

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de TIZI-OUZOU

Faculté du Génie Electrique et d'Informatique

Département d'Informatique



Mémoire

Présenté en vue de l'obtention du titre de

MASTER II

Spécialité : INFORMATIQUE

Option : SYSTÈMES INFORMATIQUES

Thème :

Conception et réalisation d'une plate-forme télé-TP avec gestion des accès concurrentiels

Proposé et dirigé par :

M^r RAMDANE Mohamed

Présenté par :

M^{elle} MENDJEL Naima

M^f LAGA Aghiles

M^{elle} RABET Lydia

Promotion
2012/2013

Soutenu le : .../.../2013

REMERCIEMENTS

Au terme de ce travail,

Nous tenons en premier lieu à remercier le Bon Dieu pour le courage et la patience qu'il nous a donné afin de mener ce projet à terme.

Nous remercions vivement notre enseignant et promoteur M'.RAMDANE d'avoir accepté de nous encadrer et de nous avoir proposé ce thème, ainsi que pour son aide précieuse, son travail encourageant, et ses conseils judicieux.

Nous remercions également les membres de jury d'avoir accepté de juger notre travail.

Sans oublié tous les enseignants et étudiants du département d'informatique de l'UMMTO.

Un grand merci à Mr LAGA Arezki pour toute l'aide qu'il nous a apportée.

Nous aimerions aussi remercier nos familles, nos amis, ainsi que tous ceux qui ont contribué de prêt ou de loin à la réalisation de ce travail

DEDICACES

A mes Chers parents

A mes deux sœurs et mes deux frères

A mes amis (es)

Naima

DEDICACES

A mes Chers parents

A mon frère Arezki

A mes amis(es)

Aghiles

DEDICACES

A mes Chers parents

A mes frères et sœurs

A Djaber

Lydia

Résumé :

Les technologies liées à l'Internet sont désormais assez évoluées pour permettre aux établissements des formations en ligne complètes, ainsi les travaux de recherche concernant les environnements d'apprentissage à distance (EAD) foisonnent mais le concept d'enseignement pratique à distance semble relativement sous-développé.

A l'instar des travaux pratiques classiques, les travaux pratiques à distance (télé-TP) sont indispensables aux environnements de téléformation, notamment dans les disciplines scientifiques et techniques. De nombreux travaux scientifiques se concentrent principalement sur la télé-opération d'un dispositif pédagogique spécifique.

Ce travail de mémoire s'inscrit dans le domaine de la e-formation, et concerne la mise en œuvre d'une plate forme d'enseignement à distance de travaux pratiques à distance mais aussi la gestion des accès concurrentiels.

Mots clés :

EAO, TICE, e-formation, téléformation, téléTP, laboratoires distants, laboratoires virtuels, accès concurrentiels.

Tables de matières

| | |
|-----------------------------|----|
| INTRODUCTION GENERALE | 14 |
|-----------------------------|----|

Chapitre I : Généralité sur l'e-learning

| | |
|---|-----------|
| 1. Introduction : | 17 |
| 2. Notion de l'e-learning | 17 |
| 3. évolution de l'e-learning : | 18 |
| 3.1. EAO : Enseignement Assisté par Ordinateur : | 18 |
| 3.1.1. Insuffisance des systèmes d'EAO : | 18 |
| 3.2. EIAO : Enseignement Intelligemment Assisté par Ordinateur | 18 |
| 3.3. EIAH : Environnement Informatique d'Apprentissage Humain | 19 |
| 3.4. La formation à distance ou l'enseignement à distance (FAD/EAD):..... | 20 |
| 3.4.1. Les modes de communications | 21 |
| 3.5. FOAD : La formation ouverte et à distance : | 22 |
| 4. Les plates formes du e-learning : | 22 |
| 4.1. Les acteurs d'une plate-forme e-learning : | 24 |
| 4.2. Le tracking | 25 |
| 4.3. Les normes et standards AICC SCORM | 26 |
| 4.4. Exemples de plateformes e-learning :..... | 26 |
| 4.4.1. Claroline..... | 26 |
| 4.4.2. Ganesha | 27 |
| 4.4.3. Moodle | 28 |
| 5. L'e-learning dans le monde : | 29 |
| 6. Perspectives d'évolution du e-learning | 31 |
| 6.1. Les avantages d'e-learning : | 31 |
| 6.2. Les faiblesses d'e-learning : | 32 |
| 7. Conclusion : | 32 |

Chapitre II : Travaux pratiques à distance

| | |
|---|-----------|
| 1. Introduction : | 35 |
| 2. Les Travaux Pratiques (TP) : | 35 |
| 2.1. Typologie de situations (TP) : | 36 |
| 2.2. Les objectifs pédagogiques d'un TP..... | 38 |
| 2.3. Les interactions dans les TPs | 39 |
| 2.4. Cycle de déroulement d'un TP | 39 |
| 3. Les travaux pratiques à distance (télé-TP) : | 40 |
| 3.1. Définition d'un télé-TP..... | 40 |
| 3.2. Les apports des téléTP : | 40 |
| 3.3. Les limites des téléTP : | 42 |
| 3.4. Facteurs de développements : | 42 |
| 3.5. Typologie de situations (téléTP) : | 43 |
| 3.5.1. Les téléTP sur des systèmes virtuels (laboratoire virtuels) : | 44 |
| 3.5.2. Les téléTP sur des systèmes physiques (laboratoire distants) : | 44 |
| 3.5.3. Les téléTP sur des systèmes distants- virtuels (laboratoires hybrides) :..... | 45 |
| 3.6. Quelques travaux sur les télé-TPs | 45 |
| 3.6.1. Un méta-modèle pour les télé-TPs [Lelevé3, 2002] | 45 |
| 3.6.2. Le projet PEARL (Open University) | 46 |
| 3.6.3. La Robotique Pédagogique au LIUM (Université du Maine - France) | 46 |
| 3.6.4. Un Modèle Générique pour les Travaux Pratiques à Distance [Ram, 2011] | 47 |
| 4. Laboratoire virtuel et laboratoire virtuel à distance : | 48 |
| 4.1. Laboratoire virtuel :..... | 48 |
| 4.1.1. Architecture d'un laboratoire virtuel : | 48 |
| 4.1.2. Les interactions dans un laboratoire virtuel :..... | 49 |
| 4.2. Laboratoire à distance (laboratoire en ligne) : | 50 |
| 4.2.1. Architecture d'un labo en ligne : | 50 |
| 4.3. Exemple de laboratoire électronique (e-lab) : | 51 |
| 4.3.1. Le projet LVEST (Laboratoires virtuels pour l'éducation en science et technologie) : | 51 |
| 4.3.2. Le projet VITELS (laboratoire virtuel de télécommunications pour la Suisse) : | 52 |
| 4.3.3. Le projet eMersion : | 53 |
| 5. Conclusion : | 54 |

Chapitre III : Les accès concurrentiels dans une plate forme téléTP

| | |
|--|-----------|
| 1. Introduction : | 56 |
| 2. La collaboration et la concurrence dans une plateforme TéléTP | 56 |
| 2.1. Gérer la collaboration : | 57 |
| 2.2. Gérer la priorité des utilisateurs:..... | 57 |
| 3. La synchronisation et les télé Tp : | 57 |
| 4. La concurrence des taches : | 58 |
| 4.1. Les processus : | 58 |
| 4.2. Les threads..... | 58 |
| 4.3. Les taches : | 60 |
| 5. Les différents types d’intercommunications et de synchronisation : | 60 |
| 5.1. Les accès concurrents..... | 60 |
| 5.2. Sections critiques..... | 62 |
| 5.3. Exclusion mutuelle par attente active (Mutex) | 63 |
| 5.4. Les Sémaphores..... | 64 |
| 5.5. Les moniteurs | 65 |
| 5.6. Les autres possibilités..... | 66 |
| 5. Conclusion : | 66 |

Chapitre IV : Analyse et conception

| | |
|---|-----------|
| 1. Introduction : | 68 |
| 2. Objectif du projet : | 68 |
| 3. La démarche d’élaboration de notre projet : | 68 |
| 3.1. Identification des acteurs et leurs besoins fonctionnels : | 69 |
| 3.2. Diagrammes représentatifs : | 70 |
| 3.2.1. Le diagramme de contexte de l’application : | 70 |
| 3.2.2. Diagrammes des cas d’utilisation : | 71 |
| 3.2.3. Détermination des diagrammes de cas d’utilisations : | 72 |
| 3.2.4. Description textuelle des cas d’utilisation : | 75 |
| 3.2.5. Réalisation des cas d’utilisation : | 80 |
| 3.2.6. Diagrammes de classes des cas d’utilisation | 85 |

| | |
|--|-----------|
| 3.2.7. Package de base de données..... | 87 |
| 4. Représentations des différents cas d'accès à l'espace de travail : | 89 |
| 4.1. Représentation des instances d'accès à l'espace de travail : | 89 |
| 4.2. Représentation de problèmes d'accès : | 89 |
| 4.3. Représentation de la solution de problème d'accès..... | 90 |
| 5. Conclusion | 93 |

Chapitre V : Réalisation

| | |
|---|------------|
| 1. Introduction : | 95 |
| 2. L'environnement de développement et l'implémentation :..... | 95 |
| 2.1. serveur :..... | 96 |
| 2.2. Serveur de base de données : [Leierer et Stoll, 2000]: | 97 |
| 2.2.1. Serveur MySQL | 97 |
| 2.2.2. Fonctionnalités de MySQL..... | 97 |
| 2.3. Les langages utilisés :..... | 98 |
| 2.3.1. Le langage PHP : (Personnal Home Page) [Leierer et Stoll, 2000]..... | 98 |
| 2.3.2. Le langage HTML et CSS: | 99 |
| 2.3.2.1. HTML 5 et CSS 3: | 99 |
| 2.3.3. Le langage de requête MySQL : (Structured Query Language) | 100 |
| 2.3.4. Le langage JAVA : | 100 |
| 2.4. Les outils de développement : | 101 |
| 2.4.1. EasyPHP | 101 |
| 2.4.2. Eclipse 3.5.1 | 102 |
| 2.5. Autres Outils : | 103 |
| 2.6. Navigateurs : | 103 |
| 3. Mode de fonctionnement de l'application (partie TP) | 104 |
| 4. Quelques interfaces de notre application :..... | 104 |
| 4.1. La page d'accueil..... | 105 |
| 4.2. Inscription utilisateur :..... | 106 |
| 4.3. Espace auteur | 107 |
| 4.4. Espace apprenant | 108 |
| 5. Conclusion :..... | 113 |
| CONCLUSION GENERALE : | 113 |

| | |
|-----------------------------|------------|
| Annexe A :..... | 115 |
| Annexe B :..... | 131 |
| Bibliographie :..... | 149 |

Table des figures

| | |
|--|----|
| Figure I.1 :fonctionnement d'une plate forme de l'e-learning..... | 25 |
| Figure I.2 : Page principale de Claroline | 28 |
| Figure I.3 :Page principale de Ganesha | 29 |
| Figure I.4 : Page principale de moodle..... | 30 |
| Figure II.1 :Typologie des systèmes de TP | 37 |
| Figure II.2 :Les interactions dans un tp | 40 |
| Figure II.3 :phase de déroulement d'un tp..... | 41 |
| Figure II.4 :les forms de téléTP | 45 |
| Figure II.5 : Modèles de téléTP | 48 |
| Figure II.6 : Architecture globale d'un laboratoire virtuel | 50 |
| Figure II.7 : Les interactions dans un laboratoire virtuel | 51 |
| Figure II.8 : Laboratoire en ligne..... | 52 |
| Figure II.9 : partition du disque dur d'une machine | 54 |
| | |
| Figure III.1 : Cycle du vie d'un thread | 59 |
| Figure III.2 : Conflits d'accès aux ressources non partageables | 61 |
| Figure III.3 : utilisation des sémaphores pour l'exclusion mutuelle | 63 |
| Figure III.4 :Fonctionnement d'un moniteur..... | 65 |
| | |
| Figure IV.1 :Diagramme de contexte de l'application..... | 70 |
| Figure IV.2 : Structuration des activités en package | 71 |
| Figure IV.3 : Diagramme de cas d'utilisation pour le paquetage « naviguer dans l'application » | 72 |
| Figure IV.4 : Diagramme de cas d'utilisation pour le paquetage « identification » | 72 |
| Figure IV.5 : Diagramme de cas d'utilisation pour le paquetage « Suivre une séance d'enseignement »..... | 73 |
| Figure IV.6 : Diagramme de cas d'utilisations pour le paquetage «suivi et assistance des apprenants» | 73 |
| Figure IV.7 : Diagramme de cas d'utilisations pour le paquetage «participer à la messagerie » | 74 |
| Figure IV.8 : Diagramme de cas d'utilisation pour le paquetage « utiliser le forum »..... | 74 |
| Figure IV.9 : Diagramme de séquence de réalisation de cas d'utilisation « identification »..... | 81 |

| | |
|--|-----|
| Figure IV.10 : Diagramme de séquence de réalisation du cas d'utilisation « consulter cours »..... | 82 |
| Figure IV.11 :Diagramme de séquence de réalisation du cas d'utilisation « faire exercice » | 83 |
| Figure IV.12 : Diagramme de séquence de réalisation du cas d'utilisation « évaluation » | 84 |
| Figure IV.13 : Diagramme de séquence de réalisation du cas d'utilisation « Faire TP»..... | 85 |
| Figure IV.14 : Diagramme de classe général de cas d'utilisation « identification » | 86 |
| Figure IV.15 : Diagramme de classe général de cas d'utilisation « consulter cours » | 86 |
| Figure IV.16 : Diagramme de classe général de cas d'utilisation « Faire TP »..... | 87 |
| Figure IV.17 : Diagramme de classe général | 88 |
| Figure IV.18 : instance d'accès à la plateforme..... | 89 |
| Figure IV.19 : Instances d'accès aux TPi | 90 |
| Figure IV.20 : Gestion d'accès aux TPi..... | 90 |
| Figure IV.21 : Module de gestion d'accès aux TPi | 92 |
| | |
| Figure V.1 : Schéma de déploiement..... | 95 |
| Figure V.2 : Serveur MYSQL..... | 97 |
| Figure V.3 :Espace de travail de l'environnement de développement Eclipse..... | 102 |
| Figure V.4 : Espace de travail de Dreamweavre..... | 103 |
| Figure V.5 :mode de fonctionnement de la partie TP..... | 104 |
| Figure V.6 :Responsive design | 105 |
| Figure V.7 :Interface page d'accueil. | 105 |
| Figure V.8 :Barre de navigation..... | 106 |
| Figure V.9 : Interface du formulaire d'inscription..... | 106 |
| Figure V.10 : Interface espace auteur..... | 107 |
| Figure V.11 : Interface formulaire ajout cours..... | 107 |
| Figure V.12 :Interface formulaire ajout chapitre | 108 |
| Figure V.13 : Interface espace apprenant..... | 108 |
| Figure V.14.1 :Interface accéder à l'espace de travail du TP | 109 |
| Figure V.14.2 : Interface de l'espace de travail du TP..... | 109 |
| Figure V.14.3 : Interface des propositions de TP..... | 110 |
| Figure V.14.4 :Interface du TP mécanique | 110 |
| Figure V.14.5 :mise en attente | 111 |
| Figure V.14.6 :libération du TP | 111 |

Introduction générale

Introduction générale

Les Technologies de l'Information et de la Communication, Internet en particulier, ont, ces dix dernières années, envahi notre quotidien tant personnel que professionnel. Après s'être immiscé dans de nombreux domaines tels que le 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. En effet, les sites d'e-formation se multiplient du fait de l'intérêt qu'ils apportent : gain de temps, économie de transport et d'hébergement, souplesse d'utilisation, interactivité, etc.

Si l'engouement pour ce nouveau concept est croissant, l'offre de réels services est encore limitée, cantonnée aux domaines où l'enseignement théorique prime sur l'enseignement pratique et les manipulations. Il est aisé pour les informaticiens de présenter un cours en ligne, de le rendre accessible à tous, de diffuser l'image de l'enseignant, de lui permettre de répondre à quelques questions... Il est, par contre, beaucoup plus difficile de donner aux apprenants les moyens de manipuler des dispositifs technologiques (instruments de laboratoire, machines-outils, robots), ... à distance dans un cadre pédagogique. Pourtant, si l'on souhaite faire de l'e-formation un outil de formation viable et largement utilisé, une recherche en amont sur les travaux pratiques à distance (télé-TP) est essentielle.

La mise en place de télé-TP se heurte, en plus des problèmes habituellement rencontrés dans l'e-formation, à une multitude de problèmes organisationnels, humains et bien entendu, techniques. Citons à titre d'exemple : **la gestion des accès concurrents** aux dispositifs technologiques (**Arpaia *et al.*, 2000 ; Lelevé, Benmohamed & Prévôt, 2004**), la sécurité du matériel et des humains, la restitution en temps réel et à distance des événements et la mise à disposition d'outils d'édition de scénarios de télé-TP pour les tuteurs (**Lelevé, Benmohamed & Prévôt, 2005**).

A partir de cette problématique. Nous considérons les télé-TP comme une activité pédagogique semblable aux autres (cours, exercices, travaux dirigés) à la différence près qu'il y a un dispositif réel ou virtuel (voire les deux) entrant en compte dans l'activité pédagogique.

Notre approche consiste à gérer les ressources pédagogiques et surveiller les manipulations simultanées dans une plate forme que nous avons conçue et réalisée.

Plan du mémoire

Le **premier chapitre** de ce mémoire s'attache à expliquer l'e-learning sous différentes facettes, parmi lesquelles: la pédagogie, les éléments techniques, les perspectives d'évolution du e-learning, enfin de manière générale le concept d'enseignement à distance (e-learning)

Le **deuxième chapitre** dresse le concept des travaux pratiques sur les différents systèmes (physique, virtuel et hybride)

Le **troisième chapitre** traite les accès concurrentiels dans une plate forme téléTP et les différents moyens de communication et synchronisation.

Le **quatrième chapitre** aborde en détail la démarche méthodologique suivie pour concevoir l'environnement de télé-TP répondant aux exigences énumérées ci-dessus.

Le **cinquième chapitre** décrit la mise en œuvre des solutions proposées dans le quatrième chapitre.

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

En fin de document figure une annexe qui est une présentation générale d'UML.

Chapitre 1 : Généralité sur l'e-learning

1. Introduction :

Depuis quelques années, les nouvelles technologies de l'information et de la communication (NTIC) s'intègrent de plus en plus dans la vie privée et la vie professionnelle. L'application des NTIC au domaine de la formation a conduit à la création d'une nouvelle réalité appelée e-learning. De manière générale, l'e-learning désigne l'apprentissage par le biais d'outils et de ressources numériques et électroniques accessibles à distance. Depuis l'ère de l'enseignement assisté par ordinateur où le véritable outil était alors le logiciel utilisé, qui porte avec lui une stratégie pédagogique et une conception de l'apprentissage, l'évolution de la technologie a introduit la possibilité de travailler à distance, à plusieurs, où on veut, quand on veut. Cela a contribué au développement d'une seconde approche dans laquelle l'ordinateur est perçu comme un moyen de permettre un apprentissage par la découverte et l'exploration en donnant le contrôle à l'apprenant.

Dans ce chapitre nous allons présenter E-learning d'une manière générale. Premièrement, nous nous intéresserons à l'e-learning et son évolution, puis, nous présenterons les différentes stratégies et plate-forme du E-learning, et nous mettrons finalement en évidence les Perspectives d'évolution du E-learning.

2. Notion de l'e-learning :

Le terme anglais « e-learning » correspond en français à l'apprentissage électronique. L'e-learning **[Learn]** est un processus d'apprentissage à distance s'appuyant sur des ressources multimédias, qui permet à une ou plusieurs personnes de se former à partir de son ordinateur. Ce mode d'apprentissage tire parti de l'usage des technologies de l'information et de la communication (TIC). En effet, ces TIC offrent une pléiade de supports multimédias (texte, graphisme, son, vidéo, etc.) qui permettent de révolutionner l'approche pédagogique, d'employer des méthodes plus ludiques où l'interactivité joue un grand rôle, de diversifier les outils employés, de s'adapter davantage au processus d'apprentissage de l'apprenant, qui devient le pilote de sa formation. Le terme e-learning peut désigner particulièrement un dispositif de formation dont les principaux objectifs peuvent être définis comme l'autonomie

d'apprentissage, la formation à distance, l'individualisation des parcours de formation et le développement des relations pédagogiques en ligne.

3. Évolution de l'e-learning :

A ses débuts, l'e-learning était réduit à la formation assistée par ordinateur. On parlait alors d'EAO – Enseignement Assisté par Ordinateur (en anglais CBT – *Computer Based Training*).

3.1. EAO : Enseignement Assisté par Ordinateur :

L'EAO est un terme qui désigne l'application des technologies informatiques dans le domaine de l'éducation. Ces systèmes utilisent l'ordinateur en tant que support de transfert des connaissances, outil de simulation, support de communication, etc.

L'ordinateur est alors un outil pédagogique puissant permettant l'apprentissage d'un domaine particulier pour les apprenants.

3.1.1. Insuffisance des systèmes d'EAO :

Les systèmes d'EAO traditionnel restent très insuffisants à cause de leurs non prise en compte de l'individualité psychologique de l'apprenant et de sa progression individuelle, ainsi que de leurs faibles capacités à expliquer les erreurs de l'apprenant.

Cette première expérience dans l'utilisation de l'outil informatique en enseignement (ou formation) a permis de révéler les limites des produits de l'EAO. Afin d'en améliorer la qualité, les chercheurs y ont intégré de nouveaux domaines tels que l'apprentissage, la psychologie cognitive, la pédagogie et la psychopédagogie. Mais vu l'hétérogénéité des connaissances à mettre en œuvre pour la réalisation de systèmes performants, une approche Intelligence Artificielle est nécessaire.

