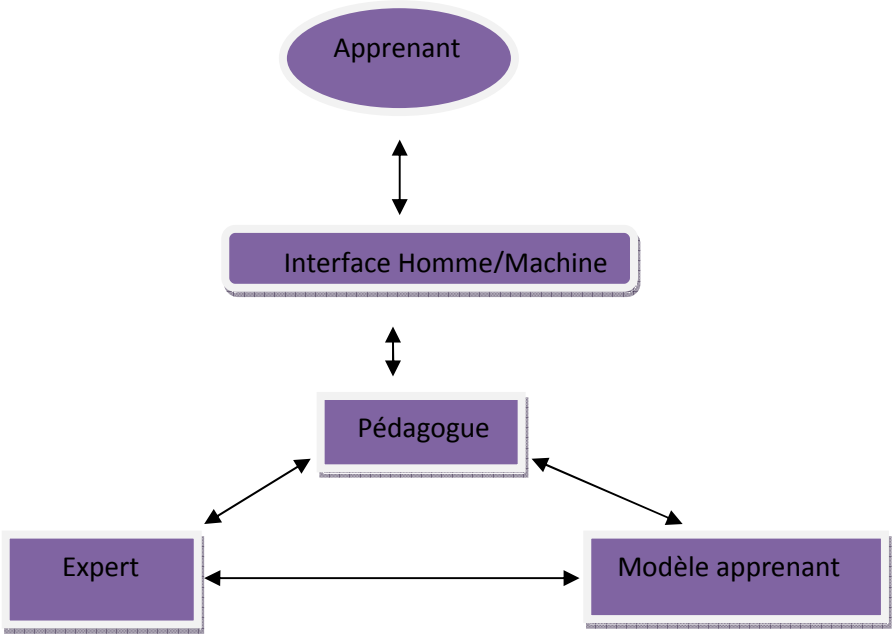


Figure 4.1. Les composantes d'un système d'EIAO.

- **L'interface apprenant :**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.
- **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.
- **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.

- **Le modèle apprenant :**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 apprenant 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.

4.2.2.4.Environnement Interactif d'Apprentissage avec Ordinateur

C'est aux environs de 1990 que le sigle EIAO prenait un autre sens : environnement interactif d'apprentissage avec ordinateur, en passe alors d'une vision de l'apprentissage qui était centrée sur le transfert des connaissances à une vision selon laquelle l'apprenant construit lui-même son apprentissage en interagissant avec un environnement, il sera confronter à un environnement dont le niveau de connaissances est supérieur au sein, provocant ainsi un écart qu'il va essayer de combler en acquérant et en remettant en cause ses connaissances.

Ces systèmes sont susceptibles d'évoluer, de se modifier en fonctions des résultats et des échecs de l'apprenant. L'apprenant doit agir et interagir pour adapter ses connaissances.

4.2.2.5. De l'EIAO à l'EIAD

C'est dernières années ont vu l'apparition de l'EIAD, qui rajoute la dimension « à distance » à l'EIAO. Cette dimension supplémentaire va faire apparaître le besoin de prise en compte de l'individu à distance et du renforcement du lien social, qui va faire intervenir les outils de communication synchrones et asynchrones entre individus, que ce soit pour les relations enseignant-élève que pour les relations entre élèves. La communication et l'interaction entre des machines et des humains distribués géographiquement, donnent naissance aux Environnement Informatiques pour l'Apprentissage Humain (EIAH).Un EIAH est un environnement informatiques qui a pour objectif de favoriser ou susciter des apprentissages, de les accompagner et de les valider, il intègre des agents humains (élève ou enseignant) et artificiels (informatiques) et leur offre des conditions d'interaction, localement ou à travers

les réseaux informatiques, ainsi que des conditions d'accès à des ressources formatives (humaines et/ou médiatisées) locales ou distribuées.

Les EIAD peuvent être définis comme étant des environnements cherchant à créer des conditions de construction de connaissances chez l'apprenant à partir d'interactions avec un système informatique et avec d'autres acteurs (apprenants, enseignants).

4.2.3. Formation à distance(FAD)

La formation à distance est une Situation éducative dans laquelle la transmission des connaissances et les activités d'apprentissage se situent en dehors de la relation directe en face à face entre l'enseignant et l'apprenant. Autrement dit, c'est un système de formation qui permet de se former sans se déplacer sur le lieu de la formation et sans la présence physique d'un formateur.

4.2.4. Formation ouverte

La formation ouverte ou Open Learning correspond à un mode d'organisation pédagogique varié qui s'appuie sur des apprentissages à distance, c'est un processus où l'apprenant à le libre choix de déterminer son itinéraire d'apprentissage (rythme, contenu, temps de travail) et même être en relation avec un enseignent ou un groupe structuré.

4.2.5. Formation ouverte et à distance(FOAD)

La formation ouverte et à distance peut être considéré comme l'évolution de la formation à distance, l'apparition du qualitatif « ouvert » introduit des notions supplémentaires, il caractérise des situations qui offrent une plus grande flexibilité dans le mode d'organisation pédagogique

La FOAD a été définie par le collectif de Chasseneuil (2000) comme « un dispositif organisé, finalisé, reconnu comme tel par les acteurs, qui prend en compte la singularité des personnes dans leurs dimensions individuelles et collectives et qui repose sur des situations d'apprentissage complémentaires et plurielles en termes de temps, de lieux, de médiations pédagogiques humaines et technologiques et de ressources »

4.2.6.Plateforme d'E-Learning

Une plate-forme de la formation à distance est un logiciel qui a pour but la diffusion a distance des contenus de formation , appelée aussi LMS (Learning Management System).Elle est basée sur des techniques de travail collaboratif et regroupe les outils nécessaires aux principaux utilisateurs de l'application : apprenant, formateur, administrateur. Elle fournit a 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, annotations, etc.

4.2.6.1. Fonctionnalités des plates-formes de E-Learning

Les plates-formes de FAD servent à permettre aux apprenants d'accéder à des offres de e-Learning sur Internet, elles proposent de nombreuses fonctions comme :

- La gestion du contenu par la création et le stockage de ressources pédagogiques (publier des cours, déposer des documents)
- La gestion de la formation (inscription des apprenants, gestion des acteurs, l'accès aux cours (connexion, identification et téléchargement)
- La gestion de l'interactivité, l'accompagnement de l'apprenant d'une manière asynchrone (messagerie, forum,…) et synchrone (vidéoconférences)
- La gestion des compétences, le développement d'outils de suivi des progrès des apprenants et la proposition des technologies permettant d'adapter la contenu de façon personnalisée aux aptitudes où difficultés détectés lors du parcours

4.2.6.2. Les acteurs d'une plateforme d'E-Learning

Les différents acteurs d'une formation à distance peuvent se répartir suivant les rôles qu'ils seront amenés à jouer, les trois catégories principales sont : les apprenants, les formateurs (enseignants) et les administrateurs

a) Enseignant

L'enseignant joue un rôle moteur dans la formation,on peut distinguer les activités suivantes d'un enseignant :

- Crée des parcours pédagogiques types et d'individualisés de sonenseignement ;

- Incorpore des ressources pédagogiques multimédias ainsi que des exercices et des évaluations automatiques (QCM par exemple).
- Aider les apprenants à progresser, il doit être en mesure de répondre aux questions qui lui seront posées ;
- La mise en pratique immédiates des connaissances acquises par l'apprenant ;

b) Apprenant

L'apprenant est l'acteur central pour lequel la formation est conçue ; pour cela il doit faire preuve de grande autonomie et de capacité d'autogestion au cours de son apprentissage donc il a besoin de :

- consulte en ligne ou télécharge les contenus pédagogiques qui lui sont recommandés ;
- Effectue des exercices, s'auto évalue et transmet des devoirs à corriger ;
- Possibilité de visualiser son niveau d'avancement dans le cours ; connaitre les parties du travail déjà effectuées et celles qui ne le sont pas encor.

c) Administrateur

L'administrateur est l'acteur qui contrôle la plateforme et veille sur le bon fonctionnement de celle-ci et n'a aucune influence sur le coté pédagogique. Ses principales tâches sont les suivantes :

- Installer les applications de la plate forme ;
- Assurer la maintenance et la gestion des accès au système ;
- gérer le droits et les rôles des différents utilisateurs ;
- la gestion des inscriptions administratives des apprenants et des formateurs ;
- agit à titre de responsable technique de la plateforme.

4.2.6.3. Exemples de plates formes e-Learning

- **Moodle**

Moodle(Object-OrientedDynamic Learning Environment)est une plate forme « open source » mise à disposition des utilisateurs librement suivant la licenceGPL (General Purpose License). Cela signifie que Moodle bénéficie d'un *copyright*, mais que l'utilisateur (administrateur, enseignant, …) dispose d'un certain nombre de libertés. Il a le droit de copier, d'utiliser et de modifier Moodle pour autant qu'il s'engage à mettre à disposition des autres le code source, à ne pas modifier ni supprimer la licence originale et les copyrights et à appliquer la même

licence à tous les travaux dérivés. D'un point de vue technique, Moodle est basée sur du code PHP et fait appel à une base de données MySQL. La figure 1.2 représente la page principale de Moodle. Les fonctionnalités principales de la plate formeMoodle sont classées en fonction des acteurs et sont résumées dans le tableau 1.1.

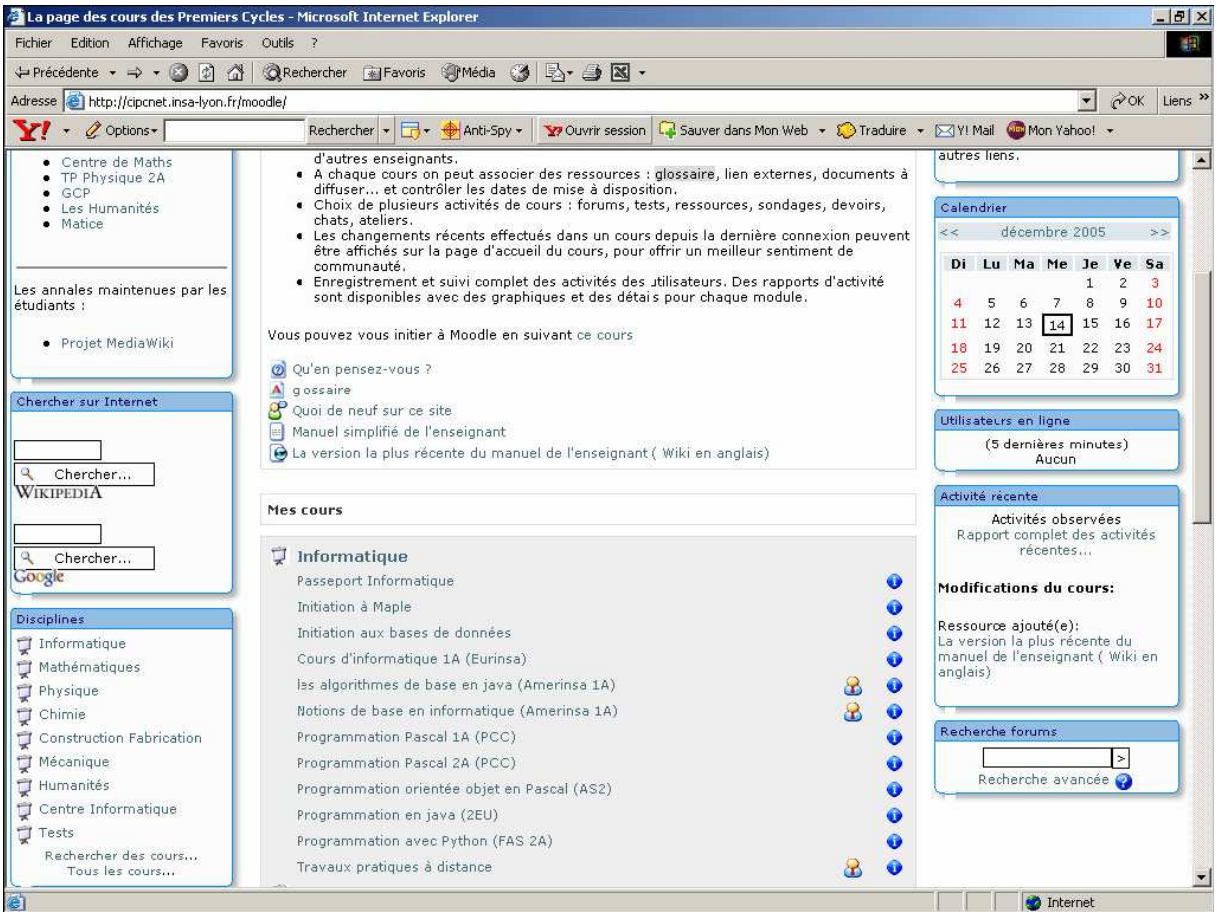


Figure 4.2. Page principale de Moodle

Espace administrateur	Espace enseignant	Espace apprenant
-Inscription et suivi des apprenants -Gestion des dossiers des apprenants -Création de groupes d'utilisateurs avec restrictions paramétrables	-Travail en groupes -Tutorat (communication enseignant-apprenant synchrone et asynchrone) -Evaluation des apprenants	-Accès aux supports de cours -Espace privé -Communication asynchrone apprenant-apprenant (forums) -Communication synchrone apprenant-apprenant par messagerie instantanée