3.2. EIAO : Enseignement Intelligemment Assisté par Ordinateur

Les capacités de raisonnement offertes par l'intelligence artificielle et les systèmes experts 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 (E.I.A.O.) qui pallient aux nombreux inconvénients des systèmes précédents. Les recherches effectuées afin d'adapter l'apprentissage au niveau de l'apprenant et par rapport à son niveau de connaissances a donné lieu à une nouvelle génération de systèmes appelés : Tutoriels Intelligents.

Les systèmes tutoriels intelligents sont des systèmes d'apprentissage un à un (tuteur - apprenant). Ces systèmes ont pour but de reproduire le comportement d'un tuteur intelligent afin de dispenser un enseignement personnalisé à l'utilisateur. Ces systèmes offrent une possibilité de génération dynamique d'exercices, des adaptations au niveau de difficultés selon les performances de l'étudiant ainsi que l'analyse de l'interprétation du comportement de l'étudiant. En effet, les systèmes tutoriels intelligents sont capables de réaliser des inférences sur des connaissances de l'étudiant, et peuvent interagir intelligemment avec lui en adaptant dynamiquement les sujets à présenter en fonction des résultats acquis et du mode d'apprentissage qui lui convient le mieux.

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

- **Le mode libre :**
Ce mode convient plus à l'enseignement de concepts abstraits et à la formation destinée aux experts. Ici l'élève doit avoir une grande liberté d'action et le pédagogue ne se doit pas d'intervenir qu'en cas d'extrême urgence.
- **Le mode guidé :**
Il convient à l'enseignement de la résolution de problèmes dans un domaine spécifique. Le pédagogue intervient à des moments bien choisis.
- **Le mode mixte :**
C'est un mode où l'initiative est partagée entre l'élève et le pédagogue. L'élève découvre lui-même son erreur sans que le pédagogue ne l'aide trop tôt ou ne le laisse trop longtemps persister dans l'erreur.

A la fin des années 80 comme le précise [Bruil, 2000], le même sigle EIAO prenait un autre sens : Environnement Interactif d'Apprentissage par Ordinateur, soulignant l'importance fondamentale de l'interactivité des systèmes.

Ce changement de sigle est plus qu'un effet de style. En effet, l'arrivée du terme « interactif » dans de tels environnements d'apprentissage recouvre d'une part les activités proposées aux apprenants et d'autre part leur capacités d'intervention pertinente et d'adaptation à leur utilisation. Nous remarquons aussi la disparition du terme « enseignement », remplacé par celui d'« apprentissage ». Ce glissement révèle qu'on s'intéresse moins au transfert des connaissances et d'avantages à la construction des connaissances par l'apprenant.

3.3. EIAH : Environnement Informatique d'Apprentissage Humain

Nous assistant depuis quelques années à un glissement des EIAO vers 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 formation repose désormais à la fois sur une dimension communicationnelle et sur une dimension éducative [Biri, 2002]. La communication et l'interaction entre des machines et des humains distribués géographiquement, donnent naissance aux Environnement Informatiques pour l'Apprentissage Humain (EIAH).

D'un point de vue diachronique, ce sigle permet de prendre en compte l'évolution récente des Technologies de l'Information et de la Communication (TIC), et élargit le champ d'étude à l'apprentissage humain dans toutes ses déclinaisons (enseignement, formation, autodidaxie, diffusion de connaissances...etc.). Il recouvre à la fois les EIAO tels que nous les avons décrits précédemment, et des EIAO qui seraient plus tournés vers la formation à distance (FAD). Cette dimension apparaît actuellement de plus en plus centrale dans le contexte des travaux fondés sur les capacités offertes par la généralisation d'internet.

3.4. La formation à distance ou l'enseignement à distance (FAD/EAD):

La formation à distance est un système de formation qui permet de se former sans se déplacer sur le lieu de formation et sans la présence physique d'un formateur. La transmission

des connaissances et les activités d'apprentissage se situent en dehors de la relation directe en face à face, dite « en présentiel » entre l'enseignant et l'apprenant.

La formation à distance est incluse dans le concept plus général de Formation Ouverte et à Distance (FOAD).

3.4.1. Les modes de communications :

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

Enseignement asynchrone :

L'enseignement asynchrone [**Elearn**] est une méthode s'adaptant aux disponibilités de l'apprenant. 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).

Les avantages d'une telle méthode sont :

- 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.

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).

Les avantages de ce type d'enseignement sont :

- 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.

Notons que plusieurs applications d'e-learning combinent les deux méthodes (synchrone et asynchrone) afin de tirer profit des avantages de l'une et de l'autre.

3.5. FOAD : La formation ouverte et à distance :

Le terme « formation ouverte et à distance » (FOAD) est la traduction et de l'adaptation par le gouvernement français du terme anglais « Open and Distance Learning ». Contrairement à la FAD, la formation ouverte ou Open Learning, permet aux apprenants des entrées et des sorties permanentes.

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

Une formation ouverte et à distance :

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

4. Les plates formes du e-learning :

Le principe étant de pouvoir accéder à ses cours depuis un poste distant (chez soi, depuis son entreprise), les lieux nécessaires au suivi d'un cursus de formation (établissements, classes, bibliothèques) n'existent plus physiquement, ils sont remplacés par le Système de Gestion des Cours ou S.G.C ou plus communément appelé LMS – *Learning Management System*) est l'élément central d'un système de formation à distance.

Une plate-forme pédagogique est un logiciel qui assiste la conduite des formations présentielles et à distance. Elle est basée sur des techniques de travail collaboratif et regroupe les outils nécessaires aux trois principaux acteurs de la formation : apprenant, tuteur, administrateur. Elle fournit à chaque acteur un dispositif qui a pour première finalité l'accès à distance au contenu pédagogique, l'auto apprentissage, l'autoévaluation et le télé tutorat via l'utilisation des moyens de travail et de communication comme la visioconférence, e-mail, forums, chats, annotations, etc. Le but est donc de combler la perte de cohésion et de stimulation de la salle que peut sentir l'apprenant devant sa machine.

Une plate forme d'e-learning doit généralement remplir les fonctions suivantes **[Paqu, 2002]** :

- La gestion du contenu, qui recouvre la production et le stockage de ressources pédagogiques (création de cours, bibliothèque de formation).
- La gestion de la formation, qui recouvre la gestion administrative (inscription des apprenants, gestion des accès et des acteurs) et l'accès aux cours (connexion, identification et téléchargement).
- La gestion de l'interactivité, qui recouvre l'accompagnement asynchrone de l'apprenant (messagerie, forum, outils collaboratifs) et les classes virtuelles synchrones (vidéo-conférences, partage d'applications).
- La gestion des compétences, qui recouvre la mise en rapport des compétences individuelles avec les besoins de formation pour bâtir des programmes adaptés.

Certaines plates formes couvrent d'autres fonctions comme l'organisation des ressources. Le fonctionnement d'une plate-forme est illustré par la figure suivante :

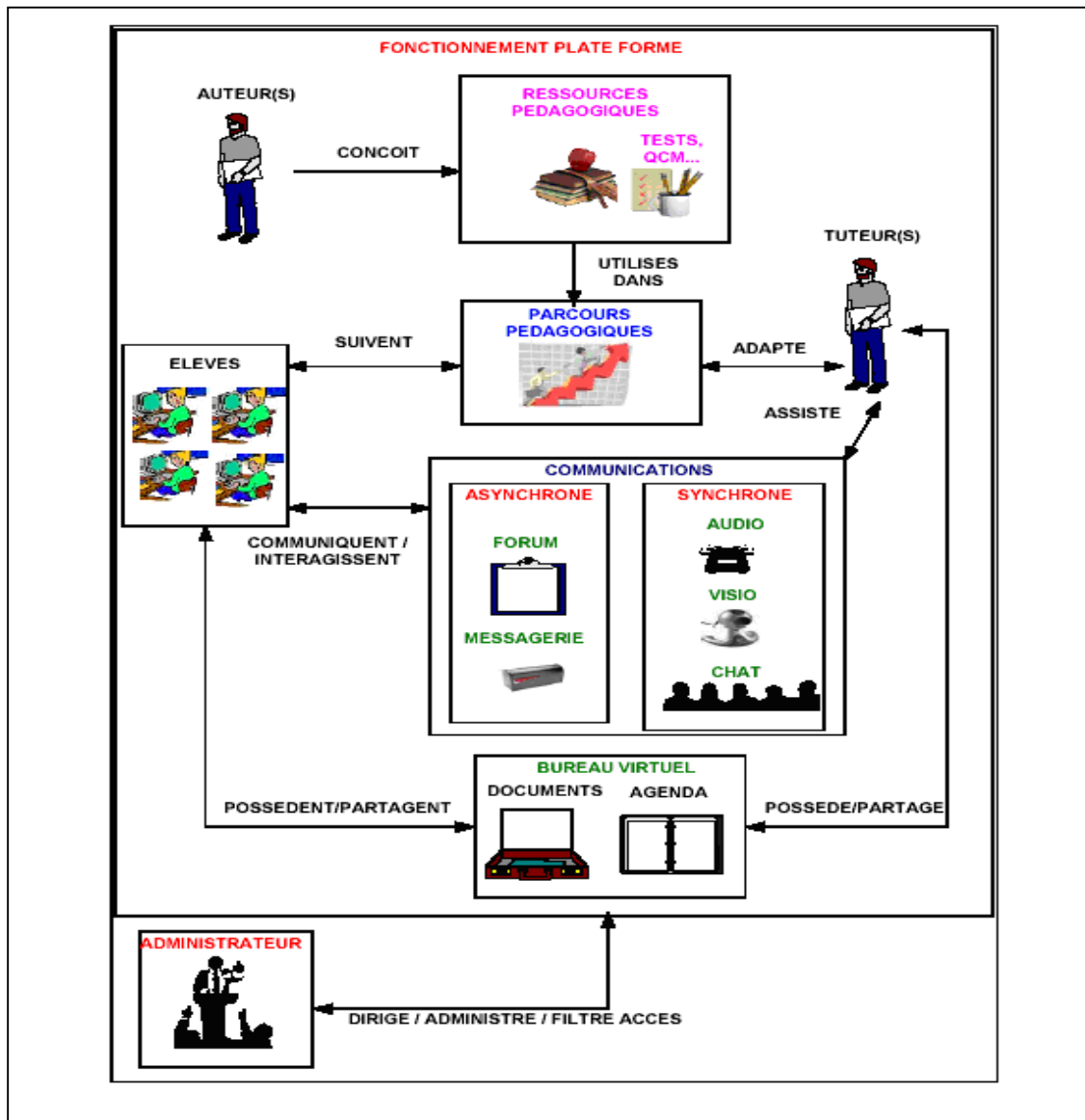


Figure I.1. Fonctionnement d'une plate forme d'e-learning

4.1. Les acteurs d'une plate-forme e-learning :

Les plateformes d'enseignements (ou de la formation) à distance intègrent des outils pour les différents acteurs, l'objectif étant de faciliter les rôles et les fonctions tenues par chacun de ces acteurs : enseignant, concepteur informatique, tuteur, apprenant et administrateur.

- **L'enseignant (ou professeur) :**

Il établit un parcours de formation personnalisé et individualisé. Il construit, adapte et maintient un système d'apprentissage. Il peut aussi avoir le rôle de présentateur. Ce rôle consiste à rendre les informations pour l'apprentissage, en d'autre termes, la

présentation des informations, la gestion des médias donnant ces dernières la clarification des contenus en réponse à des questions, l'analyse et l'évaluation des contenus des documents... etc.

- **Le concepteur informatique :**

Son rôle est de communiquer et d'animer ses groupes via les séances de télé-cours, le chat et la messagerie électronique. Orienter et évaluer les apprenants.

- **Le tuteur :**

Il accompagne les apprenants durant leur parcours pédagogique. Il a la tâche de suivre et d'orienter les apprenants, participer à la recherche de ressources, suggérer des démarches de travail et gérer les communications synchrones.

- **L'apprenant :**

L'apprenant peut apprendre à son propre rythme tout en disposant d'un ensemble d'outils et de fonctionnalités qui assistent durant le processus de son apprentissage. En effet le système lui permet de gérer ses activités et son temps, exploiter les ressources qui lui sont indiquées de façon à optimiser la quantité d'information qu'il peut en tirer, résoudre des problèmes, communiquer avec ses formateurs, échanger des informations et des idées avec d'autres apprenants via les outils de communication, s'auto évaluer et présenter des travaux qui serviront de référence à son formateur pour son évaluation.

- **L'administrateur :**

Le rôle de l'administrateur consiste à la gestion des utilisateurs du système, en leur accordant les droits d'accès et les mots de passe. Il permet aussi la gestion des inscriptions en ligne des apprenants, la gestion des classes et des groupes d'apprenants.

Les différentes plateformes ne prennent pas toujours en compte tous ces acteurs ou ne les séparent pas forcément, d'autres par contre sont plus complètes. Ainsi certaines fusionnent le rôle de l'enseignant et du concepteur informatique, ou bien encore l'enseignant et du tuteur.

4.2. Le tracking

Les fonctions de tracking permettent le suivi pédagogique des apprenants. Un grand nombre d'information concernant le parcours individuel de chaque stagiaire peut être stocké

dans le système. Il s'agit couramment des résultats obtenus dans les modules d'évaluation et de test, le temps passé sur un module de cours, le temps passé à réaliser un exercice, etc.

Ces données sont ensuite interprétées par le tuteur, qui peut ainsi se rendre compte de l'assiduité des stagiaires, des problèmes rencontrés sur certains chapitres ou sur la résolution de problèmes.

Pour que les fonctions de tracking soient pleinement opérationnelles le système complet – Système de Gestion de Contenu (SGC et contenus) devra répondre à un standard normalisé.

4.3. Les normes et standards AICC SCORM

L'objectif de ces standards est de garantir une interopérabilité entre le LMS et un contenu. Ce respect des standards AICC et SCORM va vous offrir la possibilité d'intégrer dans le LMS des contenus d'origines diverses (cours génériques produits par différents éditeurs, cours métiers produits en interne ou sous traités à un prestataire ...) et de suivre la progression des apprenants.

4.4. Exemples de plateformes e-learning :

Il existe environ plus de 200 plates-formes d'apprentissage en ligne dont une trentaine sous licences libres. Nous en citons quelques unes.

4.4.1. Claroline

Plateforme d'origine belge, initialement créée par l'Université de Louvain sous licence Open Source, puis étendue grâce à l'implication de nombreux réseaux de développeurs et de formateurs d'origine internationale.

Fonctionnalités de Claroline :

- Rédiger une description du cours
- Publier des documents dans tous les formats (texte, PDF, HTML, vidéo...)
- Administrer des forums de discussion publics ou privés
- Elaborer des parcours pédagogiques
- Créer des groupes de participants
- Composer des exercices
- Structurer un agenda avec des tâches et des échéances

- Publier des annonces (aussi par e-mail)
- Proposer des travaux à rendre en ligne
- Consulter les statistiques de fréquentation et de réussite aux exercices
- Utiliser le wiki pour rédiger des documents collaboratifs
- Gestion des contenus SCORM et Trackin
- Personnalisation de la plateforme (HomePage, style CSS).



Figure I.2. Page principale de Claroline

4.4.2. Ganesha

Plate-forme française orientée formation continue professionnelle (collaboratif autour du groupe de stagiaires, individualisation des parcours, contenus SCORM et AICC), développée par la société ANEMA.

Fonctionnalités de la plate-forme

- Forum, Messagerie intégrée, Chat, Zone de dépôt de documents
- Gère les contenus standardisés SCORM,
- Tracking et évaluation,
- Suivi pédagogique
- Administration Web
- Multilingue

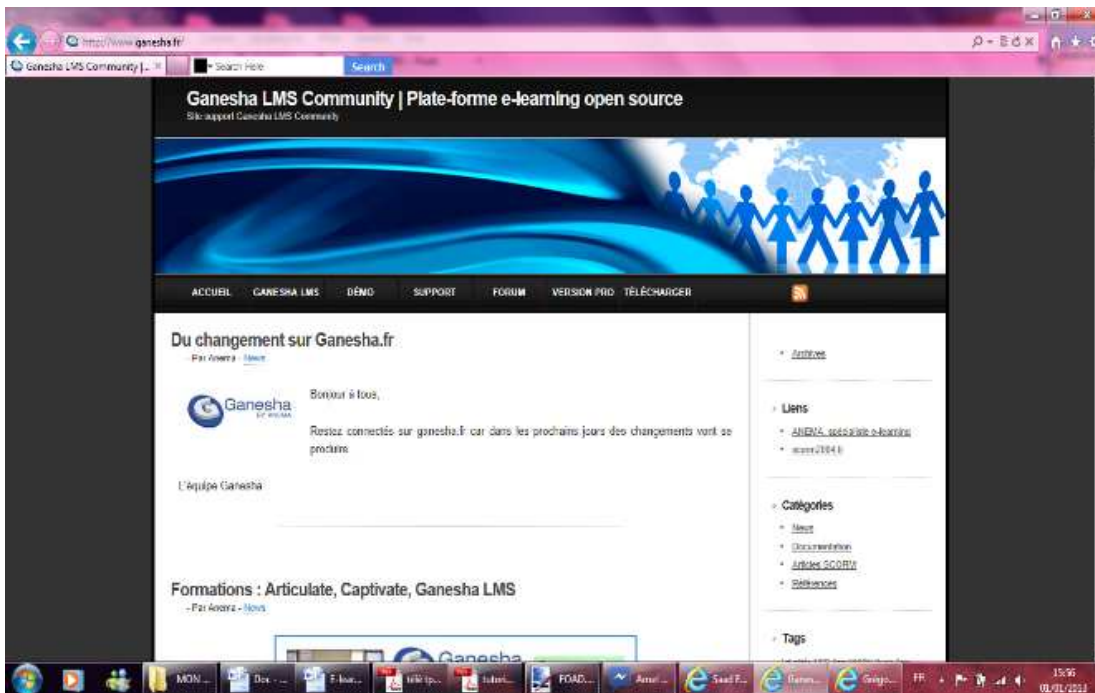


Figure I.3. Page principale de Ganesha

4.4.3. Moodle

Modular Object-Oriented Dynamic Learning Environment est une plate-forme, initiée en 1990 à Perth en Australie (Curtin University) par Martin Dougiamas qui continue à conduire le projet. Moodle est une plate-forme d'apprentissage en ligne sous licence open source servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques.

À un système de gestion de contenu (SGC), Moodle ajoute des fonctions pédagogiques ou communicatives pour créer un environnement d'apprentissage en ligne : c'est une application permettant de créer, par l'intermédiaire du réseau, des interactions entre des pédagogues, des apprenants, et des ressources pédagogiques.

De tels systèmes de e-formation sont aussi appelés dispositifs de « formation ouverte et à distance » (FOAD)

Fonctionnalités de la plateforme

- Zone de dépôt de documents,
- Multilingue,
- Administration Web,
- Suivi pédagogique,

- Tracking et évaluation,
- Messagerie,
- Générateur HTML Wysiwyg,
- Gère les contenus standardisés SCORM,
- Sondage, Forum, Chat, glossaires, wikis, Modules additionnels (Plugins).

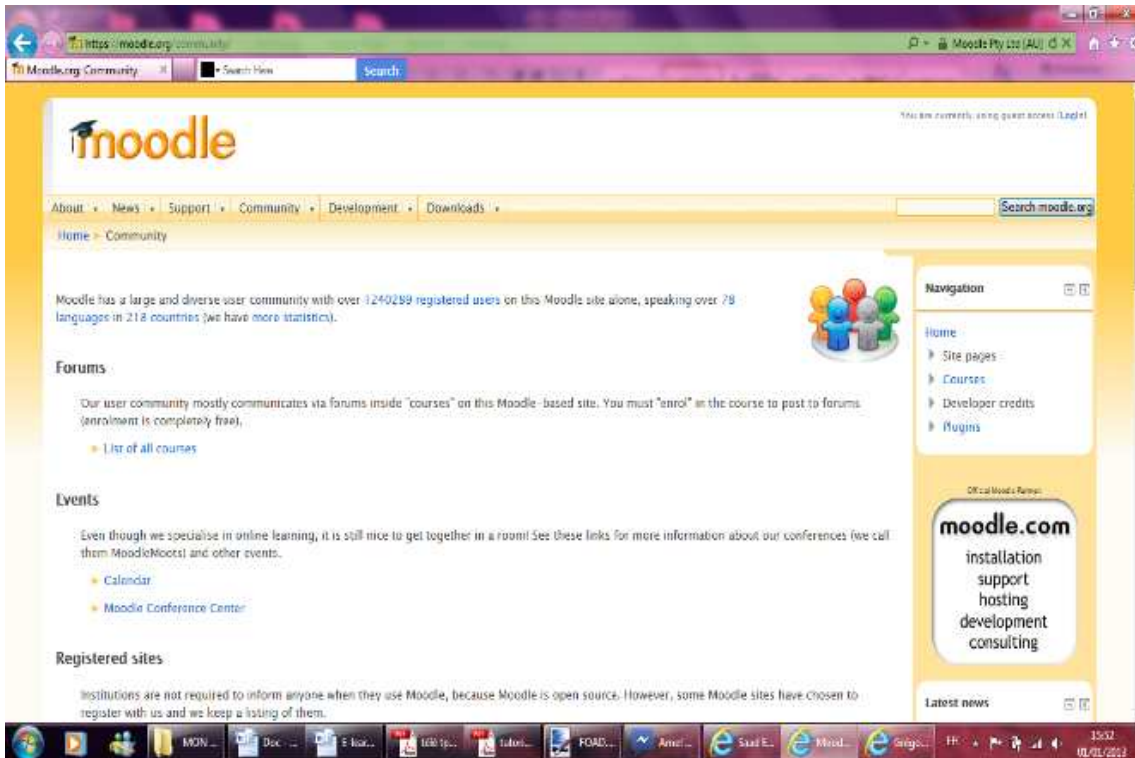


Figure I.4. Page principale de moodle

5. L'e-learning dans le monde :

L'Amérique du Nord sera considérée comme la région achetant le plus d'outils e-learning pour la période de 2010 à 2015. Cependant, sa croissance moyenne dans ce domaine est la plus faible du monde.