Tableau 4.3. Fonctionnalités de Moodle

- **Cas algérien**

En Algérie, le e-Learning en est à ses débuts. Le fournisseur d'accès Internet Djaweb[1], filiale d'Algérie Télécom a lancé en mars 2006 le service « e-Learning » par carte prépayée, réalisé en partenariat avec Thompson et Microsoft. Premier du genre en Afrique, ce service propose, via Internet, un contenu de 4.000 cursus de formation dans le domaine des technologies de l'information et de la communication (TIC) et du développement des compétences professionnelles. Il s'agit, entre autres, de l'initiation à l'informatique aux certifications les plus connues des grands éditeurs informatiques (Microsoft, Oracle, Cisco, IBM, Novell Comptia, SAP...).

«*DZCampus.com*» est la première plate-forme informatique e-Learning en Algérie, Cette initiative a été lancée en Mars 2007 par deux entreprises algériennes spécialisées dans la formation et la communication multimédia, à savoir, Comform et Actech[2]. Les étudiants, les

[1]Djaweb : http://www.djaweb.dz
[2]Actech **est une start-up Internet, créée en 2005, spécialisée dans la communication multimédia.**

entreprises, les particuliers et autres organismes de formation sont les premiers concernés par cette solution de e-Learning bilingue.

2.7. Formes de l'e-Learning

Il existe essentiellement cinq formes d'enseignement à distance :

- **La forme « tout à distance sans intervention tutoriel »** : Cette forme d'apprentissage est entièrement autonome, sur un CD Rom, ou des fichiers pdf, doc……. Elle pourrait être appliquée pour de courtes sessions avec des évaluations continues fréquentes en forme de quiz, exercice de simulation. Elle nécessite des consignes et des explications claires et un contenu segmenté de la façon la plus petite possible.

- **La forme « tout à distance avec tutorat asynchrone »** : Cette seconde méthode reprend la forme précédente mais en y incluant un tutorat qui permet de compenser le manque de présentiel. Le tuteur répond à ces questions en différé. Il n'est pas obligatoire au formateur d'être connecter à tout moment, et il communique avec l'apprenant par chat, e-mail ou forum d'échange.

- **La forme « tout à distance avec tutorat synchrone » :** ce qui diffère avec la forme précédente c'est que le tuteur fixe une plage horaire pour rassembler un groupe d'apprenant avec ou sans vision directe de l'intervenant. Le tuteur dialogueen temps réel avec les apprenants.

- **L'accompagnement en ligne :(classe virtuelle en temps réel)**
  Cette forme d'enseignement permet de définir un parcours personnalisé, le tuteur fixe une plage horaire pour réunir les apprenants autour d'un forum en ligne, l'interactivité entre apprenant et formateur est en temps réel. Le principe est celui d'un partage d'application (logiciel spécifique).

- **La Visio_ formation personnalisée :** l'efficacité de cette forme de e-Learning repose sur le parcours personnalisé défini par un tuteur pour un apprenant et qui se communique en direct et en image.

Pour mettre en œuvre et en pratique les sessions d'enseignement à distance, deux méthodes peuvent être adoptées : asynchrone et synchrone.

4.2.7.1. Enseignement asynchrone

L'enseignement asynchrone est une méthode d'apprentissage s'adaptant aux disponibilités de l'apprenant. Il s'agit d'un apprentissage délivré à la demande. Celui-ci a accès à un ou des instruments (exemples : vidéo, enregistrement audio, texte, logiciel d'apprentissage virtuel) qu'il utilisera à sa guise. Le suivi de formation avec le formateur ou entre les membres d'un groupe d'apprentissage se fera par voie indirecte (courrier électronique, forum de discussion).

a) Les avantages

- 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.

b) Les inconvénients

- Le temps qui peut s'écouler entre une question de l'apprenant et la réponse du formateur.
- L'apprenant est seul devant son écran. En cas de difficulté, il ne peut s'appuyer sur aucune aide extérieure (ça renforce le sentiment d'isolement).

4.2.7.2. Enseignement synchrone

L'enseignement synchrone, contrairement à l'enseignement asynchrone, se caractérise par l'interaction directe et en temps réel entre les apprenants et les formateurs. C'est la méthode la plus traditionnelle, celle qui s'approche le plus de la classe magistrale. Nous parlons de l'apprentissage synchrone lorsque tous les apprenants d'un groupe sont simultanément en ligne avec leur formateur et échangent des documents, partagent des applications, visionnent les mêmes écrans ou encore reçoivent des images de visioconférence (Webcast).

a) Les avantages

- Les apprenants interagissent intensivement à l'écran avec les formateurs. Le langage oral ou visuel est utilisé.

- Le modèle de la classe est familier.
- La possibilité de créer rapidement du contenu prêt à diffuser.
- La dynamique de groupe s'installe plus rapidement.

b) les inconvénients

- il faut posséder le matériel technique nécessaire.
- Elle ne permet pas à chaque apprenant d'évoluer à son propre rythme et de choisir l'ordre des éléments de cours à sa guise.
- Réduit le taux de contribution des acteurs faute de temps limité ; ce qui ne vas pas permettre aux apprenants d'approfondir certains aspects du cours.

4.2.8.Avantages et inconvénients du e-Learning

Economies, gain de temps, formation de masse, le e-Learning séduit chaque jour de nouvelles entreprises pour leurs formations professionnellesgrâce à des arguments imparables, le marché du e-Learning est considérable, de nouveaux métiers, de nouveaux acteurs émergent ce qui prédit un bel avenir à la formation à distance

Mais si celle là détient de nombreux atouts, le système possède aussi ses limites. Le e-Learning se heurte notamment à la réticence d'entreprises et de certains salariés face à une méthode d'apprentissage novatrice qui leur est peu familière.

a) Les avantages

- La formation est ouverte à toute personne, quels qu'en soient son âge, son niveau d'instruction, sa catégorie socioprofessionnelle, etc.
- Disponibilité de l'information (accès aux savoirs et aux savoirs faire sans contraintes de distance).
- L'apprentissage est plus rapide et plus durable. En effet, des études montrent que les courbes d'apprentissage peuvent être améliorées par l'e-Learning et que la mémorisation des informations est souvent meilleure. Les temps d'apprentissage sont en général plus faibles qu'en présentiel (environ 50% plus faibles).Bien utilisé, l'e-Learning permet donc d'améliorer la qualité de la formation.

- L'e-Learning permet l'accès à de nouvelles compétences qui sont plus que jamais indispensables aux exigences de la vie moderne. Chacun peut se familiariser avec les nouvelles technologies comme l'ordinateur, les systèmes multimédias et l'Internet.
- L'apprentissage est personnalisé car l'apprenant peut choisir le temps à passer sur chaque module d'une formation en fonction de ses acquis. Il peut adapter le rythme du cours à son niveau (moins de stress, moins de frustration).
- L'e-Learning coûte globalement moins cher que le présentiel du fait qu'il élimine les coûts liés à la rémunération des professeurs, la gestion des salles de cours et les déplacements.
- L'apprenant est le centre de concentration et non plus le formateur. Donc, l'apprenant est incité à être un émetteur et de participer d'une manière significative à la formation et ne plus se limiter à être un récepteur d'informations et de savoirs comme c'est le cas des étudiants à la traditionnelle.
- Créer un sentiment de liberté et de confiance de l'apprenant en lui-même. Les sentiments d'intimité et de honte des collègues et du formateur en cas de faute par exemple se dissipent en e-Learning. L'apprenant étudie tout seul, face à son ordinateur et n'est pas observé des autres (excepté le cas de visioconférence).

b) Les inconvénients

- L'absence physique de l'enseignant avec tout son poids d'émotions, d'autorité et d'expressivités humaines.
- Les problèmes techniques afférents au fonctionnement des systèmes de formation (perturbation du réseau de communications, pannes des ordinateurs, terminaux ou serveurs, attaques des documents électroniques de cours par des virus ou des pirates, etc.)
- L'accès à l'outil informatique est nécessaire. Ainsi que le cout élevé de la mise en place de l'infrastructure technique ne font que freiner la diffusion de e-Learning au prés des utilisateurs.
- L'e-Learning limite les interactions entre les individus. Certains mécanismes de communication ne peuvent pas être reproduits (langage du corps par exemple), alors qu'ils jouent un rôle important dans la diffusion du savoir.
- Les autres membres du groupe ne sont pas là pour poser les questions auxquelles l'apprenant n'aurait pas pensé et faciliter ainsi la compréhension.

- Adaptation difficile des apprenants sur les pratiques de e-Learning conduit à un taux d'abondon élevé.

4.3. Conclusion

Dans ce chapitre, nous avons étudié l'évolution de la formation à distance depuis son apparition sous sa forme la plus élémentaire « les cours par correspondance » jusqu'à sa forme actuelle l'e-Learning ou les technologies les plus évoluées sont appliquées en passant par les environnements d'enseignement basés sur les techniques de l'intelligence artificielle. Enfin nous avons présenté quelques avantages et inconvénients du e-Learning.

Dans le chapitre suivant nous évoquerons l'analyse et la conception

Chapitre V

Analyse et conception

5. Introduction

Dans les Trois chapitres précédents, nous avons présenté l'adaptabilité, l'interopérabilité et les moyens utilisés pour les mettre en œuvre. Ensuite, au niveau du quatrième chapitre, nous avons défini le domaine de notre application (e-Learning et enseignement synchrone) et les technologies qui participent à sa mise en œuvre. Ce chapitre sera consacré à l'analyse et la conception de notre application adaptable. Pour cela nous optons pour les étapes suivantes :

- ✓ Utilisation de diagrammes de cas d'utilisation pour l'aspect fonctionnel
- ✓ Réalisation du diagramme de séquence pour l'aspect dynamique
- ✓ Réalisation de diagramme de classe pour l'aspect statique
- ✓ Application du principe de protection des variations pour assurer l'adaptabilité

Pour modéliser le système, on a opté pour l'UML qui est un langage unifié de modélisation, on se propose alors de donner une définition brève de l'UML.

5.1.Analyse

5.1.1.Introduction à l'UML

5.1.1.1.Définition

UML (Unified Modeling Language), que l'on peut traduire par « langage de modélisation unifié », est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, il est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

UMLn'est pas une méthode, dans la mesure où elle ne représente aucune démarche.A ce titre, UMLest un formalisme de modélisation objet. Le mot méthode parfois utilisé par abus de langage ne doit pas donc être entendu comme une démarche.