Le marché nord américain est arrivé à maturité. C'est la seule région où les utilisateurs du e-learning pour des raisons professionnels sont restés les meilleurs acheteurs et cela même avec une petite marge. La demande pour les formations d'apprentissage en anglais est forte.

Depuis 2010, aux États-Unis, 320 000 élèves d'écoles primaires et de collèges participent à des classes virtuelles ou à des cours par tchat interactif à plein temps. Les classes virtuelles

à temps plein ne se trouvent pour le moment qu'aux États-Unis et rarement dans d'autres pays du monde. En 2015, il y aura 17.3 millions d'étudiants américains qui prendront au moins un cours sur Internet.

Ces dernières années, l'utilisation de plateformes e-learning par les pays d'Amérique Latine a augmenté de 100% par an comparé aux autres régions du monde.

L'Europe Occidentale a enregistré le deuxième taux le plus faible en ce qui concerne la croissance du secteur e-learning (formation en ligne). Avec une croissance moyenne de 6.7%, elle dépasse tout juste le taux Nord Américain. En revanche, l'Europe Orientale a enregistré une forte croissance avoisinant les 21.8%.

Ces dernières années, la part de l'Europe ne comptait que pour 15% du marché mondial de l'e-formation.

Cependant, une nouvelle directive a été votée par la Commission Européenne en 2002. Cette dernière a permis de mettre en valeur ce support éducatif et d'assurer sa prospérité en Europe.

Certains pays comme la Hongrie, la République Tchèque, la Roumanie et la Pologne connaissent une croissance dans l'utilisation de l'e-learning de plus de 20%. C'est la Roumanie qui a enregistré la plus forte croissance dans ce secteur.

Les méthodes autonomes d'apprentissage e-learning sont relativement peu utilisées dans les entreprises et établissements scolaires en France et en Allemagne.

En Ukraine, un programme appelé « Open World » a été mis en place, avec le déploiement d'outils de formation à distance dans toutes les écoles du pays.

Les trois principaux utilisateurs de l'e-learning sont le Benelux (Belgique, Néerlande et Luxembourg), les pays scandinaves (Danemark, Finlande, Islande, Norvège et Suède) et l'Europe Orientale (Hongrie, Pologne, Slovaquie, Slovénie,...).

Les développements technologiques gigantesques de ces dernières années ont ouvert des perspectives intéressantes à l'usage de l'e-learning. En permettant non seulement un accès à distance à des contenus pédagogiques, mais aussi des échanges entre les différents acteurs impliqués. Ces développements ont eu un impact direct les domaines de formations, et nous assistons depuis des années à l'émergence de projets régionaux, comme l'Université Virtuelle Africaine (UVA) financé par la Banque Mondiale, ou le campus numérique francophone initié par le sommet de la francophonie et confié à l'Agence universitaire de la francophonie (AUF).

Il existe aussi d'autres projets, comme l'Université Ouverte Arabe, lancé par le Programme du Golfe arabe pour les organisations de développement des Nations Unies (AGFUND) et enfin le projet Avicenne financé par l'Union européenne, piloté par l'UNESCO et qui a impliqué quinze pays du pourtour méditerranéen dont l'Algérie.

Wataniya (Nedjma), opérateur de téléphonie mobile algérien, a opté pour Formademos, filiale de Demos, pour mettre en place une solution E-learning associant des formations en lignes et des formations en présentiel, soutenue par la plateforme MOS. En effet, DEMOS vient de décrocher son premier contrat e-learning en Algérie. Formademos sera chargée de la formation à distance et en présentiel de près de 1800 salariés de Wataniya télécom, et cela en management, gestion commerciale et ressources humaines.

L'université du 20Aout 1955 de Skikda possède son propre campus virtuel se basant sur la plateforme pédagogique moodle depuis 2010

En résumé, tous les pays ont réagi rapidement à l'évolution de la formation ouverte et à distance en y affectant des crédits importants et en se dotant des structures nécessaires pour accompagner et organiser son développement.

6. Perspectives d'évolution du e-learning

6.1. Les avantages d'e-learning :

➤ Pour le formateur :

- Un gain de temps (pas de déplacement) ;
- Un enrichissement dynamique de contenu ;
- Une évaluation des prés-requis plus faciles grâce à l'utilisation des jeux ou de QCM interactif ;
- Une évaluation plus précise grâce à des tests en continue tout le long de l'apprentissage.

➤ Pour les apprenants

- Acquérir un diplôme ou une qualification dans un domaine donné ;
- Compléter un enseignement ou une formation (perfectionnement) ;
- Se doter de compétences nouvelles a des fins d'enseignement ou professionnelles (actualisation de connaissances) ;
- Etudier des conditions particulières mieux adaptées ;

- Chacun est responsable de son apprentissage ou de sa formation ;
- Un gain de temps considérable ;
- Une facilité d'accès (dématérialisation de lieu) ;
- Aller plus loin par le biais de tuteur adaptés à chaque classe d'apprenants ou à l'aide de liens internet vers d'autres sites traitant du même sujet ;
- La possibilité d'échange avec les apprenants à une grande échelle à l'aide de forum ou de chat ;

➤ **Pour l'entreprise :**

- L'économie des coûts liés au déplacement (transport, hébergement, coût de l'indisponibilité) ;
- La possibilité d'avoir un nombre très important d'apprenant ;
- Plus flexibilité vis-à-vis des formations ;
- La réduction des coûts globaux.

6.2. Les faiblesses d'e-learning :

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

- La formation et la motivation des tuteurs ;
- Les apprenants ne disposent pas des compétences technique minimales ;
- La nécessité d'un équipement coûteux ;
- Les formations ne sont pas accessible à tous ;
- Le challenge des contenus pédagogiques à améliorer n'est pas encore gagné ;
- L'effort d'apprentissage par l'e-learning est plus important que la formation classique.
En effet l'apprenant ne peut être passif : il est l'acteur de sa formation.

7. Conclusion :

L'e-learning est un mode d'apprentissage qui tire parti de l'usage des technologies de l'information et de la communication à tous les niveaux de l'activité de formation. Il désigne plus particulièrement un dispositif de formation dont les principaux objectifs peuvent être définis comme l'autonomie d'apprentissage, la formation à distance, l'individualisation des parcours de formation et le développement des relations pédagogiques en ligne.

Les aspects de l'e-learning sont devenus nombreux et ne cessent de se multiplier davantage par l'émergence des nouvelles technologies de l'information et de la communication.

Dans le chapitre suivant un des vecteurs de l'e-learning sera traité. Ce sont les travaux pratiques à distance appelés aussi télé-TPs.

Chapitre 2 : Travaux pratiques à distance

1. Introduction :

Depuis quelques années une amélioration de l'observation des usages et appropriations des environnements a donné une place mieux maîtrisée aux TIC dans le marché mondial de la formation en ligne. Les TIC ont fait émerger de nouveaux outils et de nouveaux repères provoquant un changement non seulement dans les pratiques pédagogiques mais aussi dans l'organisation même qui les habitent.

Dans un premier temps, la téléformation reposait essentiellement sur des enseignements conceptuels ou des études de cas, sous la forme de téléCours, téléTD ou téléProjets (parfois), sans possibilité de réelle activité de la part de l'apprenant. Dans un deuxième temps, l'émergence d'une véritable pédagogie de « e-formation », fondée sur une réelle scénarisation, a favorisé l'apparition de produits pédagogiques performants : résolution de cas, jeux d'entreprise [Akkouche, 1996] et simulateurs pédagogiques [Azzi, 1995][pernin, 1996][Buitrago, 1999]. Chacun de ces environnements relève cependant de la virtualisation (simulation) de situations réelles.

Les travaux pratiques à distance sont indispensables aux environnements de téléformation, notamment dans les disciplines scientifiques et techniques.

Ce chapitre développe une véritable recherche sur les téléTP, laboratoire virtuel, et TéléTP sur des systèmes virtuels (Laboratoire virtuels).

2. Les Travaux Pratiques (TP) :

Toute formation, qu'elle soit initiale (du primaire au supérieur) ou continue, a besoin de proposer des sessions de formation où l'élève est confronté au monde réel afin qu'il mette en pratique ses connaissances et son savoir-faire et qu'on puisse juger de son opérationnalité. Le modèle pédagogique français a globalement adopté, plus particulièrement en formation initiale le triptyque : cours magistral, TD, TP (enrichit et complété par « le projet »). Les cours magistraux représentent souvent la première étape et sont chargés d'aborder des théories et des concepts de base. Ensuite viennent les Travaux Dirigés (TD), à mi-chemin entre la théorie et la pratique, pour approfondir ces concepts en favorisant un travail généralement en groupes d'élèves avec des exemples et des exercices courts. Enfin les Travaux Pratiques

offrent aux élèves une confrontation entre ce qui a été abordé en cours, TD et la réalité. Ce découpage a été étudié par [Ahmed-Ouamer, 1992].

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

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

2.1. Typologie de situations (TP) :

Les objets utilisés, appelés «objets de TP»¹ peuvent être physique, virtuels, réels ou hybrides. Cette section tente de clarifier ces concepts et de préciser les classes de systèmes.

Un TP peut s'organiser autour de la figure suivante :

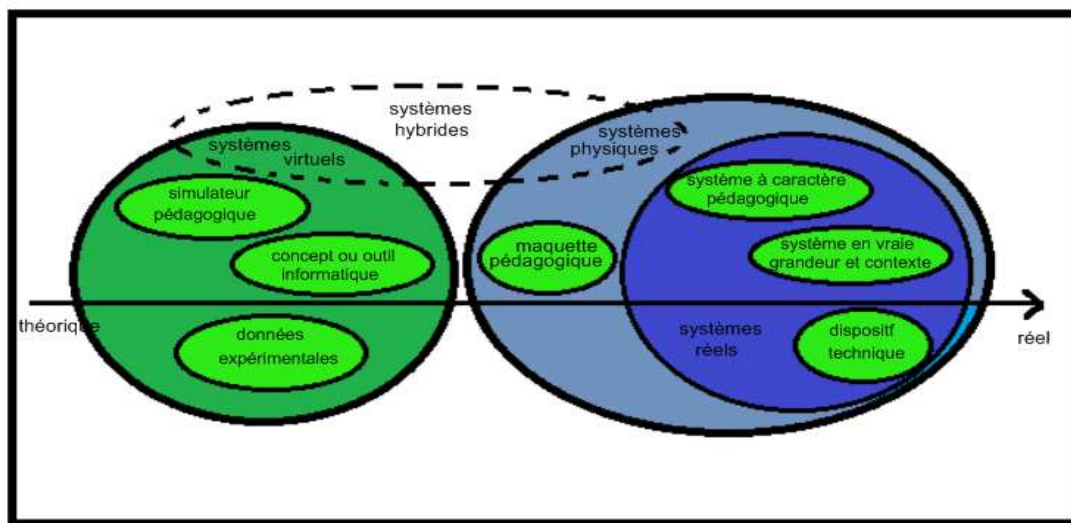


Figure II.1. Typologie des systèmes de TP

- **un système en vraie grandeur et contexte** : (unité industrielle de production, atelier, magasin,..) : généralement, il s'agit de former du personnel qualifié à la conduite de ces unités. Ce cas est rarissime, ces systèmes ne peuvent pas être utilisés dans un tel contexte pour plusieurs raisons [Azzi, 1995] :

¹ « objet de TP » : tout objet (concret ou abstrait), système ou dispositif utilisé en TP comme un oscilloscope, un robot, un simulateur ou des données expérimentales

-une perte de production inenvisageable car la plupart des installations fonctionnant en continu, des séances de formation dédiées seraient synonymes d'heures d'immobilisation de l'installation.

- un risque élevé de voir une conduite hésitante provoquer des alarmes dangereuses et risque d'être économiquement désastreux.

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

Il s'agit d'un « système réel » déconnecté de son contexte d'origine pour être dédié à la formation. Parmi les inconvénients majeurs :

- la perte du contexte (50% de l'intérêt pédagogique)
- le cout élevé de l'acquisition initiale de l'installation et sa maintenance.
- la difficulté de faire évoluer les technologies employées.

- **un simple dispositif technique :**

Cette catégorie inclut l'ensemble des équipements de mesures utilisées en chimie (pH-mètre, ...), biologie (microscope, ...), mécanique (dynamomètre, ...)...etc. dans ce cas c'est le dispositif réel qui est directement utilisé et contextualisé par le scénario pédagogique.

- **une maquette pédagogique :**

Il s'agit du modèle réduit d'un dispositif réel. Ce modèle réduit à une réalité physique (Magasin vertical, bras manipulateur,...), reproduisant l'ensemble ou une partie (certaines fonctionnalités) du système initial.

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

- **un simulateur pédagogique :**

Il s'agit d'un « programme qui imite partiellement le comportement d'un système réel ou abstrait, et qui inclut un certain contrôle pédagogique de l'apprenant » [Buitrago, 1999]. Il peut s'agir d'une simulation d'une expérimentation de laboratoire, un procédé industriel, un moyen de transport, un phénomène naturel ou un système (dangereux, trop long, trop rapide,..)

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

Les données expérimentales (réelles) sont issues d'instruments sophistiqués et coûteux, nécessitant une longue période d'acquisition, nécessitant des manipulations

dangereuses. Ces données peuvent faire l'objet d'une analyse et peuvent être utilisées pour valider des modèles théoriques acquis lors des cours magistraux ou TD.

- **la mise en application d'un concept ou outil informatique** : TP portant sur les langages de programmation (java, c, ...), les systèmes d'exploitation (windows, Unix,...) ou l'utilisation d'outils de modélisation (Merise, UML, ...).

Ces différentes situations peuvent être regroupées dans de nouvelles classes :

Systèmes réels : le terme « réel » indique que le dispositif utilisé pédagogiquement est le dispositif d'origine. Par exemple, un système en vraie grandeur et contexte, un simple dispositif technique ou système réel à caractère pédagogique.

Systèmes physiques : le terme « physique » indique la nature physique du système utilisé. C'est le cas des trois premiers exemples et des maquettes pédagogiques.

Systèmes virtuels : le terme « virtuel » désigne tout objet non physique. Les simulateurs pédagogiques sont les principaux systèmes de cette classe.

Systèmes hybrides : nous entendons ici par hybride un système composé d'une partie physique et d'une partie virtuelle. L'exemple type de ce système est l'intégration d'un sous système isolé (maquette pédagogique par exemple) dans un système globale simulé (système de production)

Au-delà de chaque classification précédente, la figure II.1 identifie la nature de chaque classe : de l'abstrait au réel. Ainsi, un simulateur pédagogique est proche de l'abstrait, tandis que les systèmes à caractère pédagogique sont plus proches du monde réel.

2.2. Les objectifs pédagogiques d'un TP

Les objectifs pédagogiques des travaux pratiques sont :

- Fournir des illustrations et des démonstrations des principes enseignés et donc une meilleure assimilation des apprenants.
- Motiver les élèves et focaliser les interactions entre apprenants et entre apprenants-formateurs.
- Développer des compétences pratiques considérées comme importantes d'un point de vue professionnel.

- Développer des compétences de travail collaboratif en équipe.
- Introduire les élèves dans la communauté de pratique des scientifiques.

2.3. Les interactions dans les TPs

Dans un TP les apprenants peuvent manipuler et visualiser le matériel (le dispositif). La séance de TP est le lieu d'un échange privilégié avec l'enseignant. Les apprenants posent plus facilement des questions profitant du fait d'être encadrés en petits groupes. C'est aussi l'occasion de plus nombreux échanges avec les autres apprenants et d'un travail en groupe.

Pour les enseignants, il s'agit d'aider les apprenants à construire un référentiel expérimental, d'évaluer leurs réactions, leur capacité de travail et leur évolution de façon individuelle, leur capacité à travailler en groupe, de leur faire prendre conscience de la performance et de l'importance de la sécurité. La figure II.2 illustre les interactions lors d'une séance de TP :

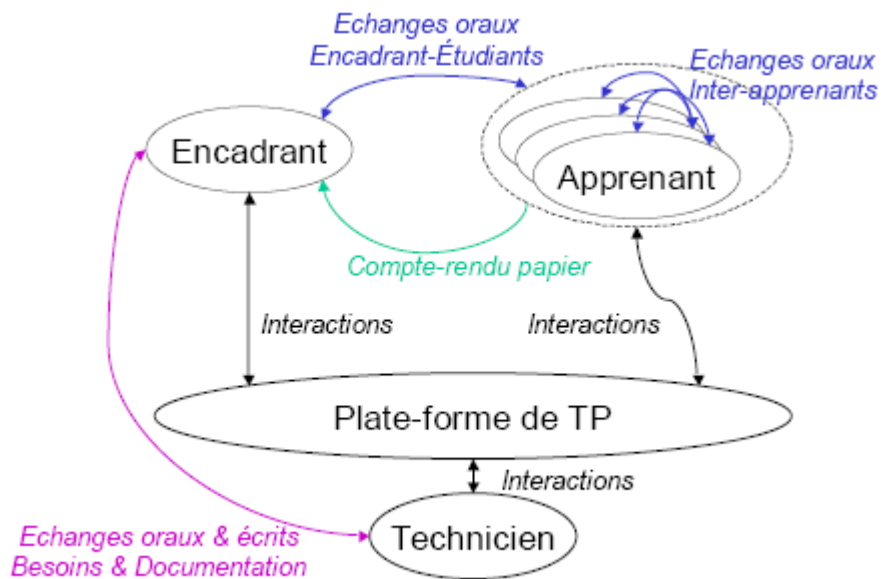


Figure II.2. Les interactions dans un TP

2.4. Cycle de déroulement d'un TP

Selon Alejo D [Alejo et al., 2003], un TP est divisé en trois étapes :

- Une phase de préparation (hors salle) : elle consiste en une lecture approfondie du texte exposant les notions théoriques utiles (étape théorique) et la manipulation

(étapes pratiques). Elle peut donner lieu à des calculs préliminaires, des analyses théoriques, ...

- La séance de TP : elle débute généralement par un rappel des objectifs du TP et une présentation du matériel puis la manipulation réalisée par l'apprenant.
- La phase de rédaction d'un compte-rendu (hors salle).

Le déroulement des trois phases d'un TP est illustré par la figure suivante :

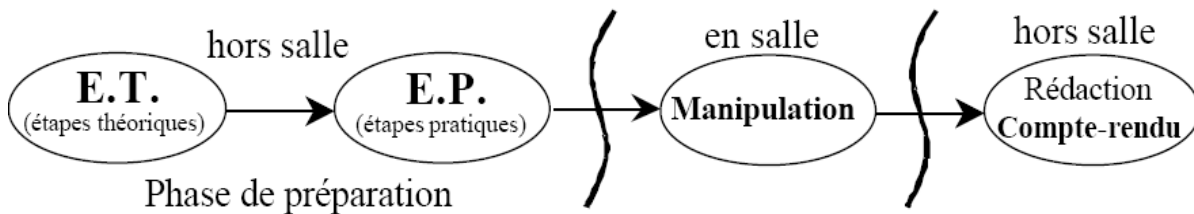


Figure II.3 : Phases de déroulement d'un TP (adapté de [Alej, 2003]).

3. Les travaux pratiques à distance (télé-TP) :

3.1. Définition d'un télé-TP

Une activité de travaux pratiques en ligne ou télé-TP désigne [Lelev, 2002] un TP classique (qu'on peut trouver dans un lycée ou une école d'ingénieurs) qui a été éventuellement modifié mais surtout étendu afin d'être accédé à distance (via *Internet*, par exemple).

3.2. Les apports des téléTP :

Les avantages immédiats des télé-TPs découlent des limites de ce type d'activités (TP) dans un contexte classique qui sont :

- **Le temps**: les tranches horaires sont limitées dans l'emploi du temps et les rotations de TP (du fait du faible nombre de systèmes accessibles comparé au nombre d'élèves) peuvent ne pas avoir lieu dans l'ordre le plus approprié pour certains groupes. En outre, les contraintes des emplois du temps font que certains TP arrivent avant le cours magistral.
- **La documentation** : Le sujet est donné (au plus tard) au moment de la séance et est parfois indisponible en dehors des séances de TP,

- **L'équipement** : certains TP nécessitent des équipements lourds et onéreux que l'on ne peut pas dupliquer (robotique, mécanique, etc.)

Les environnements de *télé-TPs* devraient offrir aux apprenants, aux encadreurs ainsi qu'aux concepteurs de nouvelles perspectives à la fois du point de vue *temporel* (accès à tout instant, adaptation dynamique du scénario, documentation contextuelle, simulation simultanée pour comparaison théorie-réalité) et du point de vue *spatial* (accès individuel ou collectif à partir d'un autre établissement). Ils peuvent être une réponse intéressante aux limitations précédentes tout en apportant de nouvelles fonctionnalités (par ex., retours d'usage : évaluation systématique et suivi des apprenants, démonstrations en ligne lors d'autres modes d'enseignement et pour d'autres publics, utilisation à des fins de recherche scientifique, ...).