5.2.1.2. Extension d'UML pour le Web (Web application extension)

L'extension d'UML pour le Web définit un ensemble de stéréotypes, d'étiquettes et de contraintes, qui rend possible la modélisation d'application Web. Ces stéréotypes et ces contraintes sont appliqués sur certains des composants propres aux applications Web, permettant ainsi de les représenter au sein du même modèle et sur les mêmes diagrammes que ceux qui décrivent le reste du système.

Le principal élément spécifique des applications Web étant la page Web , plusieurs stéréotypes ,qui lui sont destinés , sont décrits plus loin ainsi que d'autres , conçu pour les éléments tels que les cadres , les cibles et les formulaires représentant eux aussi des composants architecturales significatifs.

Ainsi, nous avons optés pour cette méthode pour modéliser notre d'application. Donc nous commencerons par l'élaboration des diagrammes de cas d'utilisations pour la spécification des besoins de notre système, les diagrammes de séquence pour le modèle d'analyse ainsi que les diagrammes de classes pour la conception de l'application.

5.2.1.3. Modélisation avec L'UML

UML permet de représenter des modèles, mais il ne définit pas de processus d'élaboration de modèles. Les auteurs d'UML conseillent tout de même une démarche pour favoriser la réussite d'un projet, cette démarche doit être :

- **Une démarche itératif et incrémentale :** Pour comprendre et représenter un système complexe, pour analyser par étapes, pour favoriser le prototypage et pour réduire et maitriser l'inconnu.
- **Une démarche guidée par les besoins des utilisateurs :** Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoin. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.
- **Une démarche centrée sur l'architecture logicielle :** c'est la clé de voute de succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

5.2.2.Spécification des besoins

5.2.2.1. Objectif du travail

Notre travail consiste à la mise en place d'une plate-forme d'e-learning adaptable permettant la diffusion et l'intégration en temps réel d'une activité pédagogique de type télé-cours. Une telle application doit permettre à l'enseignant de présenter ou d'intégrer des cours en direct à un ensemble d'apprenants géographiquement éloignés connectés à la plate-forme à l'aide d'un réseau (intranet ou internet).

Ce travail s'inscrit dans le cadre de la conception et du développement d'environnementsInformatiques multimédias pour la diffusion et l'integration de contenus pédagogiques en ligne, notammentSynchrone. L'architecture d'un tel système repose essentiellement sur les aspects suivants :

- protocoles pour satisfaire les exigences de qualité du média utilisé (image, son, vidéo, etc.)nécessaires au bon suivi d'un télé-cours.
- L'intégration d'outils de communication et de fonctionnalités logicielles adéquates.

5.2.2.2. Identification des acteurs

a)- Définition d'un acteur

Un acteur représente un rôle joué par une entité externe (utilisateur, dispositif matériel ou autre système) qui interagitdirectement avec le système étudié, il peut consulter et/ou modifier directement l'état du système, en émettant et/ou recevant des messages susceptibled'être porteur des données.

b)- Les acteurs de notre système :

Notre système comprend quatre acteurs qui sont :

- **Visiteurs de la plate-forme:** Toute personne qui se connecte à la plate-forme sans qu'elle soit inscrite. Un visiteur pourra s'informer des différentes fonctionnalités offertes par la plate-forme. L'ensemble d'action qu'un visiteur peut effectuer sont :
  - ✓ Consulter catalogue ;
  - ✓ Faire une inscription en tant qu'apprenant;
  - ✓ Faire une demande de recrutement en tant que formateur ;
  - ✓ Contacter l'administrateur.
- **L'apprenant:** Toute personne se connectant à l'application pour suivre un cours en direct, l'apprenant nécessite une inscription pour lui donner les droits d'accès (login, mot de passe), l'application doit lui permettre :
  - ✓ S'identifier pour accéder à son espace ;
  - ✓ Récupérer son mot de passe en cas d'oublie
  - ✓ Consulter et télécharger les cours ;
  - ✓ Assister au cours en « **streaming »** ;
  - ✓ Répondre au QCM ;

- ✓ Participer au chat et forum ;
- ✓ Accès à la messagerie ;
- ✓ Voir et modifier son profil.

- **Le Formateur :**C'est la personne chargée de diffuser le cours en direct pour un ensemble d'apprenants.aprés avoir confirmer son inscription on lui donne les droits d'accès et elle devient membre de la plate-forme.

  L'application doit lui fournir les servicessuivant :

  - ✓ s'identifier pour accéder à son espace ;
  - ✓ récupérer son mot de passe en cas de perte ;
  - ✓ création et mise à jour du contenu pédagogique ;
  - ✓ organiser et animer des cours en streaming ;
  - ✓ proposer des QCM ;
  - ✓ suivre l'évaluation des apprenants ;
  - ✓ participer au forum et chat ;
  - ✓ accès à la messagerie ;
  - ✓ voir et modifier son profil.

- **L'administrateur** : il a pour rôle la gestion de la plate-forme, il s'occupe aussi de la confirmation des inscriptions des apprenants et des formateurs.

  L'application doit lui permettre de :

- ✓ S'identifier pour accéder à son espace ;
- ✓ Récupérer son mot de passe en cas de perte ;
- ✓ Gérer les demandes de recrutements des formateurs ;
- ✓ Gérer les utilisateurs de la plate-forme ;
- ✓ Valider les QCM proposer par les enseignants ;
- ✓ Gérer les plannings des groupes d'apprenants et les formations ;
- ✓ Gérer le forum ;
- ✓ Accès à sa messagerie.

5.2.2.3. Diagramme de contexte

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et les liens avec l'environnement extérieur. Il permet aussi de bien délimiter le champ d'étude. Pour notre cas le diagramme de contexte est donné par la figure suivante :

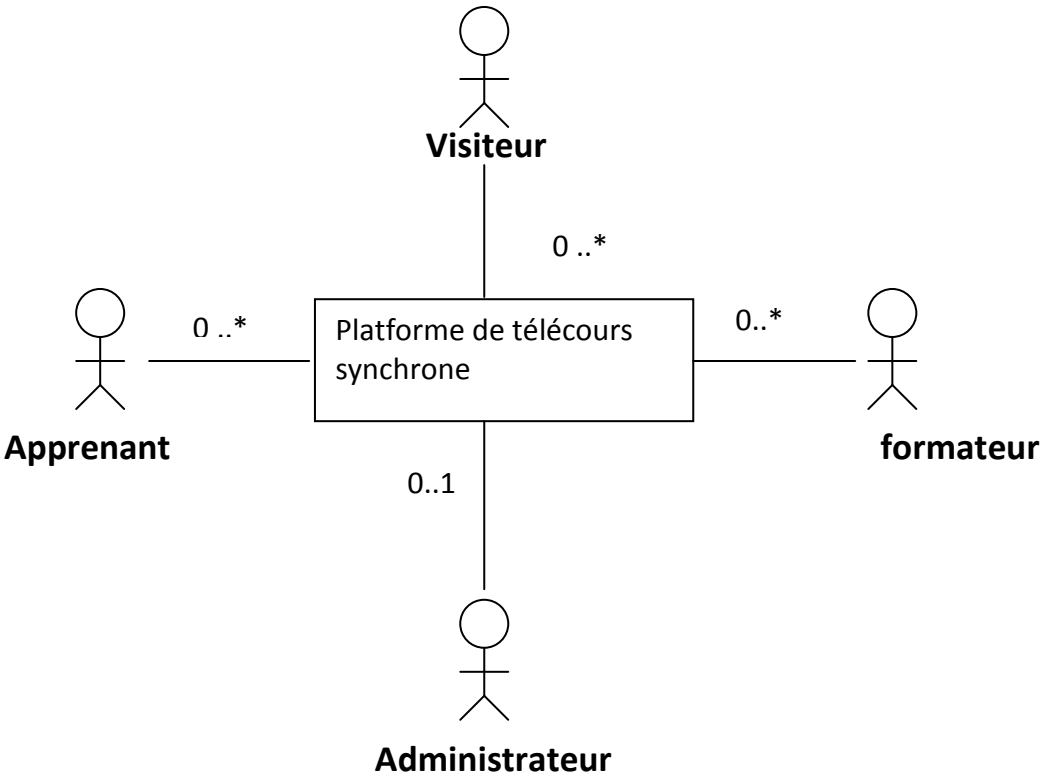


Figure 5.1diagramme de contexte de notre système

5.2.3. Spécification des tâches

Pour chaque acteur, nous spécifions les tâches qu'il assure. Le tableau suivant résume ces tâches :

Acteurs	Tâches
Visiteur	**T0 :** se connecter à la plate-forme (accéder à la page d'accueil) ; **T1 :** Consulter les informations sur les fonctionnalités de la plate-forme **T2 :** Contacter l'administrateur
Apprenant	**T3 :** S'authentifier T4 : Consulter la liste des cours T5 : Accéder à une séance synchrone T6 : utiliser les outils de communication
Formateur	**T7 :** S'authentifier ; T8 : Gérer les cours T9: Voir le planning des cours T10 : Présenter une séance de cours T11 : utiliser les outils de communication
Administrateur	T12 : Gérer les inscrits et les préinscrits T13 : Planifier les cours T14 : Administrer le forum T15 : Gérer la messagerie

5.2.4. Les cas d'utilisations

5.2.4.1. Définition d'un cas d'utilisation

Un cas d'utilisation (en anglais **use case**) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre, mais UML propose un formalisme et des concepts issus de bonnes pratiques.

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système.

L'objectif poursuivi par les cas d'utilisations est de permettre de décrire la finalité des interactions du système et de ces utilisateurs.

5.2.4.2. Relations entre cas d'utilisations

- **Relation « include » :**

  Une relation d'inclusion d'un cas d'utilisation **A** à un cas d'utilisation **B** signifie qu'une instance de **A** contient le comportement décrit dans **B,**le cas d'utilisation **A** ne peut être utilisé seul.

- **Relation « Extend » :**

  Une relation d'extension d'un cas d'utilisation **A** par rapport à un cas d'utilisation **B** signifie qu'une instance de **A** peut être étendue par le comportement décrit dans **B.**

- **Relation « use » :**

  Lorsqu'une ou plusieurs tâches sont utilisées régulièrement on peut les factoriser dans un même use case et faire de telle sorte que d'autres uses cases l'utilise en pointant par une flèche.

5.2.4.3. Les diagrammes de cas d'utilisations pour chaque acteur

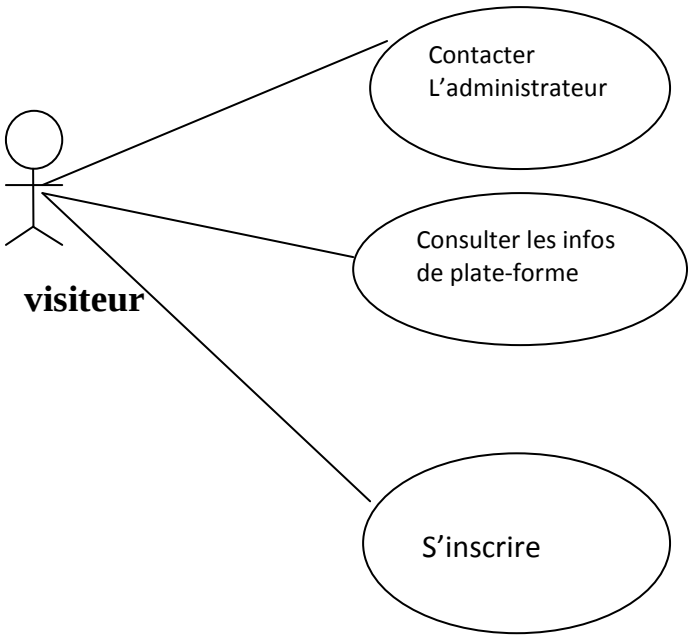


Figure 5.2: Diagramme de cas d'utilisation « acteur : visiteur»

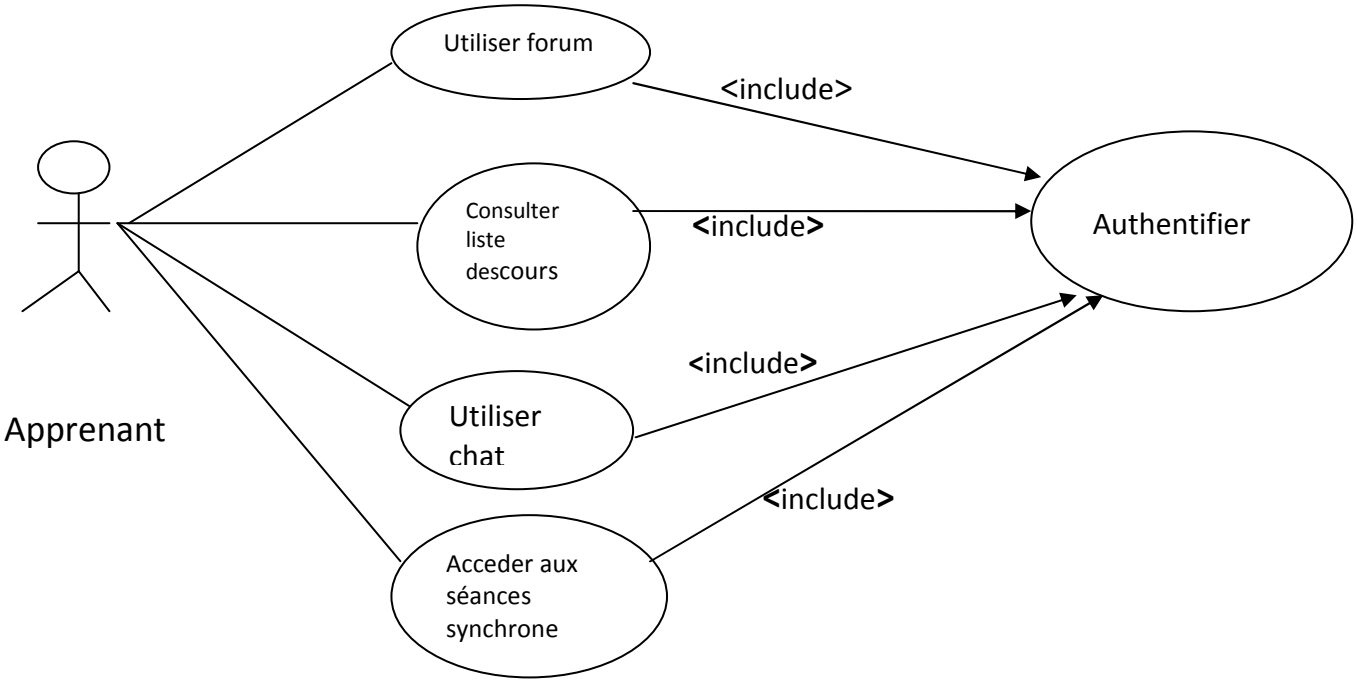


Figure 5.3: Diagramme de cas d'utilisation « acteur : Apprenant »

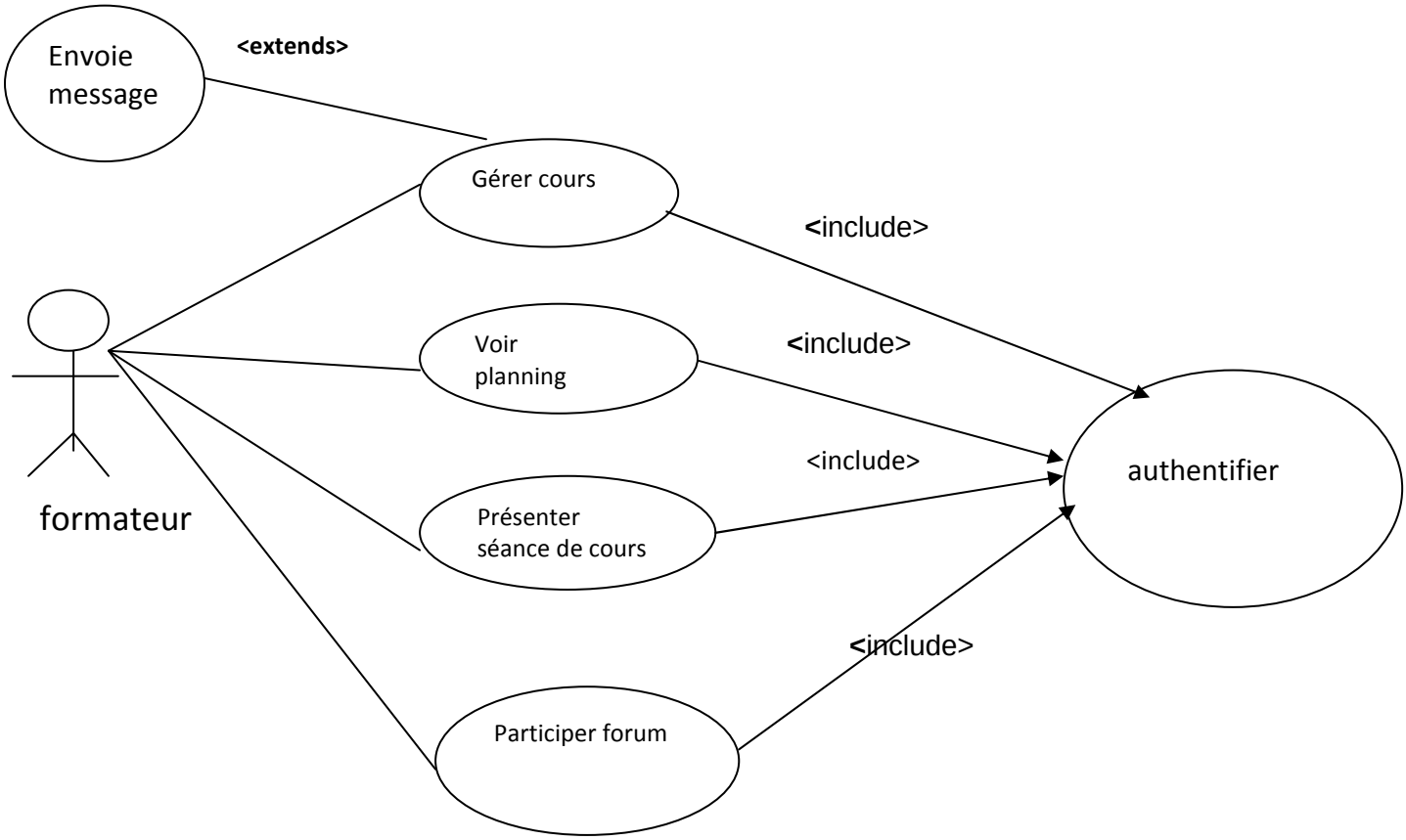


Figure 5.4: Diagramme de cas d'utilisation « acteur : Formateur»

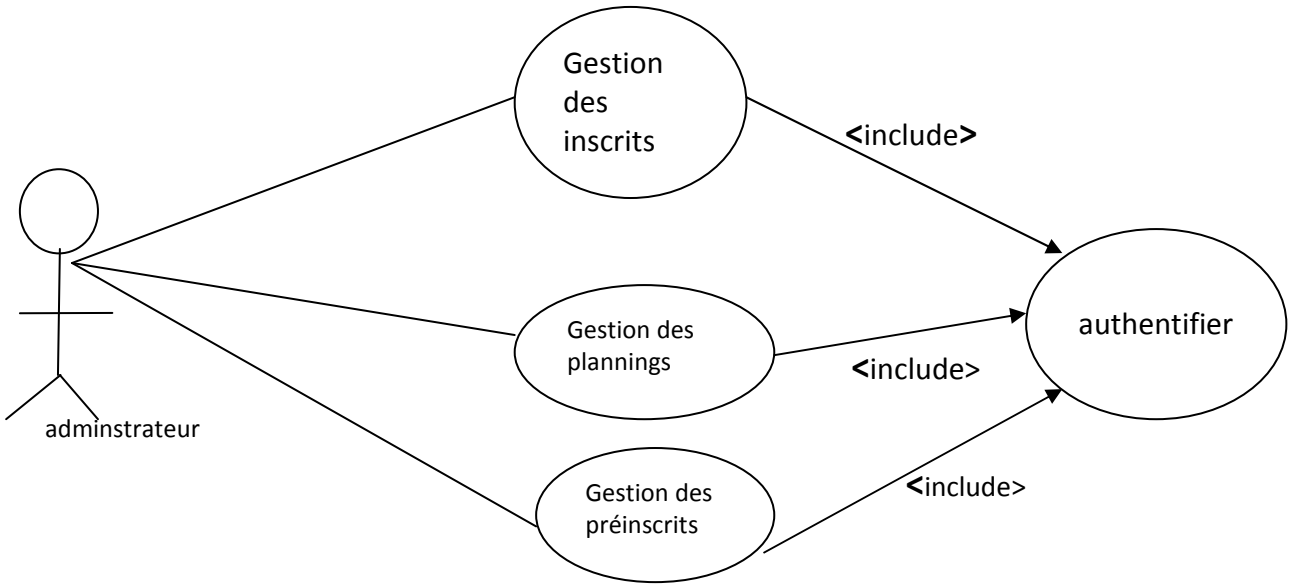


Figure 5.5: Diagramme de cas d'utilisation<< acteur : administrateur>>

5.3. Conception

5.3.1. Introduction

La conception de l'application Web se distingue de la conception d'autres systèmes par deux activités, qui sont la répartition des objets sur le client ou sur le serveur et la définition de l'interface utilisateur sous forme de page Web. Pour développer notre projet nous sommes appelés à concevoir l'application et la base de données avec laquelle interagie le système. Dans la conception de notre application,nous allons construire les diagrammes de séquence suivis des diagrammes d'activités et de classes. Nous commencerons par la description des diagrammes de séquence pour les cas d'utilisations suivants :

1. Ajouter une séance de cours.
2. Ajouter support de cours.
3. Accéder au cours synchrone.

5.3.2. Conception de l'application

5.3.2.1. Diagramme de séquence

Avec les diagrammes de séquences, l'UML fournit un moyen graphique pour représenter les interactions entre objets à travers le temps. Ces diagrammes montrent typiquement un utilisateur ou un acteur et les objets et composants avec lesquels ils interagissent au cours de l'exécution du cas d'utilisation. Un diagramme de séquence représente en général un seul 'scénario' de cas d'utilisation ou flux d'événements.