Les téléTPs dépassent certaines de ces limites en facilitant :

- **Sur le plan économique** :
 - Le partage d'instruments et d'équipement lourds et coûteux entre institutions.
 - L'accès aux instruments d'un « laboratoire » n'importe quand et de n'importe où.
- **Sur le plan pédagogique** :
 - L'illustration de téléCours ou téléTD pour des apprenants distants ne pouvant pas accéder à des dispositifs réels (équivalent à une démonstration),
 - L'intégration d'outil (simulateurs, réalité virtuelle,.. ;) apportés par les TIC,
 - L'ouverture à des utilisateurs autre que la population d'origine : scénarios pédagogiques adaptés à des niveaux scolaires différents (du lycée à l'école d'ingénieur) et même au grand public en dehors des créneaux réservés aux élèves.
 - Les démonstrations mettant en valeur l'équipement et le savoir-faire de l'établissement auprès d'un grand public
 - Possibilité de reprise, par l'apprenant, d'un essai après la période réservée à cette fin : une expérience qui a échoué ou qui a été mal conduite peut être recommencée.
 - Possibilité pour l'apprenant de revoir sa démarche, de constater ses erreurs et de se corriger.

Pour les formateurs, les télé-TPs permettent d'apprécier la progression et de voir les difficultés des apprenants (échec répétitif à une étape, erreur détectée,...) et d'avoir une vision

plus précise sur l'état d'avancement de son groupe et ainsi d'éviter d'interroger l'apprenant sur l'historique de son travail [Guér, 2004].

3.3. Les limites des téléTP :

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

- **Commandabilité :**

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

- **Observabilité :**

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

- **Sécurité :**

Nous distinguons deux niveaux de sécurité :

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

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

3.4. Facteurs de développements :

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

Cette conviction s'appuie sur trois constats :

- **Contexte favorable :**

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

- **Dimension pédagogique :**

L'émergence de nouveaux standards (internet, serveurs puissants..) dans la e-formation répondant à des besoins de structuration, de réutilisation de contenu, d'interopérabilité entre plates-formes et facilite l'intégration des téléTPs, ou téléCour et téléTD coexistant déjà.

- **Dimension technologique :**

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

3.5. Typologie de situations (téléTP) :

Par opposition aux TP en présentiel, nous distinguons trois formes de téléTP (voir la figure II.4). Des études comparatives sont présentées dans [Nedic et al., 2003][Deniz et al., 2003]. Les téléTP sur des systèmes virtuels (laboratoire virtuels), les téléTP sur des systèmes physiques (laboratoire distants) et les téléTP sur des systèmes distants- virtuels (laboratoires hybrides).

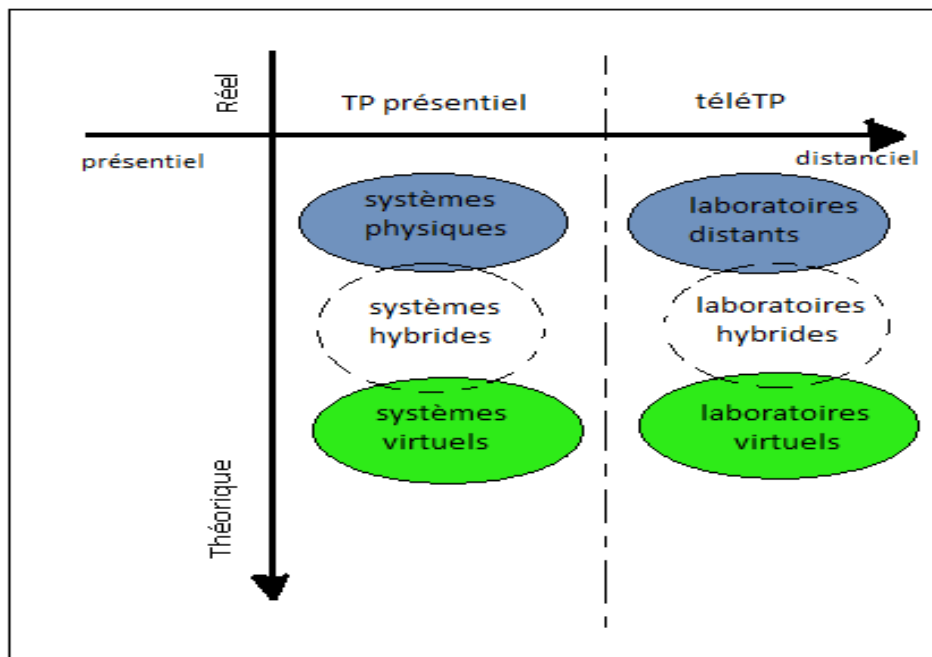


Figure II.4 Les formes de téléTP

3.5.1. Les téléTPs sur des systèmes virtuels (laboratoire virtuels) :

Les premiers téléTPs étaient fondées sur l'utilisation de simulations. L'usage des logiciels appropriés tels que Mtlab/Simulink [Dixon et al., 2002] et LabView [Chen et al., 1999][Saad et al., 2001], a contribué à harmoniser les solutions proposées. Les simulateurs informatiques offrent la possibilité de manipuler virtuellement des systèmes :

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

3.5.2. Les téléTP sur des systèmes physiques (laboratoire distants) :

Les laboratoires distants résultent simplement d'une mise à distance de plate-forme de TP physique. Ils permettent l'accès, le pilotage et la commande à distance du système à télé

manipuler (instruments d'un laboratoire par exemple) via des liaisons informatiques (en particulier internet).

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

1. Les laboratoires distants fournissant un accès à des données expérimentales
2. Les laboratoires distants offrant des moyens de télémesure
3. Les laboratoires distants proposant un type de manipulation
4. Les laboratoires distants offrant un ensemble de manipulation

3.5.3. Les téléTP sur des systèmes distants- virtuels (laboratoires hybrides) :

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

3.6. Quelques travaux sur les télé-TPs

3.6.1. Un méta-modèle pour les télé-TPs [Lelevé3, 2002]

C'est un projet visant la modélisation d'un télé-TP. Les Objectifs scientifiques derrière cette recherche est de réunir tous les outils scientifiques utiles à la modélisation d'un télé-TP : expression des besoins, formulation des objectifs, définition des contenus pédagogiques et des environnements afin de définir un noyau générique (le plus ouvert possible) proposant une architecture informatique, une méthode de mise à distance de TP, un appui pédagogique et un environnement flexible d'animation à distance. Cette approche est au carrefour de plusieurs disciplines : L'informatique, les interfaces homme-machine (IHM), la didactique et la robotique (télé-opération). Cette étude s'est intéressée aux différents aspects liés aux télé-TPs. D'un point de vue pédagogique, les télé-TPs sont considérés comme des éléments nécessaires pour les disciplines relevant des sciences de l'ingénieur. La mise en œuvre pourra être une reproduction aussi fidèle que possible du système à manipuler ou bien au contraire

les conceptualiser à travers une interface (IHM) pédagogique spécifique. D'un point de vue technologique, les télé-TPs posent un certain nombre de problèmes quant à leur mise à distance surtout l'interconnexion de la partie informatique et de la partie physique. Les auteurs de ce travail ont défini une première modélisation du système basé sur deux éléments principaux : la plate forme de télé-TP (les acteurs humains et informatiques : apprenant, encadrant, techniciens) et les concepteurs de scénarios pédagogiques.

3.6.2. Le projet PEARL (Open University)

PEARL [Coop, 2002] est un projet réalisé à l'Open University (Angleterre) visant le développement d'un système permettant des expérimentations réelles à distance dans un contexte d'e-learning où les apprenants seront capables de :

- Interagir avec l'expérimentation à distance, changer des paramètres et dans certains cas modifier et remodeler les expérimentations.
- Discuter de leurs actions, leurs prévisions, observer et analyser les résultats à travers des outils de communication/collaboration embarqués dans le système PEARL.

Ce processus a la particularité d'être réel, c'est à dire, original et non prédictible, chose qu'une simulation ne peut reproduire. Le système PEARL se compose de trois principaux composants :

- Le système de délivrance de contenus pédagogiques, d'outils de communication et de collaboration et de gestion de la formation basé sur WebCT. Ce système établit le lien avec le serveur du laboratoire qui contrôle l'infrastructure du laboratoire.
- Le LabServer : le serveur de laboratoire responsable du contrôle à distance de l'infrastructure du laboratoire.
- L'infrastructure de laboratoire à distance : composée d'une carte digitale pour chacun des contrôleurs.

3.6.3. La Robotique Pédagogique au LIUM (Université du Maine - France)

La robotique pédagogique [Leroux, 2002] au LIUM s'est basée sur plusieurs environnements technologiques, la plus aboutie est celle du micro-monde piloté par un logiciel spécifique Roboteach [Desp, 2003]. Selon S Papert [Papert] Un micro monde est un environnement où les apprenants construisent leurs connaissances en explorant et/ou en

construisant un monde constitué de micro-robots à formes multiples (tortue de plancher, micro-robots modulaires, grues, etc.) pilotés par des ordinateurs. La robotique pédagogique au LIUM est un projet de recherche et développement visant le développement d'environnements d'apprentissage avec ordinateur s'appuyant sur la conception, la construction et le pilotage de micro-robots pédagogiques. Ces micro-robots pédagogiques sont des objets techniques physiques qui sont une réduction aussi voisine et signifiante que possible des procédés et machines automatisées réellement utilisés en milieu industriel.

3.6.4. Un Modèle Générique pour les Travaux Pratiques à Distance [Ram, 2011]

Ce modèle consiste à intégrer un environnement dédié aux télé-TPs dans les plates-formes d'enseignement à distance actuelles afin de réutiliser leurs ressources pédagogiques.

La démarche propose tout d'abord la construction d'un modèle de TP à partir de l'observation et l'identification d'invariants entre différents TP (cette partie s'inspire des travaux réalisés par Hcene Benmohamed dans [Ben, 2007] [Ben, 2008]. Par la suite, le modèle ainsi obtenu est adapté pour la mise en ligne des TP (Passage d'un TP vers un Télé-TP). Finalement, le modèle est normalisé pour se conformer aux standards d'e-learning notamment IMS LD [IMS][Goun, 2005] [Moura, 2007]. La figure II.5 illustre le modèle proposé dans [Ram, 2011]

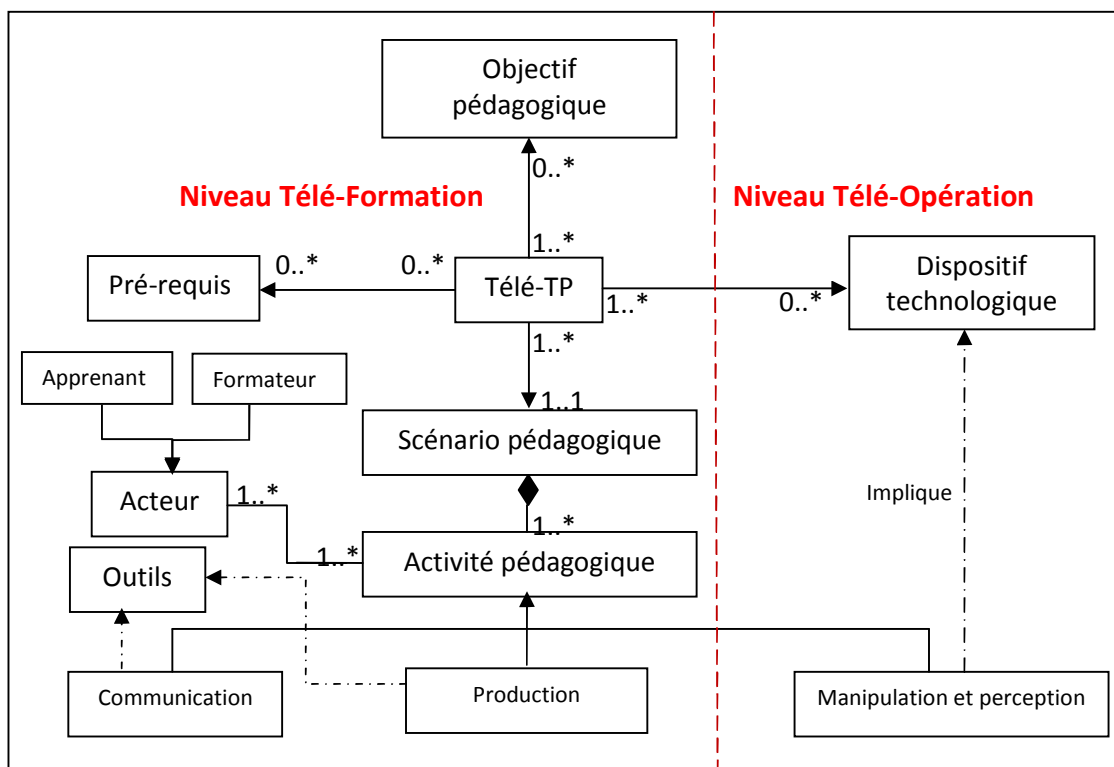


Figure II.5. Modèle de Télé-TP

4. Laboratoire virtuel et laboratoire virtuel à distance :

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

4.1. Laboratoire virtuel :

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

Un Laboratoire Virtuel (LV) est comme "un espace de travail électronique pour la collaboration à distance et l'expérimentation dans la recherche ou dans d'autres activités créatives, en vue de générer et de diffuser des résultats au moyen de technologies partagées de l'information et de la communication".

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

Un LV est caractérisé par des simulations de scénarios conçus pour être mis en œuvre à partir d'un ou plusieurs ordinateurs. Les simulations représentant des concepts théoriques ou des dispositifs se basent sur des modèles mathématiques pour une approche plus crédible . Dans certains cas de figure, il est impossible de simuler des scénarios expérimentaux ou le comportement de dispositifs réels. C'est le cas où les modèles mathématiques sont trop complexes, par manque de disponibilité de puissance de calcul ou quand les délais des traitements informatiques sont longs.

4.1.1. Architecture d'un laboratoire virtuel :

Plusieurs architectures ont été proposées pour les laboratoires virtuels notamment dans [Remi, 2006]. L'architecture type d'un laboratoire virtuel illustrée par la figure ci-dessous (figure 2.5) est composée de trois couches :

La première couche constitue le noyau du laboratoire virtuel, celle-ci est formée à son tour de composants pédagogiques (feuille de TP, manipulation, objets virtuels et résultats). La deuxième couche constitue l'élément essentiel pour les laboratoires virtuels : l'interface. Elle joue le rôle d'une passerelle de communication avec la permission d'établir des interactions entre le noyau, les acteurs du processus d'apprentissage et les ressources pédagogiques. Le laboratoire virtuel est enveloppé par une couche jugée indispensable pour le bon acheminement du travail pédagogique dénommée « ergonomie » [Falz, 1999]. Elle est un facteur très important du fait qu'elle ajoute une notion de réalisme au laboratoire virtuel.

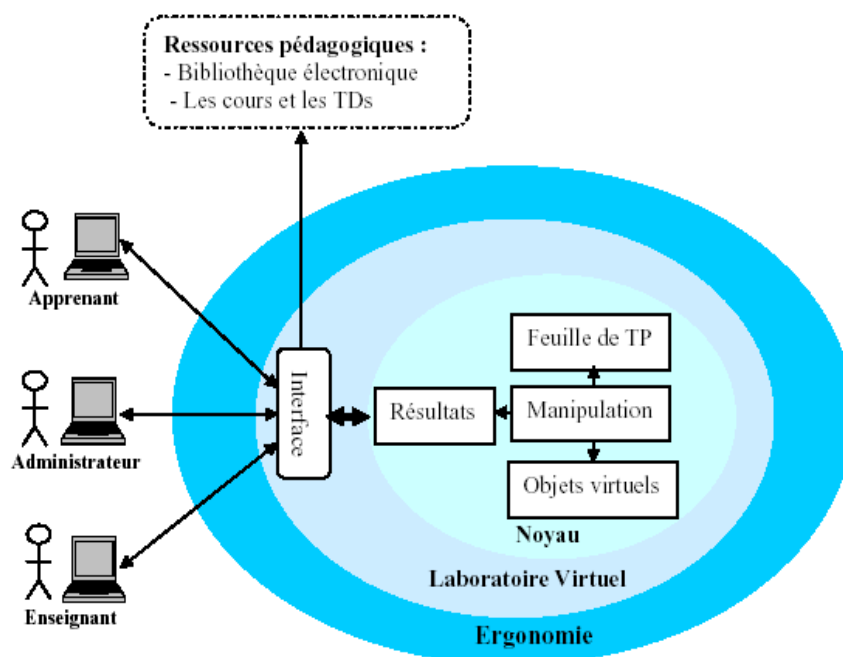


Figure II.6. Architecture globale d'un laboratoire virtuel

4.1.2. Les interactions dans un laboratoire virtuel :

Les interactions sont des actions qui permettent d'introduire un certain dynamisme. Il existe trois types d'interaction : L'interaction homme-système, l'interaction homme-homme et l'interaction système-système. Les interactions entre les différents acteurs, les ressources pédagogiques et le laboratoire virtuel sont réalisées via une interface, résumées dans la figure suivante :

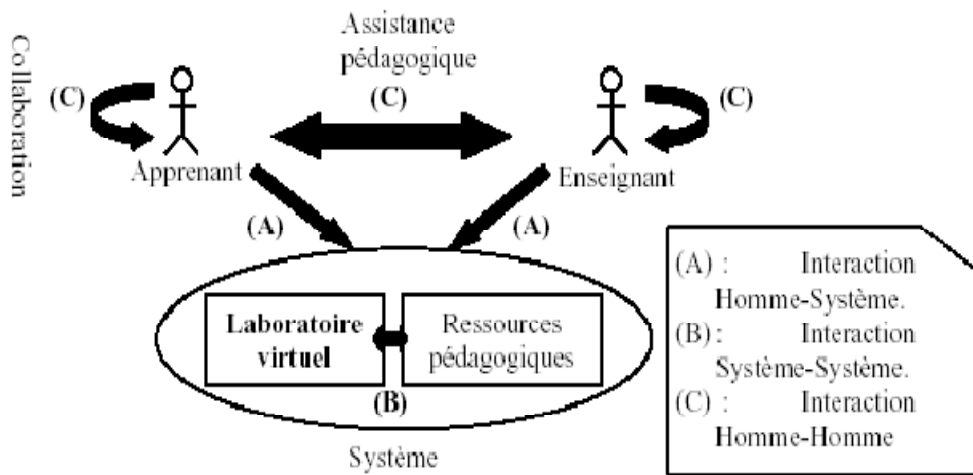


Figure II.7. Les interactions dans un laboratoire virtuel

4.2. Laboratoire à distance (laboratoire en ligne) :

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

4.2.1. Architecture d'un labo en ligne :

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

visioconférence (B) et de télémessure, il est possible d'effectuer des travaux pratiques à distance en mettant en jeu les équipements réels distants tels que l'analyseur de spectre et l'oscilloscope branché à un montage électronique, tous au site A.74

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

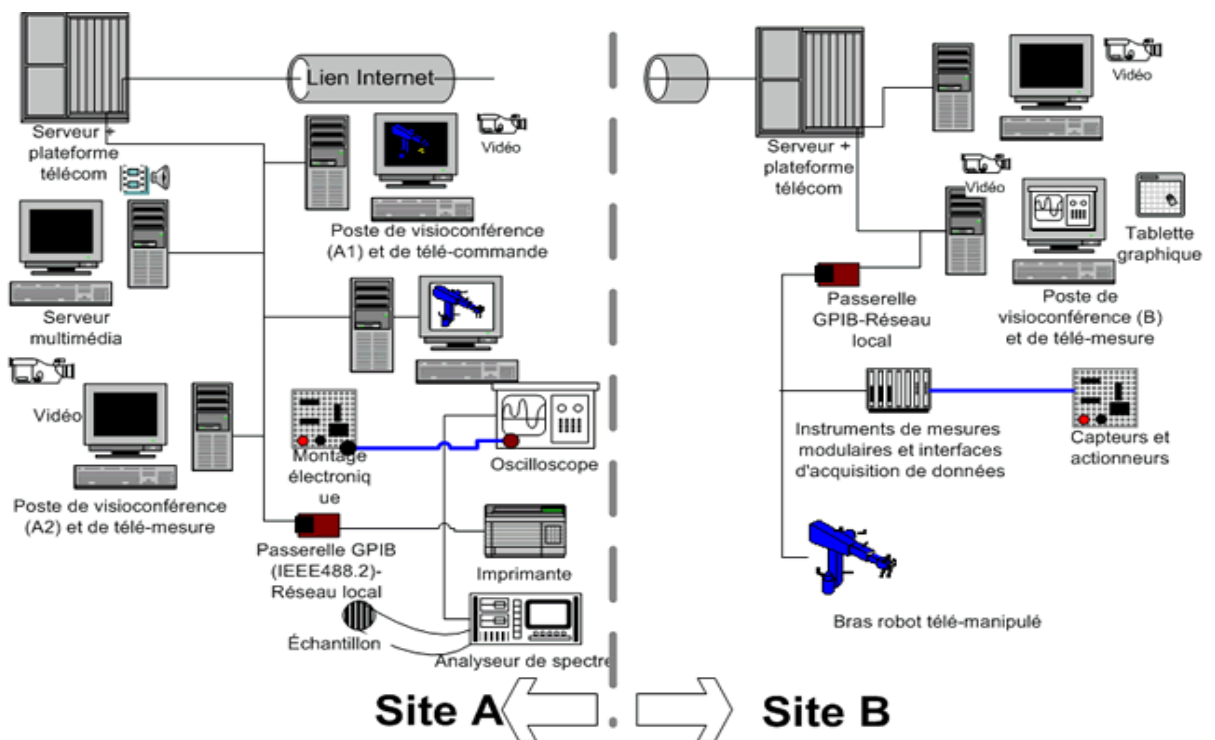


Figure II.8. : Laboratoire en ligne

4.3. Exemple de laboratoire électronique (e-lab) :

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

Le projet **LVEST** expérimente la diffusion en temps réel de séquences vidéo filmées ou simulées, la simulation interactive d'expériences, la télémanipulation et la télémessure. De plus, le projet offrait des outils d'acquisition et d'analyse de données ainsi que des outils de

communication entre apprenants et avec le formateur, ce qui permet l'échange de données expérimentales et la communication synchrone en modes texte, audio et vidéo.