Les classes d'objets utilisées dans la représentation du diagramme de séquence peuvent être réparties dans les trois catégories suivantes :

- **Les objets d'interface :** ils représentent l'interface entre l'acteur et le système. Des exemples classiques en sont les écrans de saisie et de contrôles spéciaux d'interface utilisateur, ou dans l'application Web, des pages Web complètes.
  L'icone : 

- **Les objets entité :** ce sont des objets décrits dans le cas d'utilisation. Les commandes, les produits et les fiches de paie sont des objets entités dont les instances peuvent apparaître dans de nombreux cas d'utilisation.
  L'icone : 

- **Les objets contrôle :** ils représentent les processus, c'est –à-dire les activités système assez significatives pour être nommées, telles que le calcul da la paie, la facturation ou l'inventaire du stock. Les objets contrôle dirigent les activités des objets entités et d'interface.
  L'icone : 

Les Digrammes de séquence correspondant ànotre application sont :

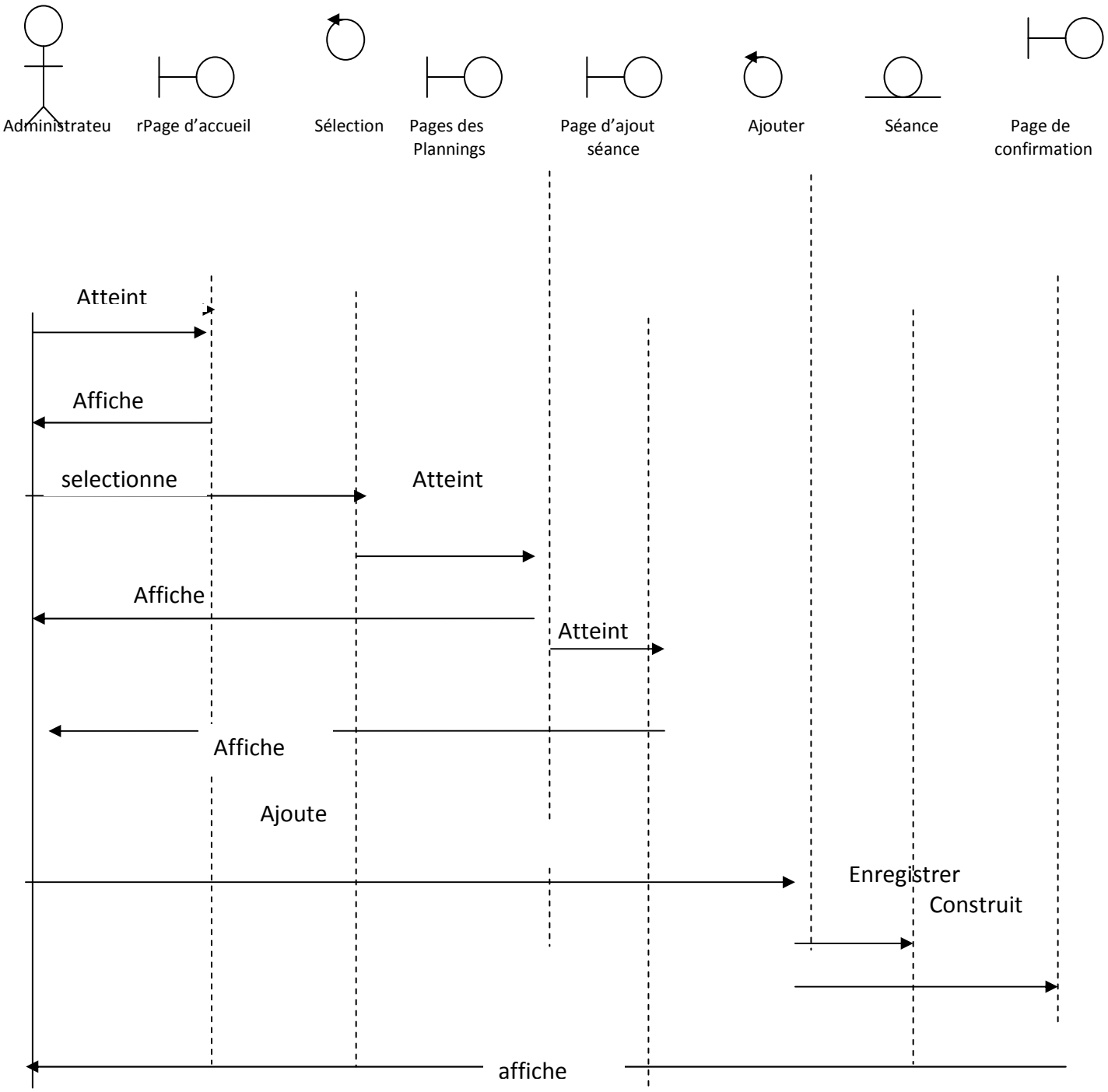


Figure 5.6: diagramme de séquence pour le cas d'utilisation «Ajouter une séance de cours ».

1. Après identification l'administrateur atteint la page d'accueil de son espace ;

2. Le système le lui affiche la page d'accueil avec le lien «Gestion des plannings»

3. l'administrateur sélectionne le lien « Gestion des plannings »;

4. le système lui affiche la page «planning » avec le lien «ajouter une séance » ;

5. l'utilisateur clique sur le lien « ajouter une séance » ;

6. le système affiche le formulaire d'ajout de séance;

7. l'administrateur remplit les champs requis et soumet le formulaire;

8. le système met à jour la BDD en ajoutant la nouvelle séance ;

9.le système affiche un message ou il confirme l'ajout de la séance.

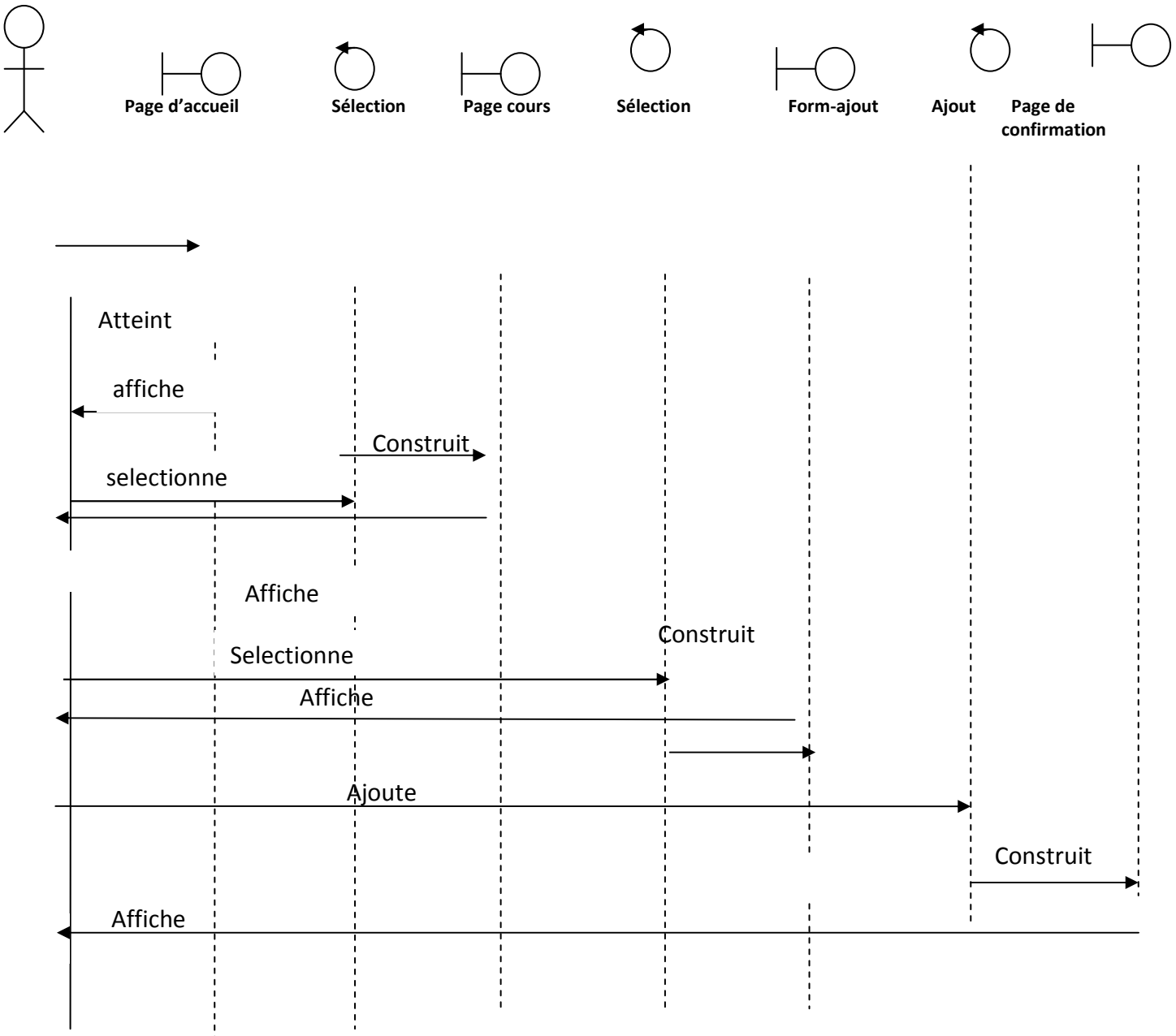


Figure 5.7 : Diagramme de séquence pour le cas d'utilisation<< Ajouter séance de cours >>.

1)- Après identification le formateur atteint sa page d'accueil;

2)- le système lui affiche sa page d'accueil avec le lien «cours » ;

3)- le formateur clique sur le lien « cours » ;

4)- le système affiche la page « cours » avec le lien «ajouter cours »;

5)- le formateur clique sur le lien«ajouter cours » ;

6)- le systèmeaffiche le formulaire d'ajout de cours avec le bouton ajouter;

7)- le formateur sélectionne le fichier à ajouter et clique sur le bouton ajouter;

9)- le système affiche enregistre le fichier et affiche un message de confirmation d'ajout.

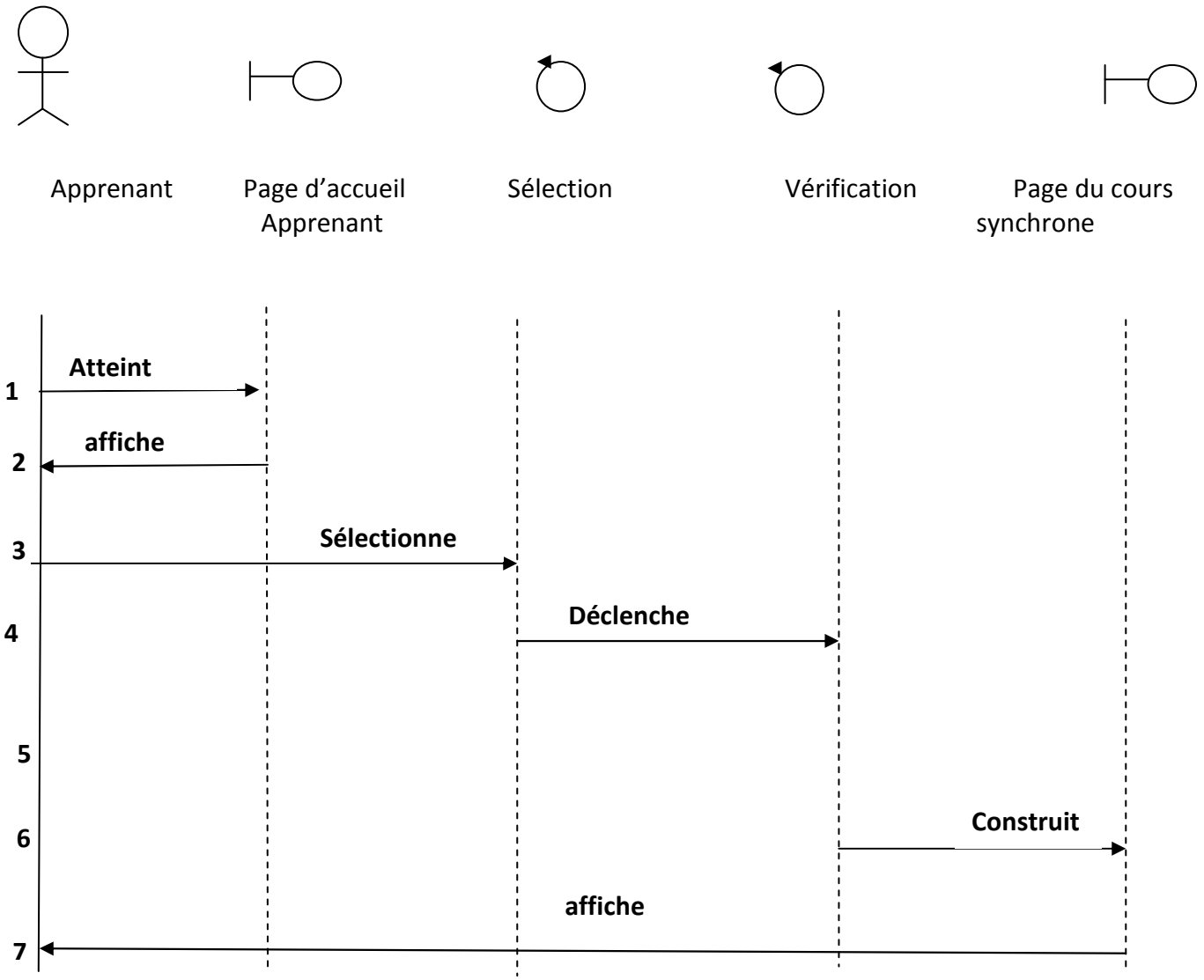


Figure 5.8:Diagramme de séquence pour le cas d'utilisation « Accéder au cours synchrone ».

1)- Après identification l'apprenant atteint sa page d'accueil ;

2)- le système lui affiche sa page d'accueil avec le lien **«** Accéder au cours synchrone ».

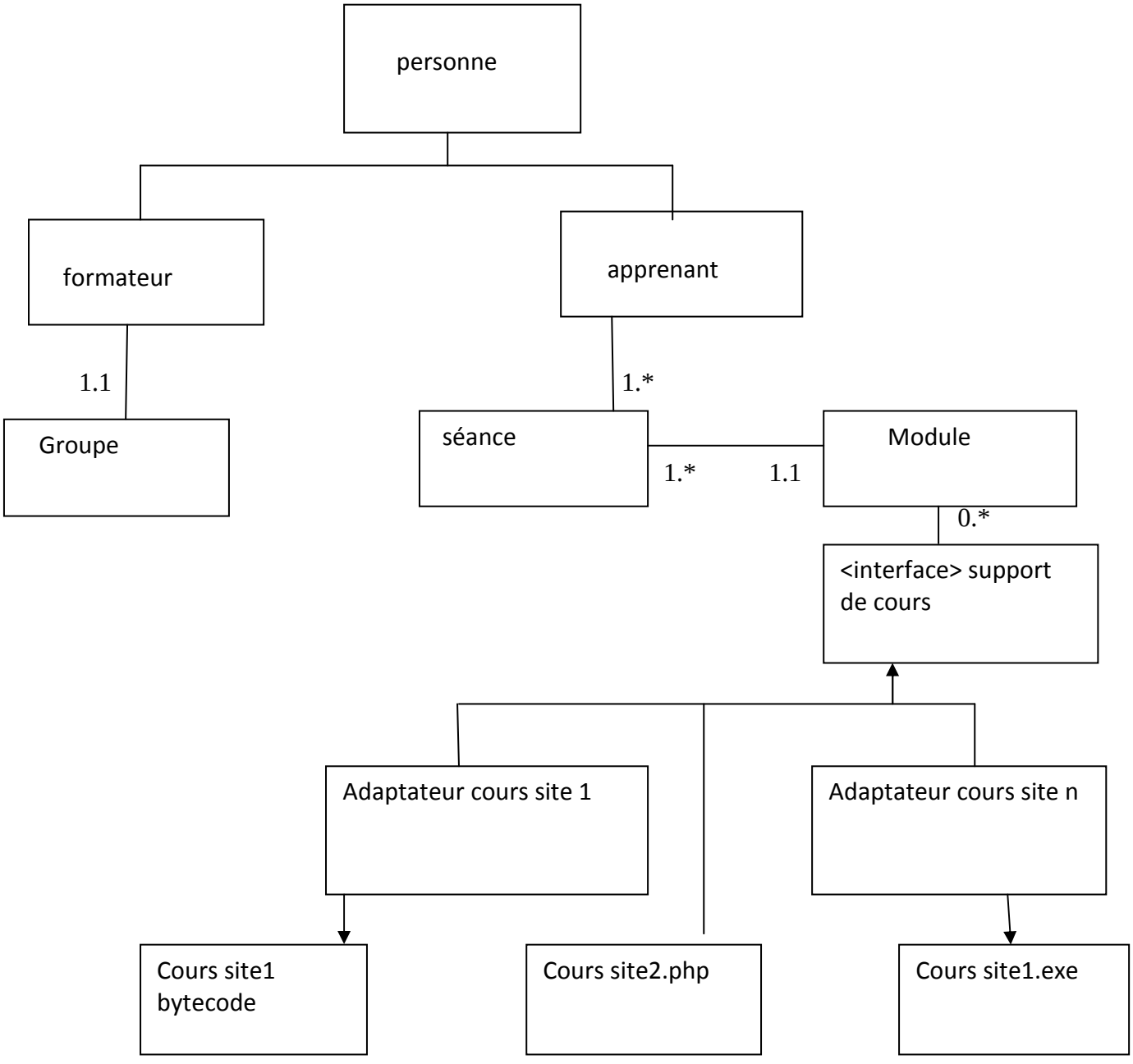
3)- l'apprenant clique sur le lien **«** Accéder au cours synchrone »;

4)- le système vérifie dans la base de données qu'une séance est prévue pour l'apprenant ;

5)- Si la séance est effectivement prévue le système affiche la page «cours synchrone », sinonle système affiche le message **«** vous n'avez pas de séance synchrone prévue en ce moment»

3.2.2.diagramme de classe :

Figure5.9. diagramme de classe



5.4. Conception de la base de données

On peut définir une base de données de la manière suivante :

« Une base de données est une collection de données cohérente et structurée ».

Pour concevoir la base de données du système, nous avons commencé par recenserles différentes entités qui interviendront dans l'application.

En se basant sur ces entités, et en respectant les différentes règles du modèle relationnel nous avons déduit les tables de la base de données.

5.4.1. Règles de gestion

- Un module peut avoir 0 ou plusieurs supports de cours.
- Un support de cours correspond à 1 seul module.
- Un formateur peut ajouter 1 ou plusieurs supports de cours.
- Un apprenant appartient à 1 seul groupe.
- Un groupe est constitué de 0 ou plusieurs apprenants.
- Un formateur présente zéro ou plusieurs séances.
- Un module est fait dans zéro ou plusieurs séances.
- Une séance est présentée par 1 seul formateur à 1 seul groupe et concerne 1 seul module.
- Un groupe fait 0 ou plusieurs séances.

Pour avoir le modèle relationnel, il suffit de faire un mapping Objet/Relationnel qui consiste à transformer chaque classe à une table, et cela vue que les classes présentées sont pas complexes (pas de classe interne).

5.4.3. Le modèle logique de données

Apprenant (id_apprenant,nom_apprenant, prenom_apprenant, adresse_apprenant date_naiss_apprenant, tel_apprenant, confirme_apprenant).

Formateur (id_Formateur, nom_form, prenom_form, adresse_form, date_naiss_form, grade_form, tel_form, confirme_form)

Seance(id_seance, intitule seance, date_seance, heure_deb_seance, heure_fin_seance, id_Formateur*, Id_groupe*, id_module*).

Groupe(id_groupe,nom_groupe, description)

Module (id_module, nom_module, descript_module)

Support_cours (id_support, nom_support, chemin_support)

5.5.Conclusion

Dans ce chapitre, nous nous sommes exclusivement concentrés sur les aspects analytique et conceptuel de notre application.Pour parvenir à nos fins, nous avons décidé d'utiliser le principe de protection des variationsainsi que le langage de modélisation UML et de son extension pour le Web.

Pour la phase d'analyse, nous avons définis divers cas d'utilisation puis nous avons élaboré leurs diagrammes de cas d'utilisation, en revanche, pour la conception , nous nous sommes attelés à construire les diagrammes de séquence et de classes.Nous avons conclu cette partie par la modélisation des tables contenues dans notre base de données.

Le chapitre suivant sera consacré à la partie réalisation de notre application ainsi que les différentes fonctionnalités dont elle dispose.

Chapitre VI

Réalisation et mise en oeuvres

6. Introduction

La réalisation du produit est le résultat des activités d'analyse et de conception de tous processus de développement logiciel, elle traduit les différents modèles issus des ces activités d'analyse en code exploitable. En antre, la réalisation concerne aussi le déploiement de l'application dans son environnement réel.

Nous y verrons en effet les langages utilisés ainsi que les différentes technologies utilisées pour son développement (système d'exploitation, outils de conception Web…).

6 .2. environnement logiciel et matériel

Pour la réalisation de notre travail, nous avons utilisé un micro-ordinateur portable, ces caractéristiques sont :

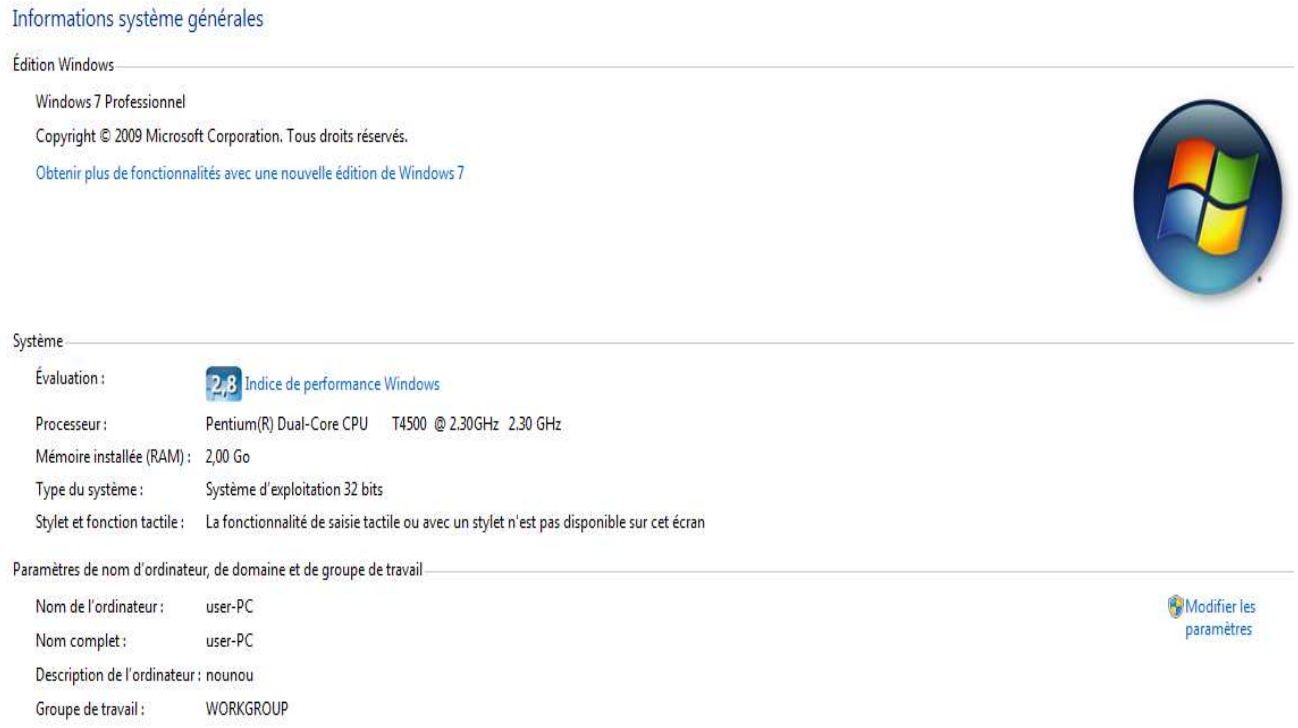


Figure 6.1.Caractéristiques matérielles et logicielles.

Pour la réalisation de notre application, nous avons fait appel à plusieurs outils (logiciel).

Système d'exploitation : Windows 7

Serveur Web : Serveur apache offert par WAMP Server

Serveur de bases de données :

- Un serveur MySQL.
- Une interface graphique PHPMyAdmin pour manipuler les bases de données.

Langages utilisés :

- Pour la partie statique de l'application : le langage HTML.
- Pour la partie dynamique de l'application : le langage PHP.
- Langage de requête MySQL pour interroger la base de données.

6.3. Environnement de programmation

On a utilisé le Dreamweaver CS4 car il permet d'écrire des programmes avec les deux langages HTML et PHP.