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

- **Le laboratoire virtuel de génie électrique** : est un environnement de simulation tenant compte des technologies novatrices émergentes relatives au développement des produits commerciaux en instrumentation
L'intégration de ces produits dans un contexte de formation à distance posait des défis technologiques de taille, dont le choix du langage décrivant des dispositifs interconnectés, leurs adresses et leurs fonctions.
- **Le laboratoire de Génie Informatique** est un environnement de simulation visant l'apprentissage de la démarche expérimentale et des concepts scientifiques à l'œuvre dans les expériences. Le laboratoire de réseautique comprend des expériences de divers types réalisables individuellement ou en équipe et une interface de navigation entre les domaines, groupes d'expériences. Expériences et espaces de ceux-ci, offrant de plus un accès vers les outils d'analyse des données et les services de communication éventuellement offerts par la plateforme.

4.3.2. Le projet VITELS (laboratoire virtuel de télécommunications pour la Suisse) :

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

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

Il est possible, pendant l'activité, de retourner à tout moment à une étape précédente. Le type d'installation peut être prédéfini selon le niveau de connaissance de GNU/Linux de l'utilisateur. Le modèle de laboratoire en ligne mis au point reproduit l'environnement DrakX

le plus fidèlement possible pour permettre aux apprenants et aux formateurs de se retrouver facilement au moment d'une vraie installation sur un poste de travail.

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



II.9. Partition du disque dur d'une machine

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

4.3.3. Le projet eMersion :

L'École Polytechnique Fédérale de Lausanne (EPFL), par l'intermédiaire du projet eMersion du Département de génie mécanique, fait des recherches dans le domaine des laboratoires en ligne. Ce projet intervient dans le domaine du génie mécanique, dans des modules de formation flexible en ateliers de simulation et en laboratoires d'expérimentation. Le projet eMersion a pour but de réintroduire l'apprentissage par l'erreur dans la formation des ingénieurs en développant et en flexibilisant les possibilités de simulation en réalité virtuelle et d'expérimentation en laboratoire. Un corollaire est de favoriser la rénovation du

schéma actuel d'études grâce à la mise à disposition de moyens alternatifs de conduite d'activités pratiques, essentiels tant sur le site qu'à domicile.

5. Conclusion :

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

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

*Chapitre 3 : Les accès concurrentiels dans une
plate forme téléTP*

1. Introduction

Dans le contexte d'enseignement à distance, les apprenants sont amenés à réaliser conjointement un objectif en utilisant un dispositif commun. L'environnement produit est alors tantôt collaboratif, tantôt concurrentiel.

Les accès concurrents à des données partagées, ou même plus généralement à des ressources uniques et chères, peuvent produire des incohérences de données et d'accès.

Dès lors, sur quels critères s'appuyer pour désigner l'utilisateur le plus prioritaire qui aura l'accès au dispositif, tandis que les autres seront mis en attente ? Comment départager deux utilisateurs qui demandent l'accès exclusif à une ressource ?

Ce sont les questions aux quels nous répondrons dans ce chapitre.

2. La collaboration et la concurrence dans une plateforme TéléTP

Une plate-forme TéléTp peut être vue comme un environnement concurrentiel ou collaboratif, qui permet de gérer et de donner accès à un ensemble d'activités et de ressources pédagogique.

Lorsque plusieurs utilisateurs (l'enseignant, le concepteur informatique, le tuteur, l'apprenant, l'administrateur) accèdent à la ressource d'une plateforme en même temps, on parle d'accès concurrentiel.

La collaboration a lieu lorsque plusieurs auteurs ou bien enseignant apportent des commentaires et des modifications à un cours ou bien à un document partagé, que ce soit en mode asynchrone ou simultanément (mode synchrone), alors il est utile de :

- suivre les modifications et les commentaires au sein du document et du cours en repérant leurs l'auteur;
- gérer les versions successives du document ;
- gérer les conflits d'accès ou accès concurrent au document : cela consiste à verrouiller tout ou une partie du document qu'un auteur est en train d'éditer pour éviter qu'un autre auteur la modifie simultanément. On distingue deux cas d'édition collaborative :

L'édition en ligne : le document ou le cour est intégré à une application en ligne permettant de le modifier directement. Le fonctionnement peut être synchrone ou

asynchrone.

L'édition hors ligne : le document est souvent sur un espace de stockage partagé. Lorsqu'un auteur veut l'éditer, il le télécharge pour le modifier. Ce qui est possible que si un autre auteur n'est pas en train de l'éditer, ce qui implique un fonctionnement en mode asynchrone.

2.1. Gérer la collaboration

Les besoins de gérer la collaboration au sein des télé-TPs sont très différents d'une séance à l'autre, et dépendent aussi de la pédagogie mise en place par le tuteur enseignant.

C'est alors que se dressent les limites des modèles actuels. Un algorithme gérant l'accès à une ressource ne change pas au cours du temps, ce sont les mêmes conditions d'accès quel que soit le contexte d'exécution. Dans ces conditions, il conviendrait de fournir autant d'algorithmes décrivant une administration qu'il peut exister de modalités de partage de ressources.

2.2. Gérer la priorité des utilisateurs

Pour apporter une nouvelle vision du partage d'une ressource, il faut mettre en œuvre une approche efficace qui consiste à prendre en compte les éléments du contexte dans le but de prendre une décision sur la priorité entre les utilisateurs faisant la demande d'accès au dispositif.

3. La synchronisation et les téléTP

Dans une plateforme TéléTP il est important de parler des moyens de synchronisation pour que les utilisateurs puissent se partager d'une manière structurée et équitable un dispositif ou des ressources unique à des couts élevés.

Pour bien comprendre le concept de la concurrence des utilisateurs a des ressources critiques on les considère comme des **processus** système et les TP comme des **sections critiques** du système, nous allons présenter et détailler dans ce qui suit les différents moyens de synchronisation afin d'éviter les conflits d'accès aux données partagées.

4. La concurrence des tâches

Avant de parler de tâche nous devons d'abord parler de processus et de threads.

4.1. Les processus

Un processus est un programme qui s'exécute et qui possède son compteur ordinal, ses registres et ses variables. Conceptuellement, chaque processus a son propre processeur virtuel. En réalité, le vrai processeur commute entre plusieurs processus. » (ndlr Le Tannenbaum système)

C'est là qu'intervient la notion d'intercommunication entre les processus et qui induit en même temps tout un flot d'incertitude et d'inquiétude quant à la réalisation et à la mise en pratique de ce genre de concept.

Dans le contexte d'exécution d'un système d'exploitation multitâche, plusieurs processus sont amenés à travailler simultanément, et parfois même sur des données communes.

De nombreux systèmes d'exploitation modernes offrent la possibilité d'associer à un même processus plusieurs chemins d'exécution (multithreading, multiflot d'exécution).

4.2. Les threads

Un thread est un fil d'instruction (chemin d'exécution) à l'intérieur d'un processus. Contrairement au processus, les threads d'un même processus partagent le même espace d'adressage, le même environnement (par exemple les mêmes variables d'environnement).

Chaque thread a :

- un identificateur unique
- une pile d'exécution
- des registres (un compteur ordinal)
- un état...

Un thread peut passer par différents états, tout au long de son cycle de vie. Le diagramme suivant illustre ces différents stades ainsi que les différentes transitions possibles

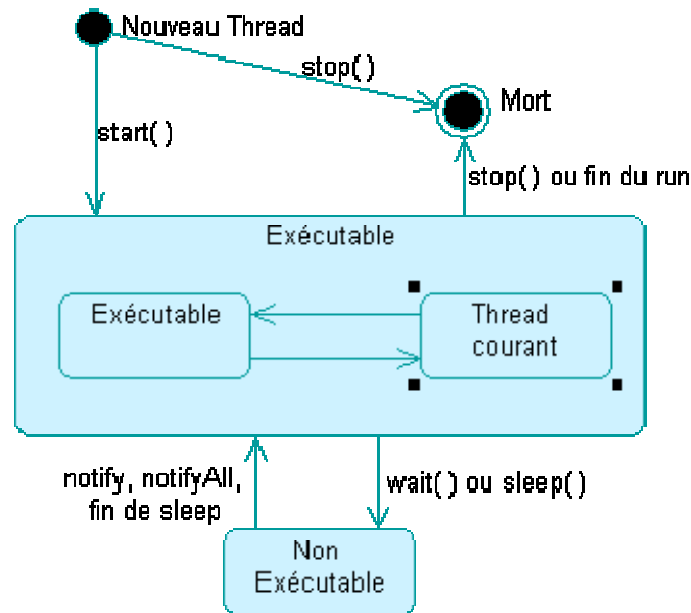


Figure III.1 cycle de vie d'un thread

Comme on le voit sur le schéma, un thread passe par les états : initial, transitions entre exécutable et non exécutable, et fin (mort).

A sa création, un thread est par défaut dans son état initial. Il faut invoquer la méthode `start()` pour lui permettre de passer dans un état exécutable. Le thread passe à l'état mort soit à la fin de l'exécution de la méthode `run()`, soit à l'exécution de `stop()`. Un thread à l'état mort ne peut plus redémarrer.

L'état exécutable du thread lui permet de recevoir le processus et de s'exécuter un certain temps (quelques millisecondes) puis de laisser la main à un autre thread exécutable.

- Dans les systèmes préemptifs (systèmes actuels) : Le thread n'a aucune action à faire pour laisser la main à un autre thread.
- Dans les systèmes non préemptifs (plus rare actuellement) : le thread doit appeler la méthode `yield()` pour céder la main à un autre thread.

Un thread passe d'un état exécutable à un état non exécutable sous l'invocation de l'une de ces méthodes :

- `Thread.sleep(long durée)`, qui suspend le thread durant quelques instants, où

- les méthodes *wait* de la classe *java.lang.Object*. (cf. synchronisation)

Le passage à un état exécutable à nouveau, à partir d'un état non exécutable se fait soit :

- après l'écoulement du temps indiqué dans la méthode *sleep*
- soit après la réception d'une notification d'un autre thread (*notify()*, *notifyAll()*).

Contrairement aux processus, les threads d'un même processus partagent le même espace d'adressage, le même environnement. Ainsi 2 threads d'un même processus communiquent beaucoup plus facilement que 2 processus. (Voir plus de détail dans l'annexe B)

4.3. Les taches

Nous parlons de tache par ce qu'elles représentent de manière concrète les processus et leur mise en place dans un système.

Qu'est ce qu'une tache ?

Une tache est une unité de programme lancé par un thread, qui s'exécute en parallèle avec d'autres taches lancées par d'autres threads.

Vu de cette façon on pourrait imaginer logiquement une solution constituée de nombreuses taches indépendantes. Mais physiquement, les taches peuvent s'exécuter sur des systèmes multi-ordinateurs, Multiprocesseurs ou par exécution entrelacée sur processeur unique.

5. Les différents types d'intercommunications et de synchronisation

5.1. Les accès concurrents

Considérons deux processus (ou plus) qui s'exécutent concurremment dans un environnement multitâche, et qui sont chargés de la réservation de places d'avion.

Le système d'exploitation leur accorde à tour de rôle un temps d'exécution.

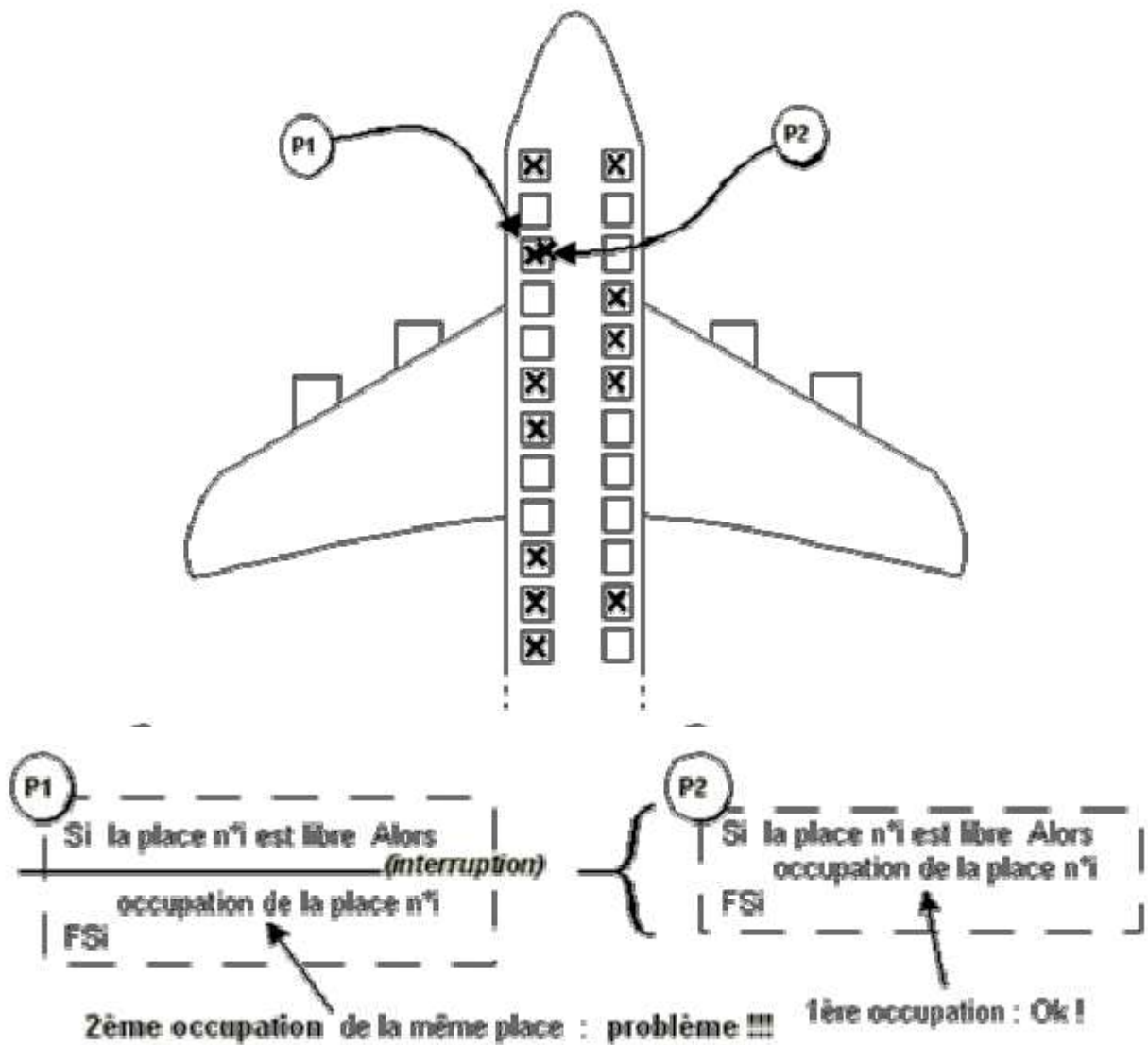


Figure III.2 : Conflit d'accès aux ressources non partageables

Imaginons que le processus P1, soit interrompu juste après avoir testé qu'une place donnée est libre et avant d'avoir pu la marquer comme occupée. Dans ce cas, le système peut alors passer la main au processus P2 qui désire réserver la même place et le fait intégralement. Lorsque P1 peut à nouveau s'exécuter, il n'est pas informé que P2 à occuper cette même place, et finit de la réserver lui aussi. Au final, la même place est réservée deux fois !

Le problème est que lorsque le processus P2 entre en action, les variables communes ne reflètent pas exactement la valeur qu'elle devrait avoir, car P1 n'a pu finir son traitement correctement.

D'une manière générale, on ne peut déterminer le temps d'exécution accordé à un processus. En effet, la vitesse relative de deux processus est imprévisible puisque cela dépend de la fréquence des interruptions de chaque processus, ainsi que de la durée de travail et de la fréquence d'attribution en temps d'exécution aux processus. Nous disons que les processus se déroulent de manière asynchrone les uns par rapport aux autres.

Cependant, il est nécessaire de pouvoir trouver un moyen de synchroniser les processus afin d'éviter les accès concurrents aux données partagées.

5.2. Sections critiques

Une séquence d'instructions manipulant des données partagées par plusieurs processus est appelée Section Critique, car l'entrelacement des exécutions de ces instructions par différents processus peut conduire à des incohérences. C'est la zone où se produit les conflits d'accès.

C'est en travaillant sur la section critique, elle même, que l'on doit éviter les problèmes dans les mises en œuvre de partage de ressources (fichiers, mémoires, etc.). Il faut s'assurer que plusieurs processus ne puissent accéder à une section critique en même temps. Mais dans ce cas on ralentira et on limitera le travail de deux processus parallèles partageant une ressource.

Dans ce cas quatre conditions sont nécessaires :

1. deux processus ne peuvent être en même temps en section critique
2. Aucune hypothèse ne doit être faite sur les vitesses relatives des processus et sur le nombre de processeurs.
3. Aucun processus suspendu en dehors d'une section critique ne doit bloquer les autres processus.
4. Aucun processus ne doit attendre trop longtemps avant d'entrer en section critique.

(ndlr Tannenbaum Système)

Pour résoudre ce problème de contrôle d'occupation de la zone critique, par un et un seul processus à la fois, il existe l'exclusion mutuelle.

5.3. Exclusion mutuelle par attente active (Mutex)

Le problème de l'exclusion mutuelle peut être simplement résolu à l'aide d'un sémaphore dont la valeur initiale est fixée à 1. Un tel sémaphore est couramment appelé sémaphore d'exclusion mutuelle (mutex).

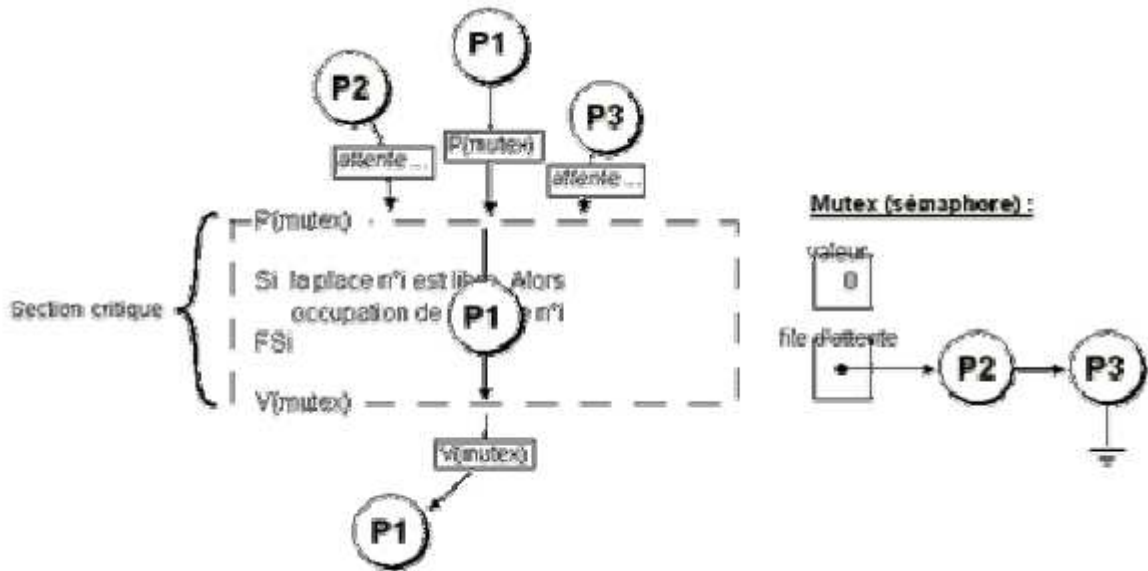


Figure III.3 : Utilisation des sémaphores pour l'exclusion mutuelle

Ainsi pour garantir le respect de la section critique, il suffit de l'encadrer entre deux appels à respectivement P et V sur ce sémaphore.

Le premier processus désirant entrer en section critique (on suppose que celle-ci est libre), effectue un appel à P, et prend donc le seul ticket disponible. La valeur du sémaphore passe à 0.

Dès lors, tout autre processus désirant entrer en section critique, se voit mis en attente sur le sémaphore (lors de son appel à P), car la valeur de ce dernier est nul.

Enfin, lorsque l'exécution de la section critique est terminée, le processus l'ayant occupé effectue son appel à V afin de rendre son ticket : la valeur du sémaphore redevient positive, en l'occurrence égale à 1, et le (ou les) processus éventuellement bloqués peuvent accéder, à tour de rôle et selon la méthode précédemment décrite, à la section critique.

La section critique étant protégée par un sémaphore ne disposant que d'un ticket : un seul processus peut exécuter l'opération P avant celle-ci ne devienne bloquante, et seul l'opération V exécutée après la fin de la section critique autorise à nouveau l'exécution de P. Donc l'exclusion mutuelle est donc bien assuré.

Précision : pour un fonctionnement correct, il doit être garanti, comme dans la définitions sémantique, que les opérations P et V soient indivisibles, c'est-à-dire non interruptibles au profit d'un autre processus.

5.4. Les Sémaphores

La plus importante contribution au problème de la synchronisation entre processus fut l'introduction par Dijkstra, en 1965, du concept des Sémaphores.

Conceptuellement, un sémaphore joue le rôle d'un distributeur de tickets, qui, initialement dispose d'un certain nombre de tickets, éventuellement nul.