6.3.1. Le serveur Web Apache

le logiciel apache est un serveur http en Open Source , c'est le fruit de travail d'un groupe volontaire the Apache Group , la première version est apparue en Décembre 1995, par la suite des centaines d'utilisateurs ont contribué à son amélioration, c'est le serveur Web le plus utilisé au monde actuellement, car il englobe plusieurs avantages dont :

- C'est un Serveur gratuit.
- Un niveau de performance élevé pour des besoins matériels modestes.
- Modulaire,
- Configurable.
- Robuste
- Très portable (Linux, Windows …) contrairement à l'IIS (Internet Information Service).

6.3.2. Le serveur de base de données MySQL

MySQL signifie « My Structured Query Language » en anglais.

Est un système de gestion de base de données (SGBD). Selon le type d'application. Sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (application web principalement) que pas des professionnels, en concurrence avec Oracle et Microsoft SQL Serveur.

Elle fonctionne avec le système d'extraction de données SQL. Ce SGBD fonctionne sous Linux et Windows. MySQL est généralement utilisée pour des applications web.

Les caractéristiques de MySQL :

- Grande vitesse de traitement.

- La fiabilité.
- Compatibilité SQL.
- Sécurité.

6.3.3. L'outil PHPMyAdmin

L'outil PHPMyAdmin est développé en PHP et offre une interface intuitive pour l'administration des bases de données du serveur.

PHPMyAdmin permet de :

- Créer/ supprimer de nouvelles bases.
- Créer/modifier/supprimer/copier des tables.
- Afficher/ajouter/modifier/supprimer des tuples dans des tables.
- Effectuer des sauvegardes de la structure et/ou des données.
- Effectuer n'importe quelle requête.
- Gérer les privilèges d'accès des utilisateurs.

Les fenêtres qui suivent représentent des interfaces de PHPMyAdmin :

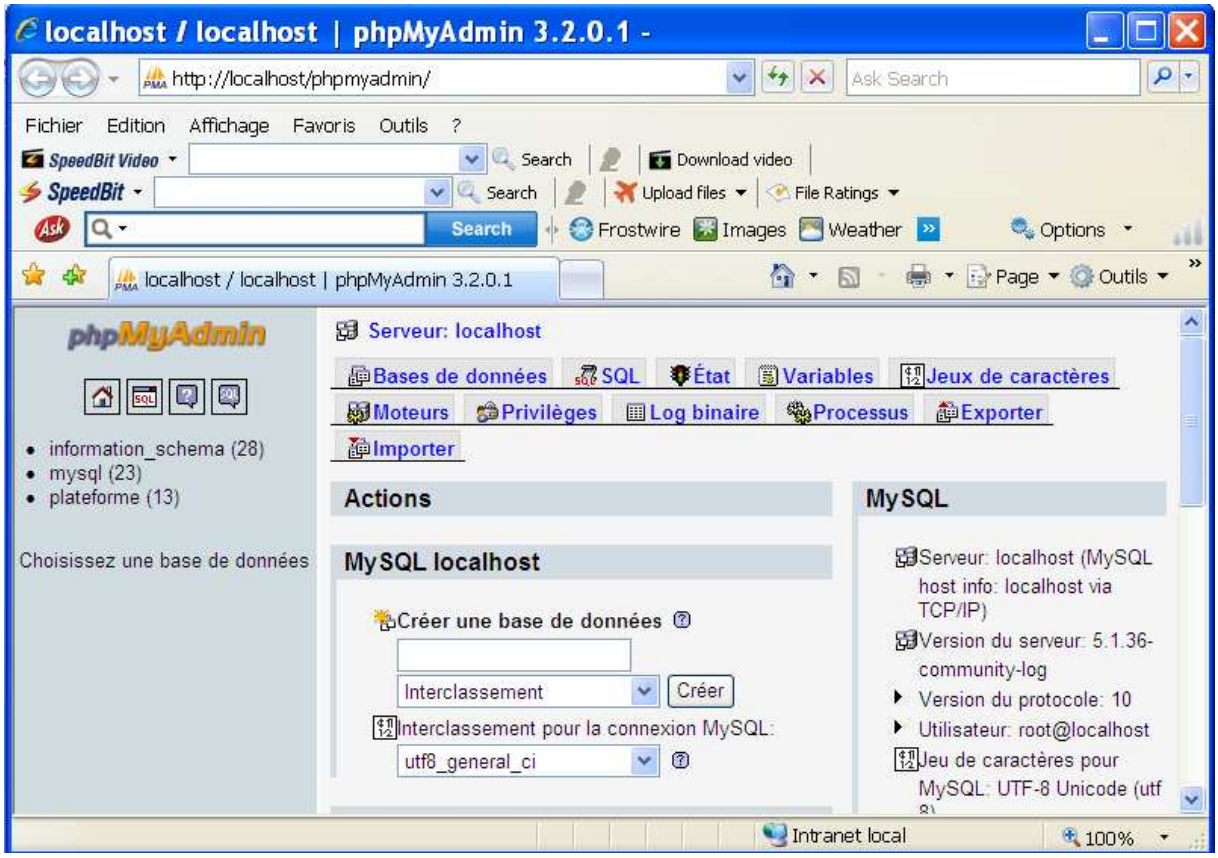


Figure 6.2- .Interface PHP MYADMIN-administration.

6.4. Les outils de développement

a- WAMP Server

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

b- Adobe Dreamweaver

Adobe Dreamweaver CS4 est un outil convivial et très puissant destiné à la conception, au codage et au développement de sites, de pages et d'applications Web (éditeur HTML professionnel). Quel que soit l'environnement de travail utilisé (codage manuel HTML ou environnement d'édition visuel), dreamweaver propose des outils qui aident à créer des applications Web.

Les fonctions d'édition visuelles de Dreamweaver permettent de créer rapidement des pages sans rédiger une seule ligne de code. On peut rationaliser les tâches de développement en créant et en modifiant des images dans Adobe Fireworks ou toute autre application graphique, puis en les important directement dans Dreamweaver, ou en ajoutant des objets Adobe Flash.

Dreamweaver propose également un environnement de codage complet comprenant des outils de modification du code (comme la coloration des codes et la création de balises) ainsi que des documents de référence sur les feuilles de style en cascade (CSS-Cascading Style Sheets), java scripts et dreamweaver permettent d'importer des documents HTML codés manuellement sans en modifier le code pour pouvoir ensuite reformater ce dernier avec le style de formatage choisi par le programme.

Dreamweaver permet aussi de créer des applications Web reposant sur des bases de données dynamique au moyen de technologies serveur comme ASP, JSP et PHP.

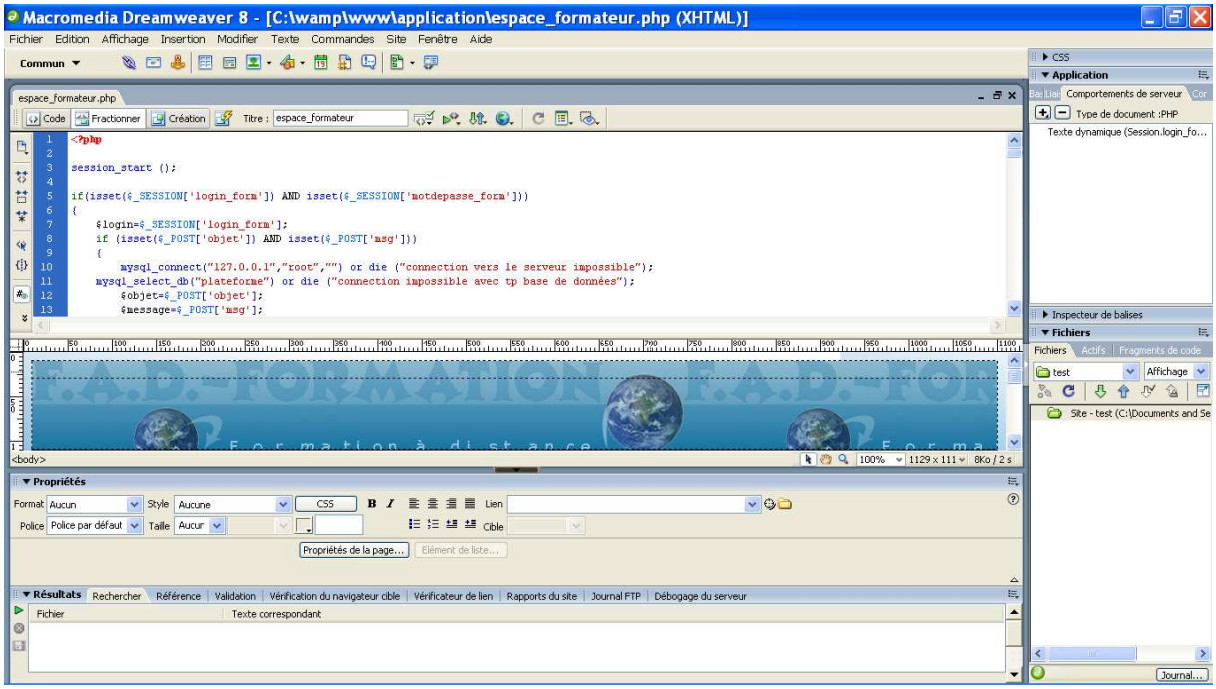


Figure 6.3 .Interface Dreamweaver.

6.5. Les langages utilisés

1- HTML

HTML signifie Hyper Text Markup Language. Comme son nom l'indique, c'est un langage qui permet de définir l'habillage d'un document.

Une page HTML est en fait un fichier texte sauvegardé sous l'extension « .html », enrichi d'un certain nombre de codes ou commandes, appelés balises.

Ces balises sont toujours exprimées sous la forme d'un mot clé, encadré par les caractères ''< '' et ''> ''. Exemple : <balise>.

Pour la plupart des balises, il existe une balise de fermeture associée, reprenant le même nom , mais précédé du caractère ''/'' . Exemple : </balise>. La commande spécifiée s'applique donc uniquement au texte situé entre le couple de balises ainsi formé.

Notons que :

- Une balise peut indifféremment être indiquée en minuscules ou en majuscules,
- Le formatage « manuel » du document (espaces, saut de lignes, …) est toujours ignoré.

Par exemple : <HTML>…</HTML>est interprété de la même façon par le navigateur Web que la syntaxe sur plusieurs lignes.

Toute page doit débuter par la balise <HTML> et se termine par </HTML>.

Entre ces deux balises, on définit deux zones : l'en-tête, spécifié par les commandes <HEAD> et </HEAD>, ainsi que le corps, délimité par : <BODY> et </BODY>. Ce qui donne, comme structure de base :

<HTML>

…

</HEAD>.

<BODY>

…

</BODY>

</HTML>

Dans l'en-tête, on ne met généralement qu'une seule information, le titre du document qui sera affiché en haut de la fenêtre du navigateur et qui apparait dans les listes d'URL gérées par un navigateur (une sorte d'annuaire).

Ce titre est indiqué entre les balises <TITLE> et <TITLE >. Exemple : <TITLE> ceci est un titre </TITLE>.

Dans le corps, on met en fait tout le document à afficher (texte , définition des images , etc.).

Signalons l'existence d'une balise de commentaire, qui peut être utilisée partout dans les documents HTML, définie comme suit :

< !—ceci est un commentaire-- !>

Les commentaires ne sont jamais affichés à l'écran du navigateur.

<HTML>

<HEAD>

<TITLE>titre du document</TITLE>

</HEAD>

<BODY>

…

…

</BODY>

</HTML>

2- Le PHP :

LE PHP est actuellement le langage de script (côté serveur) le plus utilisé pour la réalisation de sites Web dynamique. Langage très complet, il permet de réaliser la plupart des applications, mais il est principalement employé pour mettre des bases de données en ligne dans les sites Web.

Les principaux atouts du PHP sont :

- La simplicité d'écriture de scripts qui sont exécutés directement sur le serveur, avant l'envoi de la page à l'internaute. ;
- La possibilité d'inclure le script PHP au sein d'une page HTML ;
- La richesse fonctionnelle : PHP comporte plus de 1000 fonctions, parmi lesquelles les fonctions d'images, les protocoles Internet, etc.