Un processus utilisateur du sémaphore demande un ticket en invoquant une opération particulière appelée P : si au moins un ticket est disponible, le processus appelant le prend et poursuit son exécution ; dans le cas contraire, le demandeur est enregistré dans une file d'attente et est bloqué dans l'attente de l'obtention d'un ticket. Autrement dit, l'opération P n'est exécutable par le processus appelant que s'il existe un ticket libre, elle peut se retrouver bloquante dans le cas contraire.

Grâce à une opération inverse, appelée V, un processus dépose son ticket dans le distributeur. Si des processus sont en attente dans la file du sémaphore, le premier d'entre eux est débloqué et obtient son ticket.

Plus précisément, un sémaphore est une variable entière non négative à laquelle, sauf pour l'initialisation, on accède seulement à travers deux opérations standards atomiques (non interruptibles) : P et V.

- P, de l'hollandais *proberen* (tester), aussi appelé parfois *wait*, consiste à décroître la valeur du sémaphore d'une unité à condition que le résultat ne devienne pas négatif. En effet, dans le cas où un processus effectue P et que la valeur du sémaphore est de zéro, alors ce dernier se retrouve gelé jusqu'à ce que cette valeur redevienne supérieur strictement à zéro et qu'il puisse alors la décrémenter.

Algorithme de P :

```
TQ sémaphore <= 0 Faire
    rien (nop ...)
FTQ
Décrémenter sémaphore
```

- V, pour *proberen* (tester), aussi appelé parfois *signal*, consiste simplement à accroître la valeur du sémaphore d'une unité.

Algorithme de V :

Incrémenter sémaphore

Ce mécanisme permet donc de limiter l'exécution d'une portion de code (et donc l'accès à une ressource quelconque), encadrée par des appels respectifs à P et à V, à un nombre fixé de processus. Nous allons voir par la suite, que cela permet de réaliser l'exclusion mutuelle nécessaire à la prévention des conflits d'accès aux données non partageables.

5.5. Les moniteurs

Le type moniteur est une autre structure de synchronisation de haut niveau introduite par Hoare en 1974. Un moniteur peut être assimilé à une sorte d'objet de synchronisation, contenant les données communes, et caractérisé par un ensemble d'opérations définies par le programmeur et seules autorisées à accéder (aussi bien en lecture qu'en écriture) à ces données.

Les moniteurs assurent l'exclusion mutuelle sur leurs opérations.

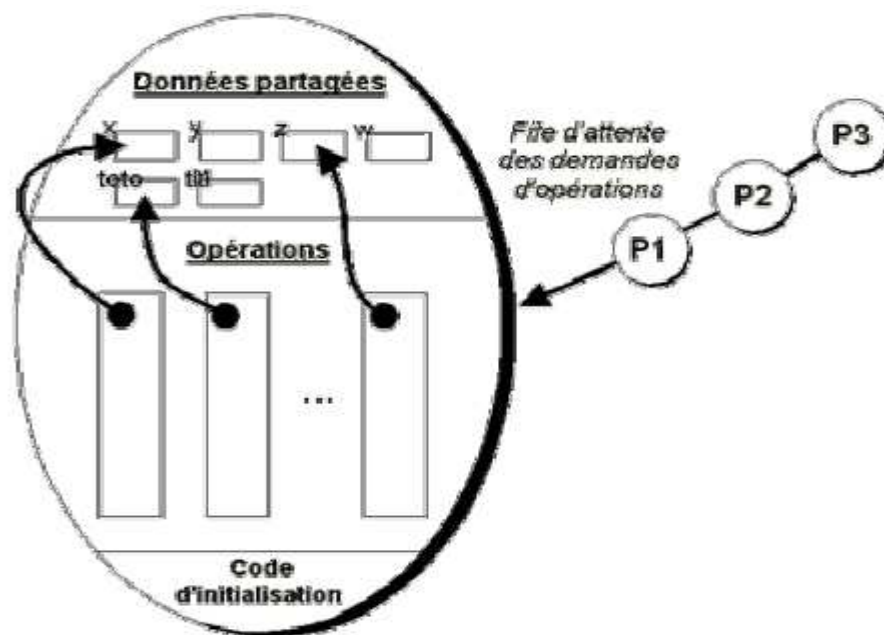


Figure III.4 : Fonctionnement d'un moniteur

Les processus, lorsqu'ils désirent effectuer des traitements sur les données communes, expédient une demande au moniteur en appelant l'opération correspondante.

La structure du moniteur assure qu'un seul processus à la fois puisse être actif dans celui-ci, les autres étant suspendus en entrée, par le moniteur, dans une file d'attente.

Ainsi, les moniteurs permettent de décharger le programmeur de la contrainte de la gestion des sémaphores, par exemple pour ce qui concerne l'exclusion mutuelle, en déplaçant cette tâche au niveau du système, et en l'occurrence au niveau du moniteur dédié.

Ils permettent aux programmeurs à déclarer explicitement les ressources et données partagées et à les placer dans les moniteurs correspondants.

5.6. Les autres possibilités

- les primitives Sleep et Wake up : fonctionnent grâce à deux primitives : l'une, Sleep, suspend le processus appelant en attendant un signe de réveil de l'autre Wake up qui le réactive.
- l'échange des messages fonctionnant grâce à deux primitives SEND et RECEIVE fonctionnant comme des appels systèmes
- les compteurs d'événements qui fonctionnent par trois états du compteur d'événement ce qui peut provoquer sa lecture, son avance ou son attente.

6. Conclusion :

Comme nous avons pu le voir l'intercommunication des processus amène son flot d'incertitude et d'inconvénient mais également de solutions pour contourner ces difficultés. Ainsi l'utilisation des processus et la programmation de tâches parallèles amènent de grandes possibilités de gestion parallèle s'appliquant dans notre vie de tous les jours. L'exemple des réservations de place d'avion en est un.

La solution des problèmes d'intercommunication de processus vient du fait que l'on arrive à trouver le meilleur compromis entre la solution la plus efficace et la solution la plus rapide qui sont rarement les mêmes.

Chapitre 4 : Analyse et conception

1. Introduction :

Avant toute réalisation d'une application informatique ; il convient de suivre une démarche méthodologique et rigoureuse pour planifier et concevoir l'application, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Afin de faciliter l'accès de plusieurs utilisateurs à notre plateforme de travaux pratiques, nous allons présenter dans ce chapitre les moyens qu'on a utilisés pour bien orchestrer et partager les TPS entre utilisateurs, nous commencerons par la description de notre plateforme , ensuite nous présenterons la solution apportée pour la gestion des accès concurrentiels avec les sémaphores.

2. Objectif du projet :

L'objectif visé dans la réalisation de notre application est celui d'assurer un enseignement de qualité pour tout apprenant souhaitant approfondir ses connaissances dans plusieurs domaines tel que la chimie, la physique...etc. tout cela par une simple navigation dans l'application.

Dans ce travail, on a adopté un système d'enseignement à distance, et cela en proposant plusieurs cours, où chaque cours est constitué d'un ensemble de chapitres et est suivi d'une suite d'exercices et de tests d'évaluations, et aussi un module de gestion des accès simultané à un tp, tel que plusieurs apprenant peuvent accéder à l'interface TP's mais un seul et unique apprenant peut accéder à un tp à un moment donné.

3. La démarche d'élaboration de notre projet :

Le développement d'applications informatiques n'est pas une chose aisée. Un processus rigoureux permet de mener à terme des projets d'une façon fiable et renouvelable.

La conception d'applications web se distingue de la conception d'autres systèmes par deux activités majeurs : la répartition des objets sur le client ou le serveur et la définition de l'interface utilisateur sous forme de pages web.

3.1. Identification des acteurs et leurs besoins fonctionnels :

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

- **Administrateur :**

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

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

- **Formateur :**

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

- Assurant le suivi du cours au prés des apprenants ;
- Communiquant avec ses apprenants via la messagerie interne, le forum ;
- Evaluant ses apprenants par des exercices et des tests d'évaluation.

- **Apprenant :**

L'apprenant est toute personne qui suit un cours dans son propre espace réservé, il peut aussi :

- Consulter des cours et des exercices ;
- Communiquer avec les autres apprenants et son formateur via les outils collaboratifs
- (Forums, messagerie interne) ;
- Faire d'exercices et des évaluations ;
- Accéder a l'espace des TPs.

- **Auteur :**

L'auteur représente l'acteur chargé de gérer le contenu pédagogique, il assure un ensemble d'actions tels que :

- Dépôt de cours et d'exercices;

- Mise à jour des cours et des exercices ;
- Proposition des évaluations.

• **Visiteur :**

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

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

- Se préinscrire en tant qu'apprenant ;
- Se préinscrire en tant que formateur ;
- Se préinscrire en tant que auteur ;
- Contacter l'administrateur pour d'éventuelles informations ;
- Naviguer dans le site pour avoir une idée des possibilités des formations.

3.2. Diagrammes représentatifs :

3.2.1. Le diagramme de contexte de l'application :

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

La définition des besoins fonctionnels des acteurs présentée dans le paragraphe (3.1) (nous permet de dégager le diagramme de contexte suivant :

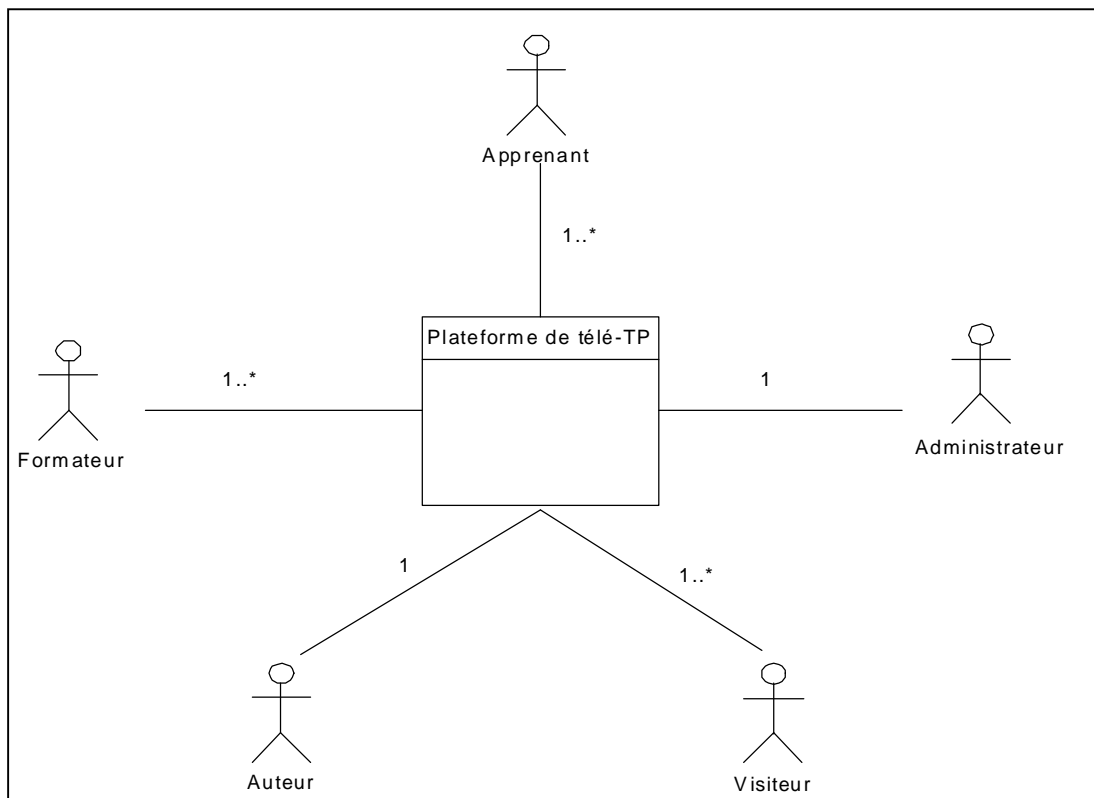


Figure IV.1 : Diagramme de contexte de l'application.

3.2.2. Diagrammes des cas d'utilisation :

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

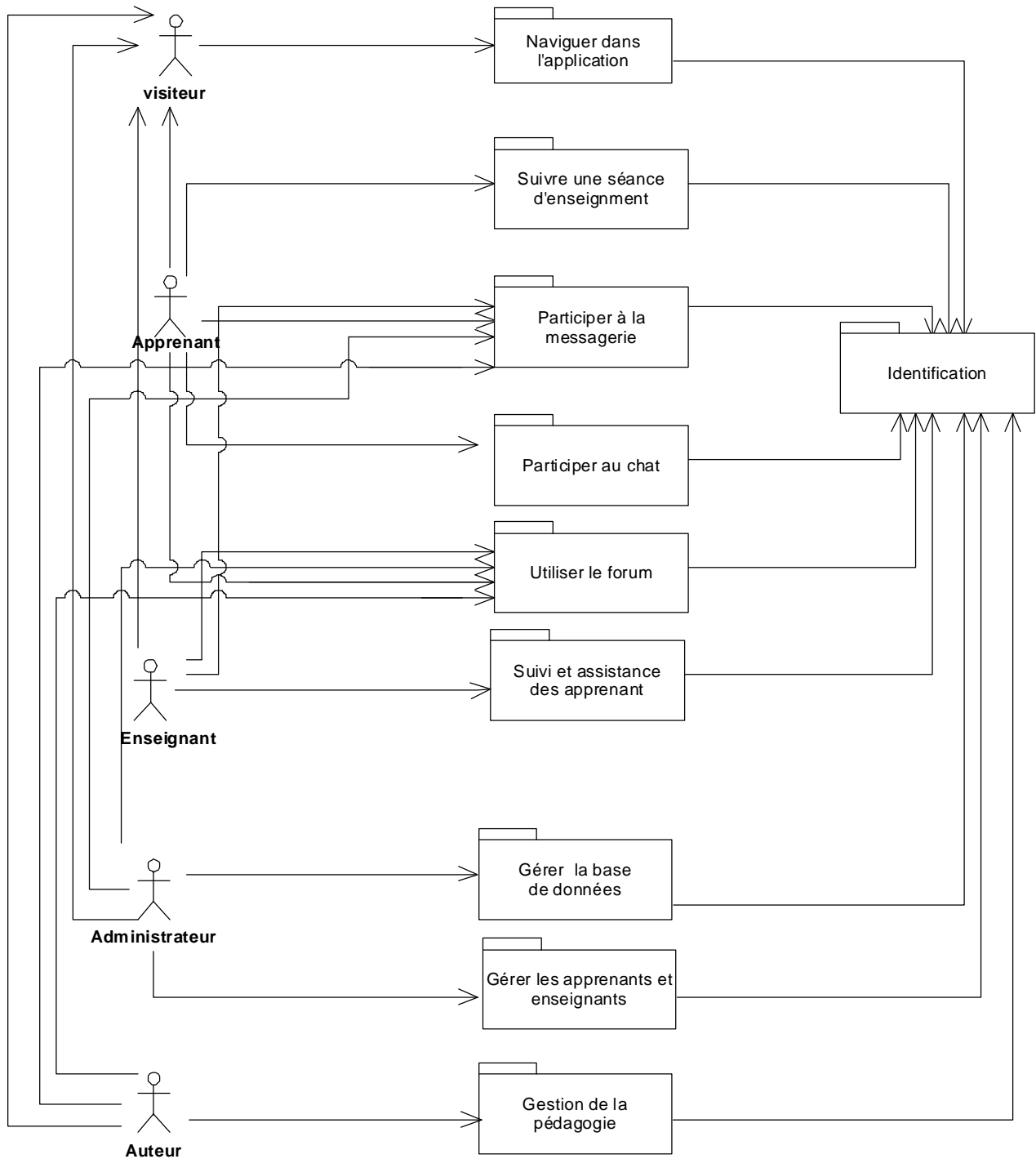


Figure IV.2 : Structuration des activités en package.

3.2.3. Détermination des diagrammes de cas d'utilisations :

Etant donné que les acteurs et les packages de leurs activités sont identifiés, nous sommes en mesure de modéliser les cas d'utilisation de ces acteurs, qui sont des détails des packages définis dans la figure IV-2. Un cas d'utilisation décrit ce que l'utilisateur veut fondamentalement faire avec le système. Il s'agit d'une représentation macroscopique des interactions entre acteurs et le système fondée sur une sémantique "faible" qui rend ce modèle aisément compréhensible par les utilisateurs.

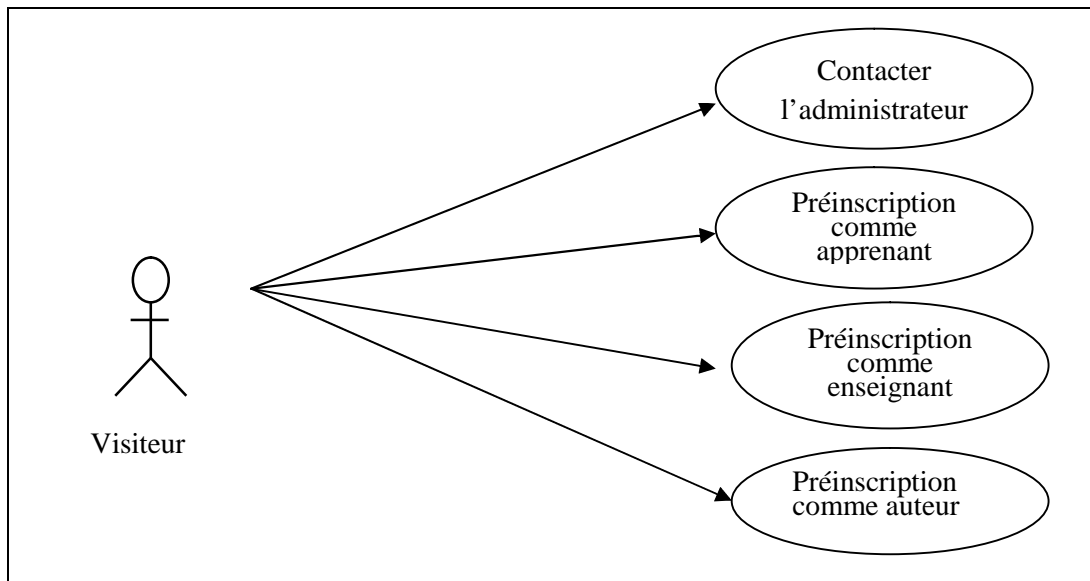


Figure IV.3 : Diagramme de cas d'utilisation pour le package « naviguer dans l'application »

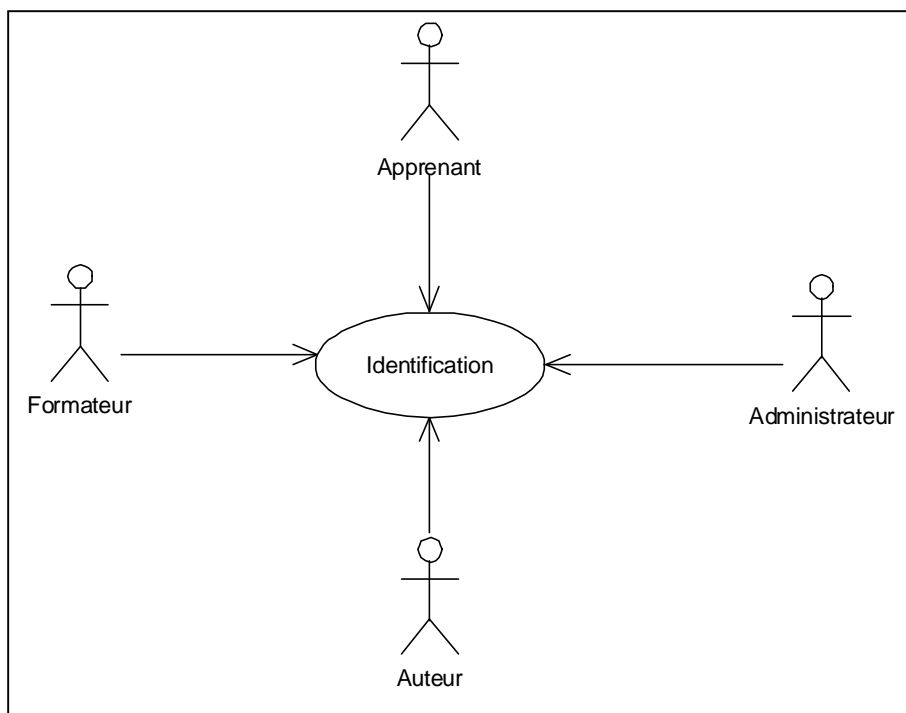


Figure IV.4 : Diagramme de cas d'utilisation pour le package « identification ».

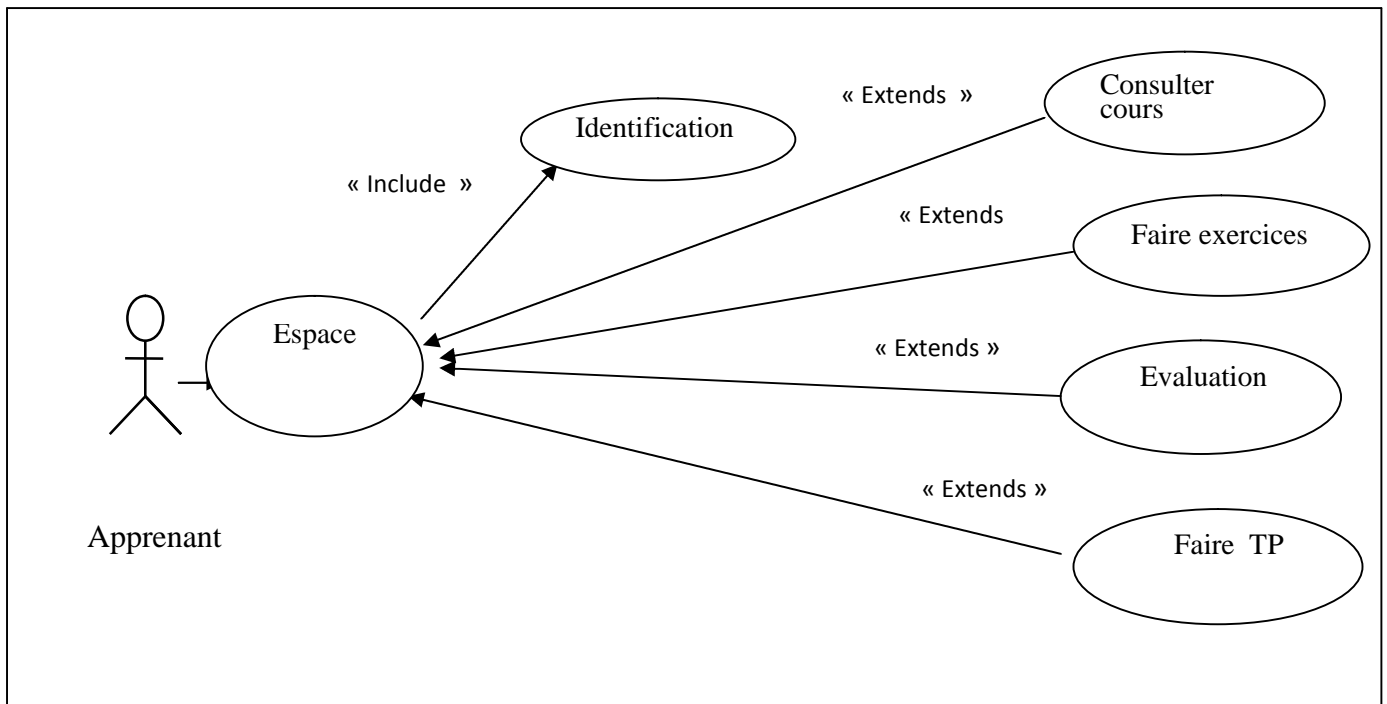


Figure IV.5 : Diagramme de cas d'utilisation pour le paquetage « Suivre une séance d'enseignement »

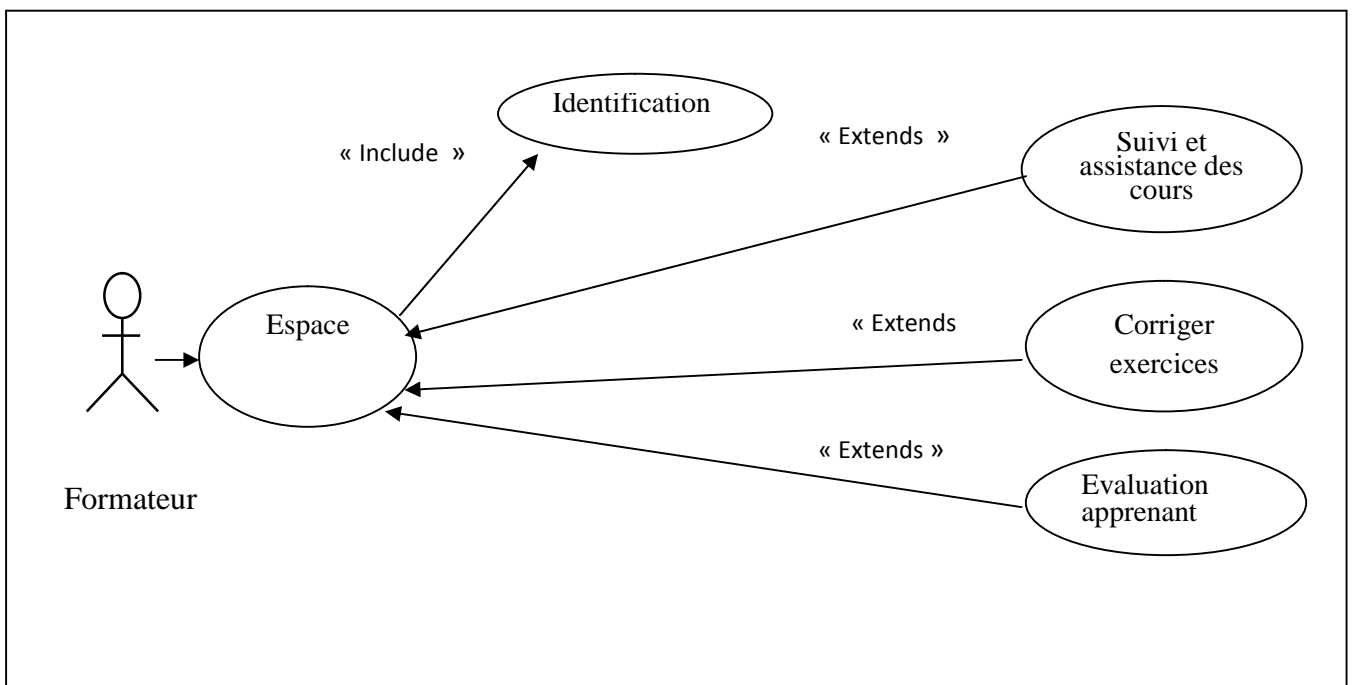


Figure IV.6 : Diagramme de cas d'utilisations pour le paquetage «suivi et assistance des apprenants».

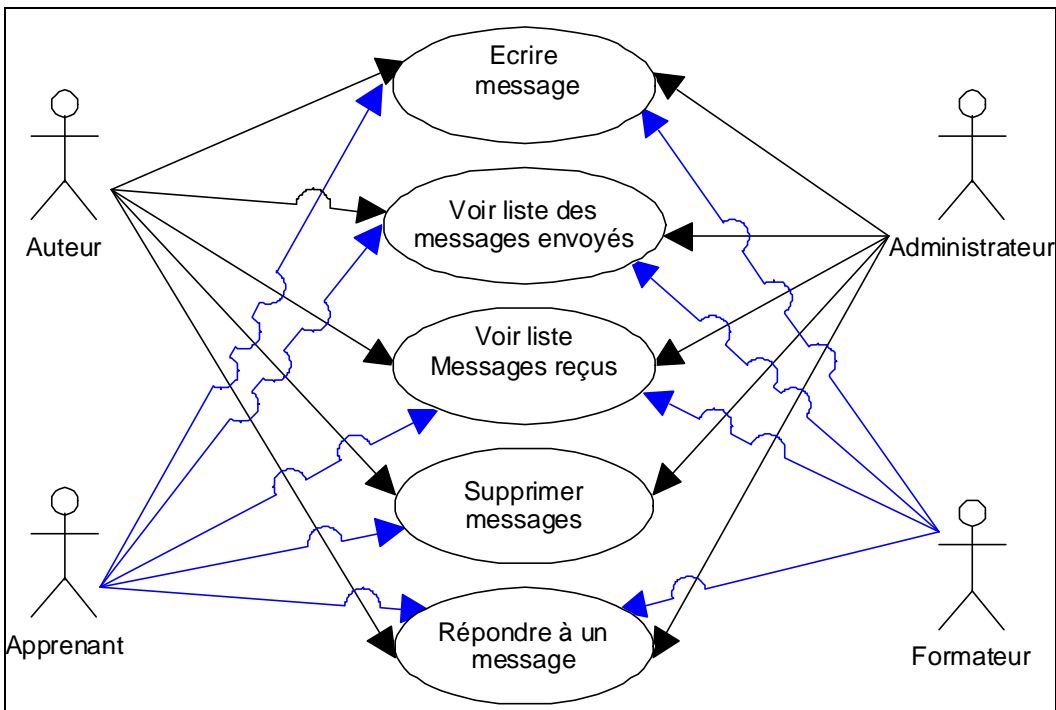


Figure IV.7: Diagramme de cas d'utilisations pour le paquetage « participer à la messagerie».

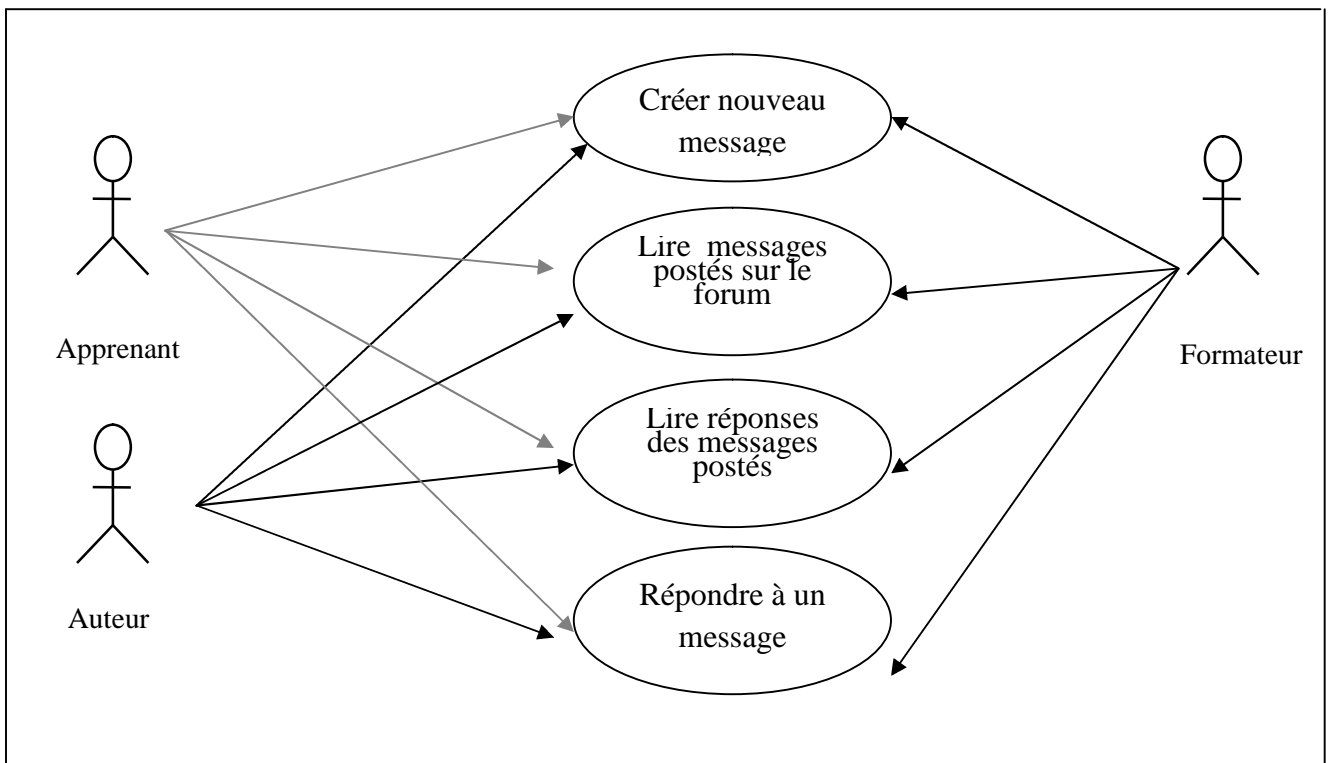


Figure IV.8 : Diagramme de cas d'utilisation pour le paquetage « utiliser le forum ».

3.2.4. Description textuelle des cas d'utilisation :

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

- **Pré inscription :**

Titre : Pré inscription.

Acteurs : Visiteur.

- **Scénario nominal 1 :**

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

- **Enchaînement alternatif 1 :**

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

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

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

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

- **Enchaînement alternatif 2 :**

B : *S'il y a pas d'erreurs dans le formulaire*

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

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

➤ **Enchaînement alternatif 2.1 :**

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

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

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

➤ **Enchaînement alternatif 2.2 :**

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

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

- **Identification**

Titre : Identification

Acteurs : Apprenant, enseignant, auteur ou administrateur

➤ **Scénario nominal 2 :**

1. L'utilisateur atteint la page d'identification ;
2. Le système répond en affichant le formulaire d'identification pour accéder à son propre espace;

➤ **Enchaînement alternatif 1 :**

A : *L'utilisateur possède un compte :*

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

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

➤ **Enchaînement alternatif 1.1 :**

A.1 : *Login et mot de passe corrects.*

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

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

➤ **Enchaînement alternatif 1.2 :**

A.2 : *Login ou mot de passe incorrect*

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

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

➤ **Enchaînement alternatif 2 :**

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

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

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

- **Suivre une séance d'enseignement**

Titre : Suivre une séance d'enseignement

Acteur : Apprenant.

➤ **Scénario nominal 3 :**

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

➤ **Enchaînement alternatif 1 :**

A : *l'apprenant décide de consulter le cours.*

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

3. L'apprenant clique sur le lien « cours » ;
4. Le système charge depuis la base de données les cours disponible;
5. L'apprenant consulte son cours.

➤ **Enchaînement alternatif 2 :**

B. : *l'apprenant décide de faire un exercice.*

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

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

➤ **Enchaînement alternatif 3 :**

C : *l'apprenant décide de consulter les résultats de l'évaluation.*

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

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

➤ **Enchaînement alternatif 4 :**

D : *l'apprenant décide de faire le TP.*

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

3. L'apprenant clique sur le lien «TP» ;
4. Le système répond en chargeant la page d'accès au TP qui contient un lien vers l'espace de travail ;
5. L'apprenant clique sur le lien « accéder a l'espace de travail » ;
6. Le système lui affiche la fenêtre de l'espace de travail.

• **Gestion des cours et exercices**

Titre : Gestion du contenu pédagogique.

Acteur : Auteur.

➤ **Scénario nominal 4 :**

1. Après identification, l'auteur atteint sa page d'accueil ;
2. Le système répond en affichant la page d'accueil. Sur la page figure le lien hypertexte «cours, exercices, évaluation» ;
3. L'auteur clique sur le lien « cours » ;

4. Le système interroge la base de données et retourne une page contenant la liste de tout les cours ;

➤ **Enchaînement alternatif 1 :**

A : l'auteur décide de consulter un cours.

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

5. L'auteur clique sur le lien « consultation » ;

6. Le système interroge la base de données et retourne une page contenant une liste de tous les chapitres constituant le cours à consulter.

7. L'auteur sélectionne un chapitre ;

8. Le système lance le téléchargement du cours en transférant ses différents chapitres au coté client (environnement de l'auteur) ;

➤ **Enchaînement alternatif 2 :**

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

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

5. L'auteur clique sur le bouton « ajouter » ;

6. Le système interroge la base de données et retourne une page contenant une liste de tous les cours ;

7. L'auteur « sélectionne » un cours le système affiche un formulaire proposant le nom du chapitre et son contenu ;

8. L'auteur clique sur le bouton « envoyer » ;

8. Le système interroge la base de données et retourne une page contenant une liste de tous les chapitres y compris celui ajouté et propose d'appuyer sur le bouton « valider ».

➤ **Enchaînement alternatif 2 :**

A : l'auteur décide de supprimer un cours.

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

5. L'auteur clique sur le lien « supprimer » ;

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

- **Gérer les préinscriptions des apprenants**

Titre : gérer les prés inscriptions des apprenants.

Acteur : administrateur.

➤ **Scénario nominal 5:**

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

➤ **Enchaînement alternatif 1 :**

A : *si le profil du candidat est correct.*

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

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

Le scénario nominal démarre au point « 4. ».

3.2.5. Réalisation des cas d'utilisation :

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

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

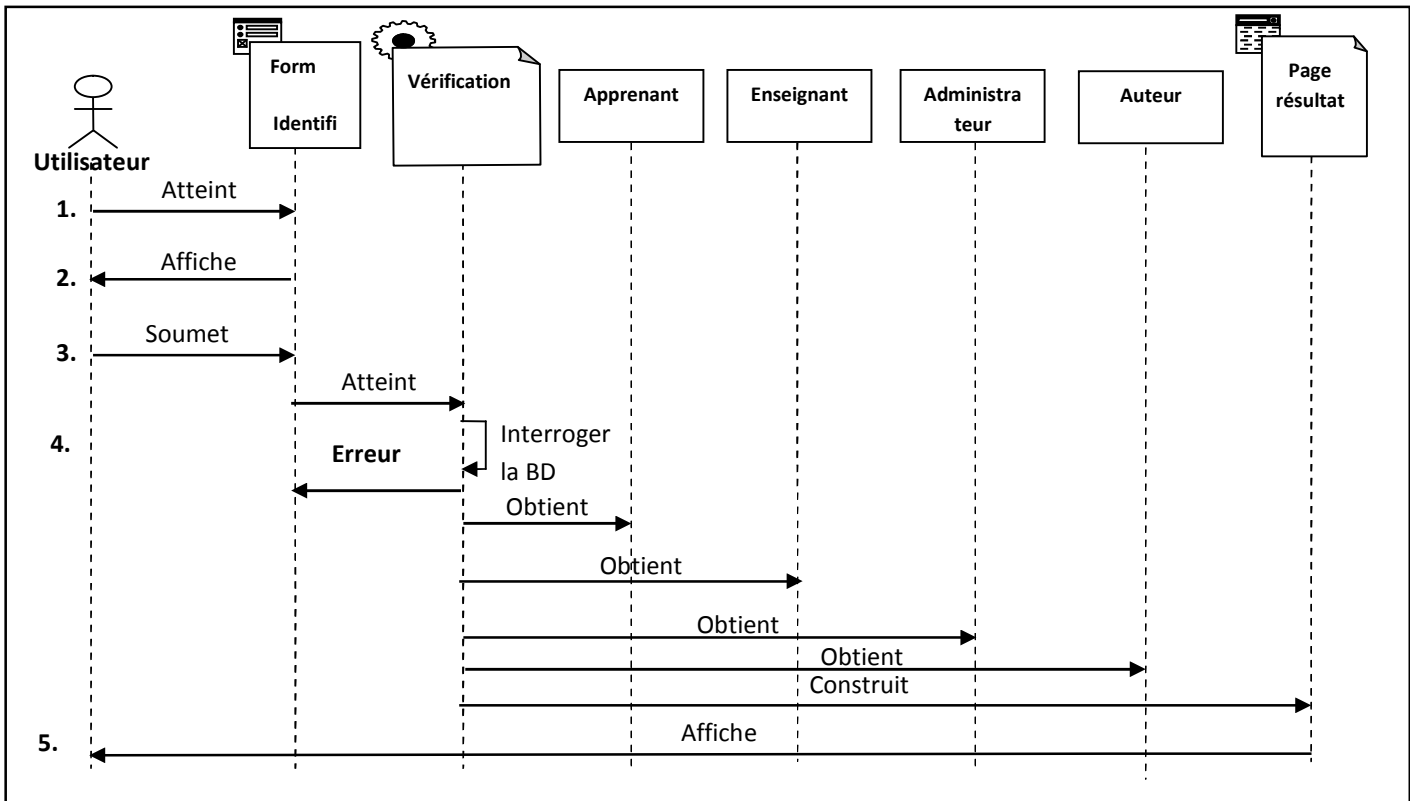


Figure IV.9: Diagramme de séquence de réalisation de cas d'utilisation « identification »

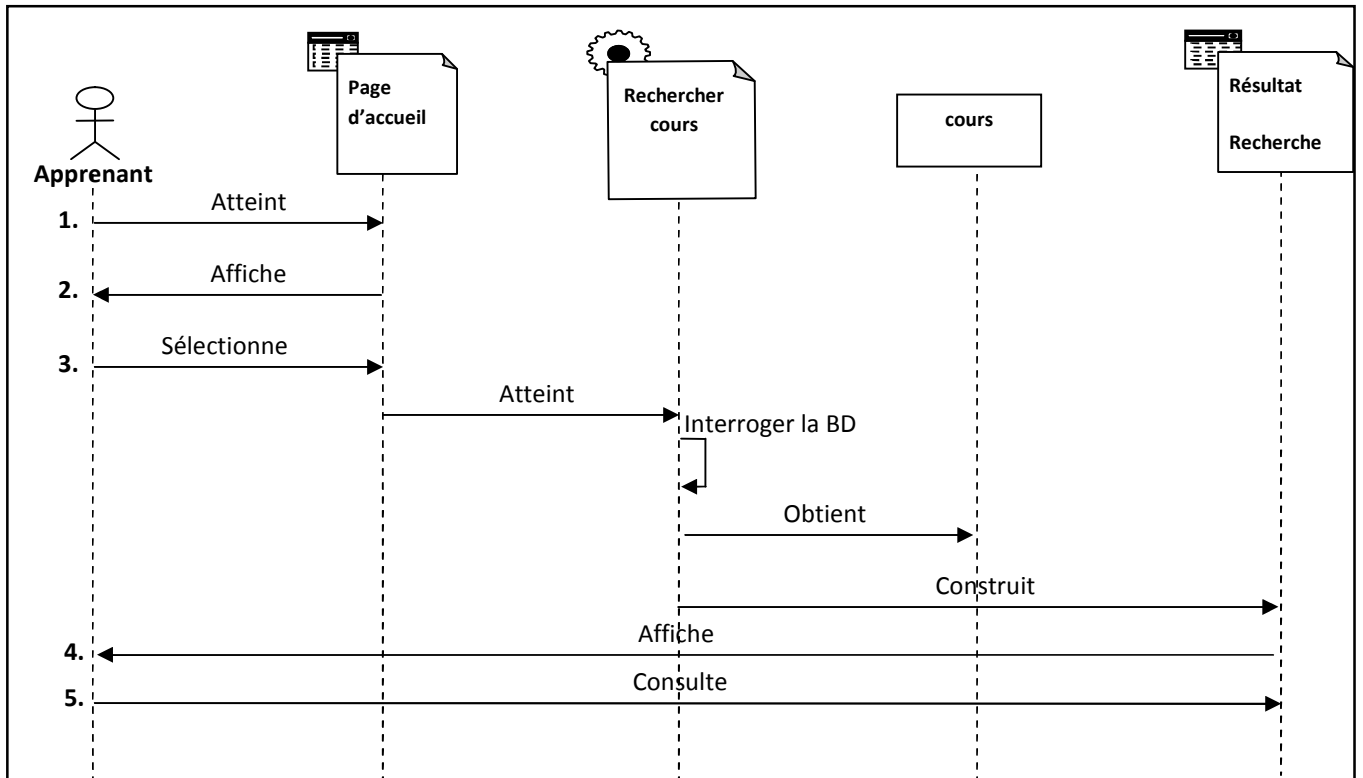


Figure IV.10 : Diagramme de séquence de réalisation du cas d'utilisation « consulter cours ».