Pour que le serveur Web interprète le script PHP, les lignes de code PHP doivent être délimités par les expressions : < **?** Et **?>** ou **< ? PHP** et **?>.**

Exemples :

```
< ?
...
Scripts PHP
…
?>
```

```
< ? PHP
…
Scripts PHP
…
?>
```

Les lignes comprises entre les délimiteurs (donc les instructions) doivent être terminées par le point-virgule **« ; ».**

La 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 plateformes Unix , Linux , et Windows.

- **Les sessions :**

Les sessions sont utilisées :

- Pour conserver de page en page les valeurs de certaines variables,
- Pour pister le parcours du visiteur.
- Détruire les variables utilisées conservées au sein de la session.

Les sessions sont activées manuellement par la commande **session_start ()** ou automatiquement si session.auto_start est a 1 dans le fichier de configuration php.ini.

Les sessions tout au long de la visite d'un internaute sur le site, de conserver des informations de façon transparente par la fonction **session_register ().**

Le serveur attribut à chaque visiteur un identifiant unique par la fonction **session_id ().** le serveur détruit toutes les informations sauvegardées tout au long de la session par la fonction **session_destroy ().**

Dans notre application nous avons utilisé les sessions pour plusieurs raisons :

- Pour l'espace administrateur :

  Conserver le mot de passe de l'administrateur depuis son authentification pour qu'il ne se ré-identifie pas, à chaque fois, au sein de sa session en utilisant **session_register ('password'**), et le tester à chaque fois avec **session_is_registered ('password').**

6.6. Présentation de quelques interfaces de l'application web réalisée

Nous allons présenter dans ce qui suit, les principales interfaces illustrant le fonctionnement de l'application. Commençons par celles liées au client (ou visiteur) puis nous donnerons celles liées à l'opérateur et à l'administrateur.

6. 6 .1 La page d'accueil : (authentification)

Figure 6.4. Page d'Accueil

C'est la première page téléchargée et visualisée par l'internaute, elle permet de s'authentifier pour découvrir les différentes fonctionnalités offertes par la plate-forme.

6.6 .2 Espace administrateur :

Figure 6.5. Page de l'espace administrateur

6.6.3 Espace formateur :

Figure 6.6 :Page de l'espace formateur

6.6 .4 Listes des formations :

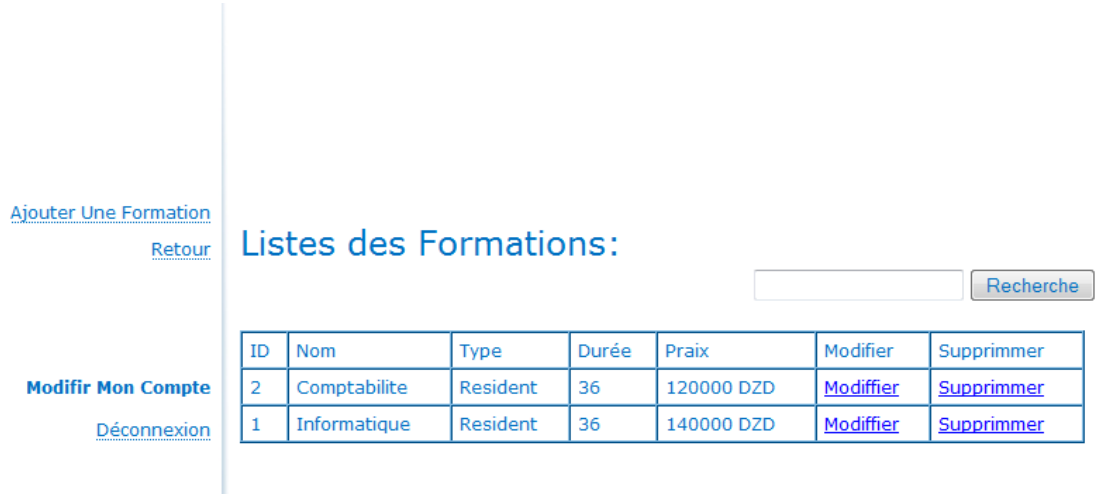


Figure 6.7. Page de listes des formations

6 .6.5 Espace Etudiant :

Figure 6.8. Page Espace Etudiant

6.6 .6 Ajouter un cours :

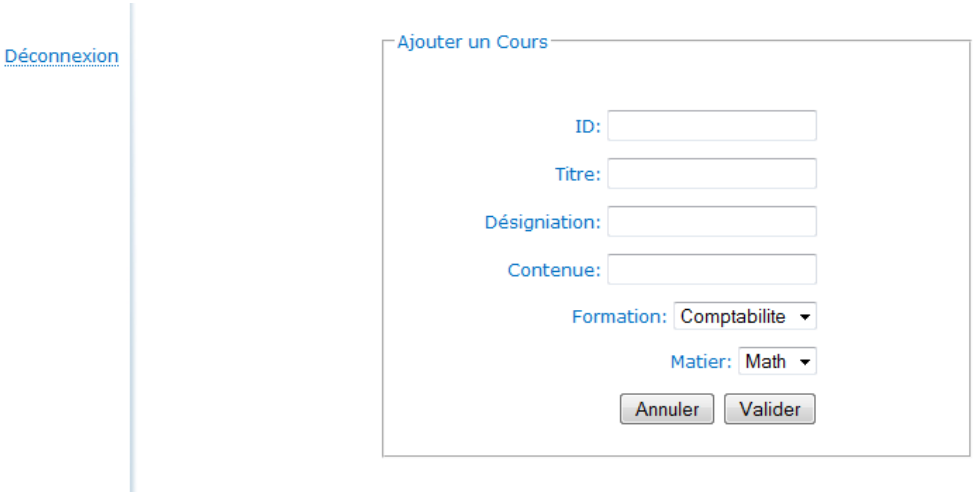
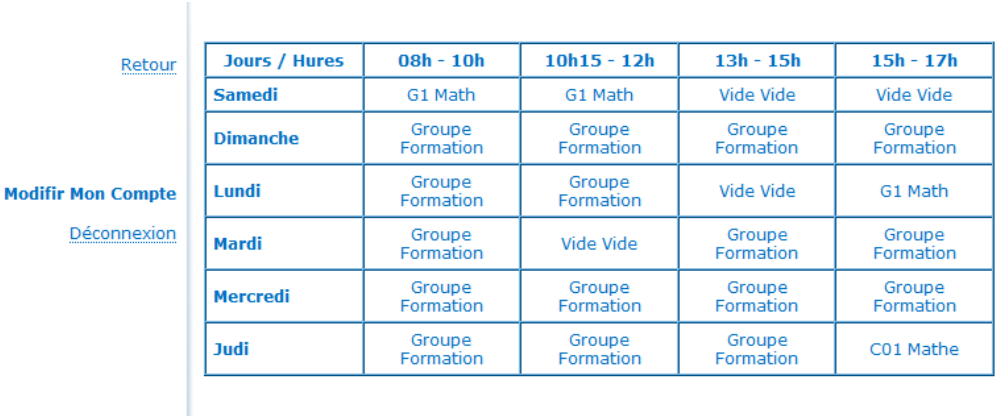


Figure 6.9. Page d'ajout de cours

6.6 .7 Tableau du planning :



Jours / Hures	08h - 10h	10h15 - 12h	13h - 15h	15h - 17h
Samedi	G1 Math	G1 Math	Vide Vide	Vide Vide
Dimanche	Groupe Formation	Groupe Formation	Groupe Formation	Groupe Formation
Lundi	Groupe Formation	Groupe Formation	Vide Vide	G1 Math
Mardi	Groupe Formation	Vide Vide	Groupe Formation	Groupe Formation
Mercredi	Groupe Formation	Groupe Formation	Groupe Formation	Groupe Formation
Judi	Groupe Formation	Groupe Formation	Groupe Formation	C01 Mathe

Figure 6.10. Page du planning

6.7. Conclusion :

Dans ce dernier chapitre nous avons fais la présentation de l'étape réalisation de notre application. Ainsi, nous avons présenté les outils logiciels qui nous ont permis de réaliser cette application, à savoir l'environnement de développement, les langages de programmation et les logiciels de création de pages web, de traitement d'image et d'animation. Puis le fonctionnement de notre site a été décrit en présentant plusieurs interfaces commentées.

conclusion

Conclusion générale

L'essor d'Internet a amené le développement d'ensembles de dispositifs de formation qui reposent essentiellement sur des enseignements conceptuels ou des études de cas, sous la forme de télé-Cours, télé-TDs ou télé-Projets, sans possibilité de réelles activités pratiques. Pourtant, le campus virtuel de demain doit intégrer ce type d'activités au sein d'un ensemble de téléactivités pédagogiques afin de confronter réellement l'apprenant aux dispositifs technologiques.

La recherche bibliographique présentée dans ce mémoire a montré que de nombreuses solutions sont proposées se limitant à un besoin spécifique dans une discipline donnée. Or, une telle solution ne peut s'intégrer dans une plate forme de téléformation qui favorise l'adaptabilité, la réutilisation et l'échange de ressources pédagogiques entre plates-formes.

Forts de ces constats, nous avons entrepris d'élaborer un système d'e-learning capable de se greffer à une plate-forme de téléformation existante afin de fournir un service continu. L'objectif d'un tel système est de pouvoir gérer des télé-TPs indépendamment de toute discipline et de tout dispositif technologique, ouvrant ainsi la voie à des échanges et à la réutilisation de scénarios pédagogiques de télé-TP entre auteurs et tuteurs.

Apports et contributions

Nos apports et contributions à la mise à distance des cours, des travaux pratiques qui peuvent se résumer aux points suivants :

- Un état de l'art sur l'e-learning.
- Une méthodologie globale proposée en vue d'établir une interface assurant l'adaptabilité.

Bilan

Nous résumons ici les résultats obtenus de nos travaux :

- Un modèle générique decours
- Un modèle générique de TP
- Un scénario générique qui décrit le déroulement d'une séance de travaux pratiques à distance.

- Des améliorations apportées à la plate forme existante (outils de collaboration et environnement d'édition et d'exécution de scénarios pédagogique.

Perspectives

Même si nos résultats sont intéressants, ils reste néanmoins difficile à mettre en pratique avec des tests en vraie grandeur et cela vue le nombre de médiateurs nécessaire. Pour cela, nous pensons qu'un méthamodèle est intéressant .

Bibliographie

http://fr.openclassrooms.com/informatique/cours/les-services-web

https://www.ibisc.univ-evry.fr/~tmelliti/cours/CPAR/cours6.pdf

http://www-inf.int-evry.fr/cours/WebServices/Docs/Bob_WS-1.pdf

http://fr.wikipedia.org/wiki/Service_web

http://www.yoyodesign.org/doc/w3c/uri-clarification/

http://fr.wikipedia.org/wiki/Multipurpose_Internet_Mail_Extensions

http://technet.microsoft.com/fr-fr/library/aa996114(v=exchg.65).aspx

http://www.irisa.fr/prive/bcousin/Cours/3b-TLS.2P.pdf

http://www.infres.enst.fr/~talel/cours/bda/enst-xml-xpath-xslt-t3-02-03.pdf

http://home.gna.org/brillant/2004-Moniot/technos-XML.html

http://www.redcad.org/members/wajdi.elleuch/fr/paper/Expose-XML.pdf

https://moodle.insa-rouen.fr/file.php/153/CM/XML.pdf

Billois G, Bernard J, Ramel JY,Nait Abdesselam F, Milliot S « Une plateforme multimédia pour la diffusion d'enseignements synchrones sur Internet ». Article de la revue TICE.

Ahmed-ouamer R « Développement de systèmes d'EIAO dans AGEDI » Séminaire National d'Informatique » : SNITO 96, Tizi-Ouzou .