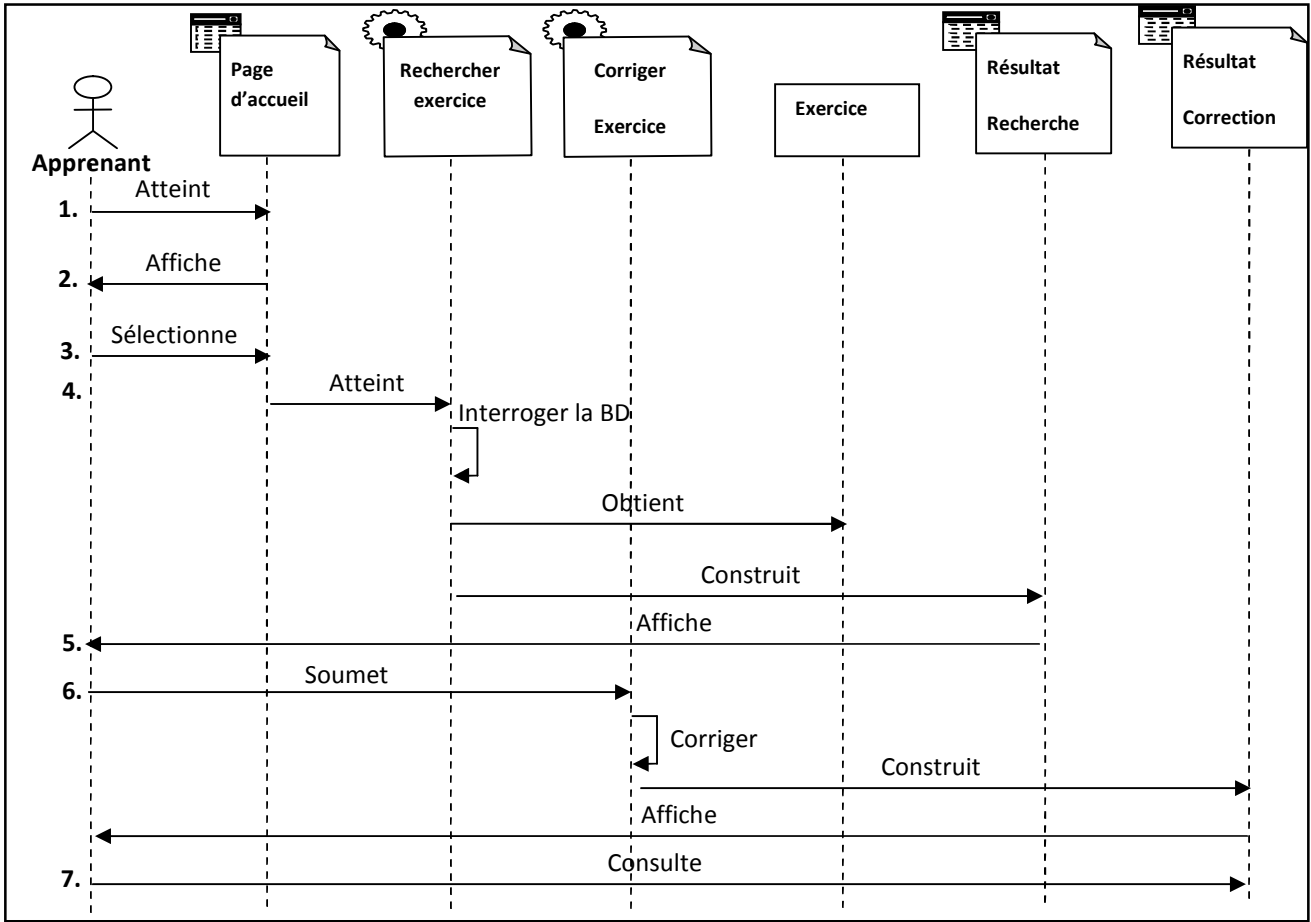


Figure IV.11 : Diagramme de séquence de réalisation du cas d'utilisation « faire exercice ».

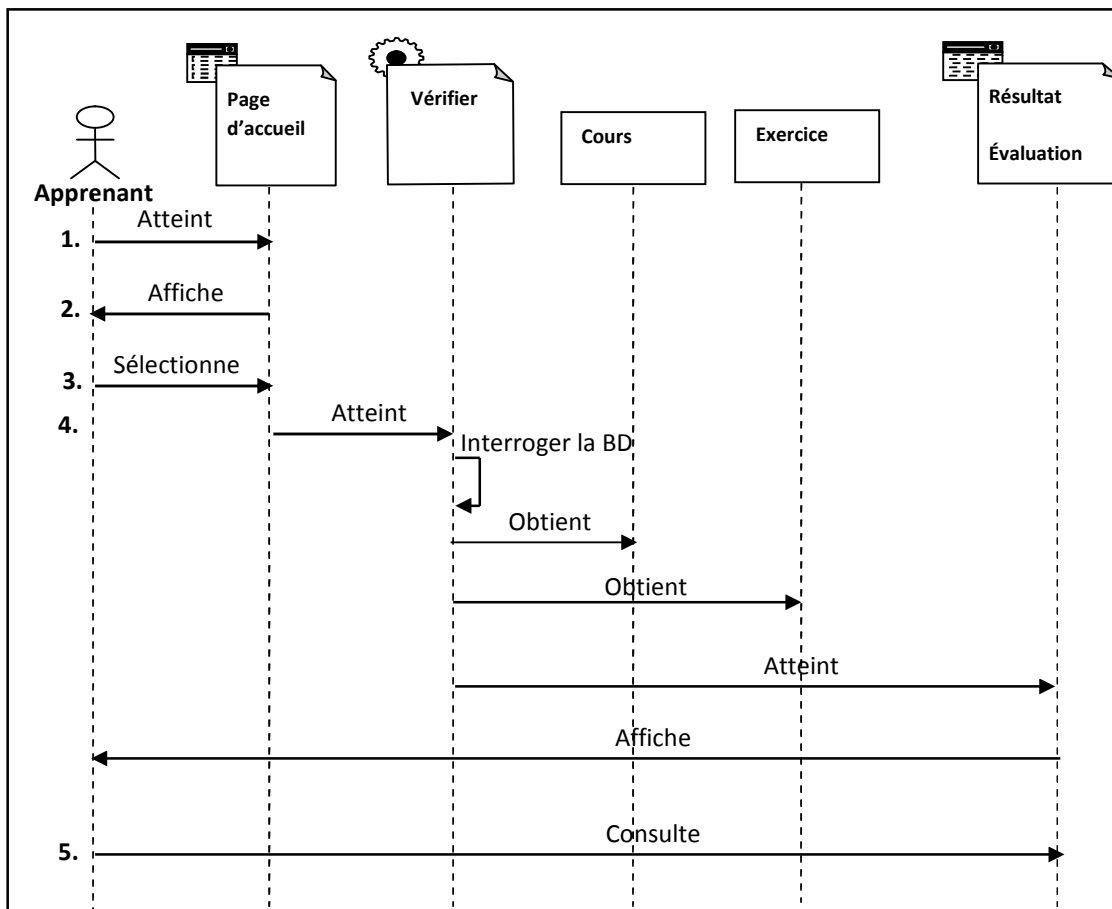


Figure IV.12 : Diagramme de séquence de réalisation du cas d'utilisation « évaluation ».

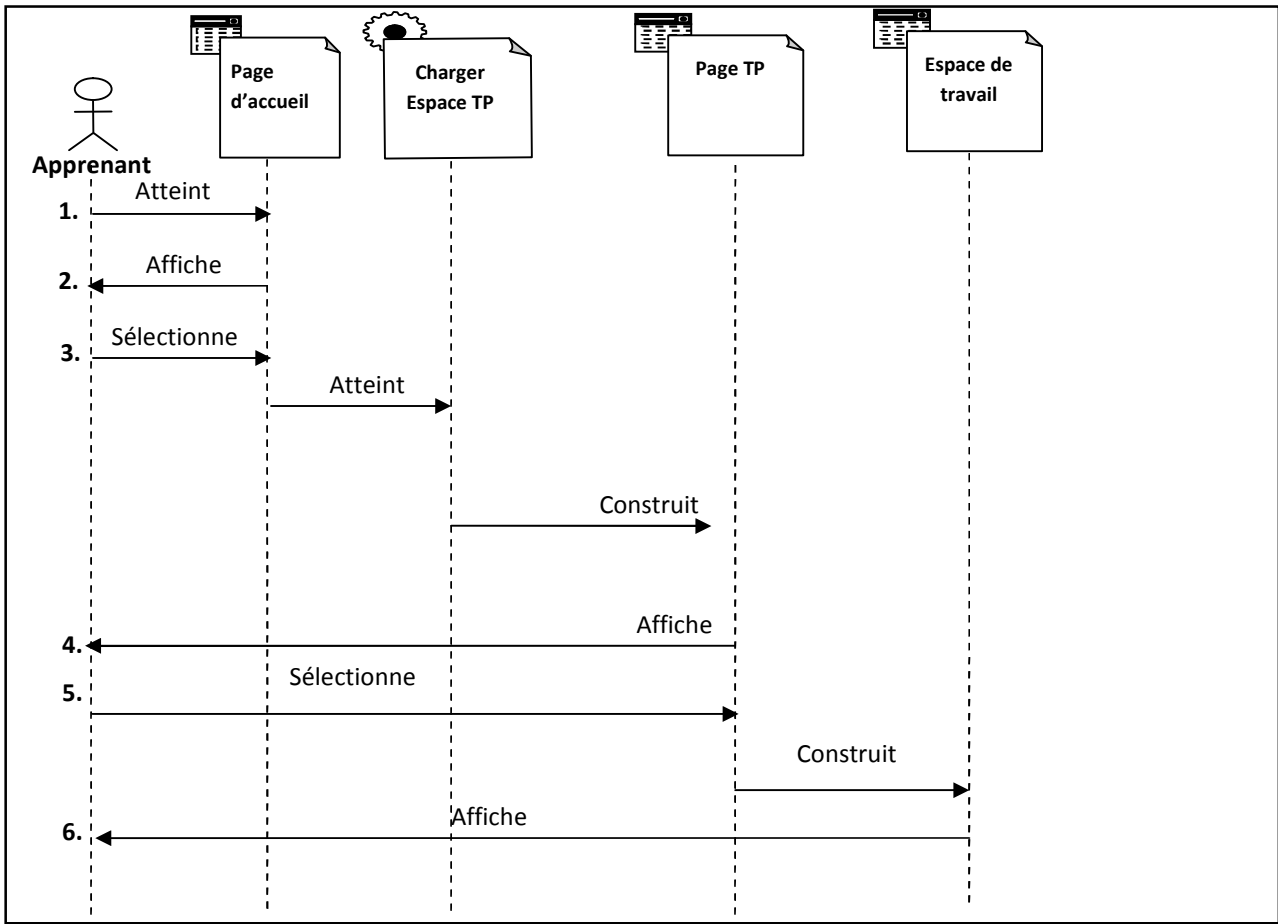


Figure IV.13 : Diagramme de séquence de réalisation du cas d'utilisation « Faire TP ».

3.2.6. Diagrammes de classes des cas d'utilisation

Pour exprimer les relations existantes entre les pages client et serveur nous devons construire les diagrammes de classes.

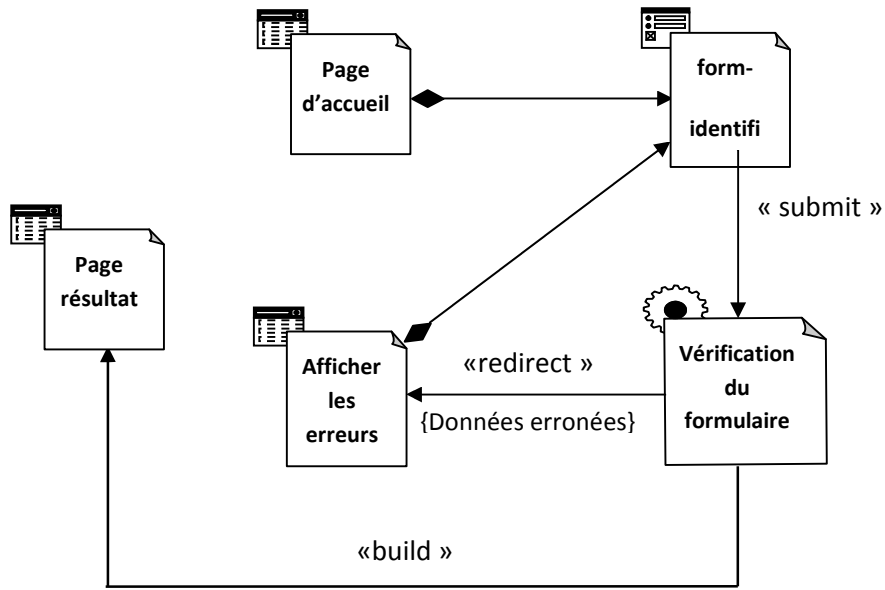


Figure IV.14 : Diagramme de classe général de cas d'utilisation « identification ».

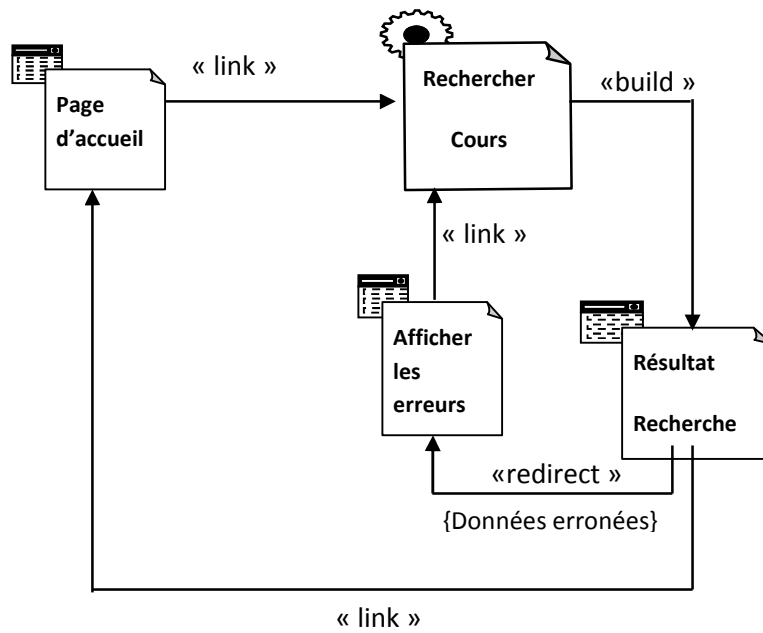


Figure IV.15 : Diagramme de classe général de cas d'utilisation « consulter cours ».

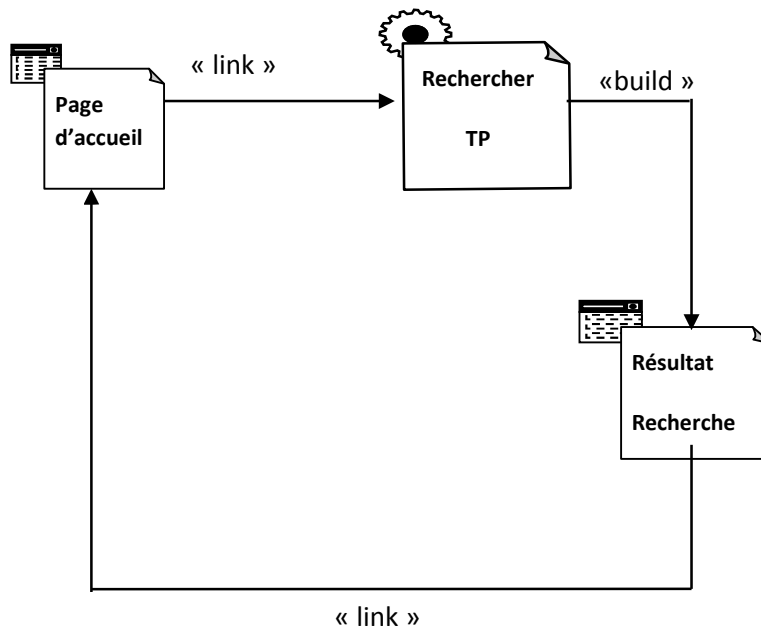


Figure IV.16: Diagramme de classe général de cas d'utilisation « Faire TP ».

3.2.7. Package de base de données

a. Le niveau conceptuel de la base de données

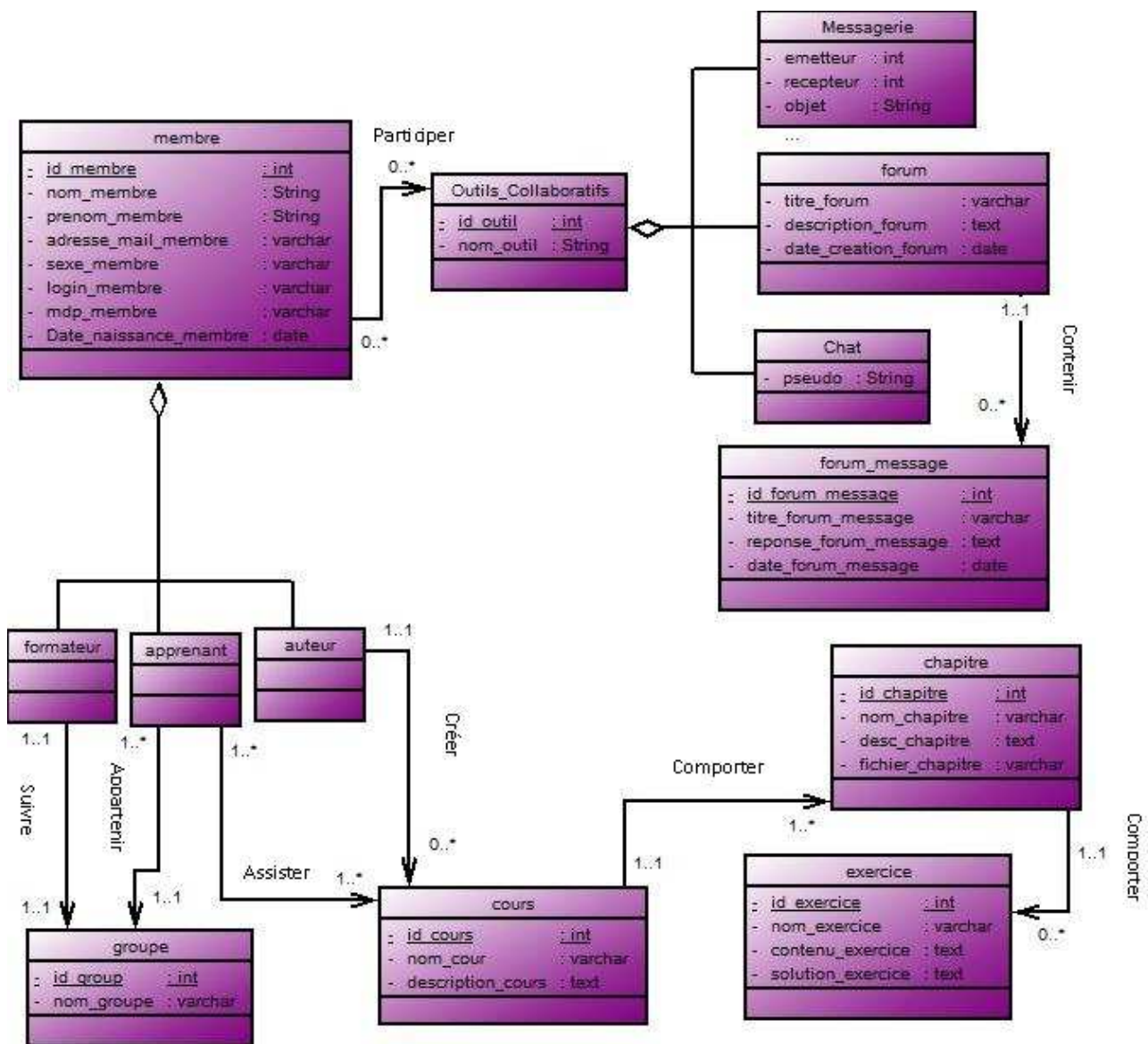


Figure IV.17: Diagramme de classe générale

b. Le niveau logique de la base de données

Membre (id_membre, nom_membre, prénom_membre, date_naiss_membre, adresse_membre, email_membre, sexe_membre, login_membre, mdp_membre, id_groupe_fk).

Groupe (id_groupe,nom_groupe).

Cours (id_cours,nom_cours,desc_cours).

Chapitre (**id_chapitre**, nom_chapitre, desc_chapitre, fichier_chapitre, **id_cours_fk**).

Exercices (**id_exercice**, nom_exercice, contenu_exercice, solution_exercice, **id_chapitre_fk**).

Forum (**id_forum**, titre_forum, desc_forum, date_création_forum, **id_membre_fk**).

Forum_Message (**id_forum_message**, titre_forum_message, reponse_forum_message, date_forum_message, **id_membre_fk**, **id_forum_fk**).

Messagerie (**id_message**, objet_message, contenu_message, **id_membre_em_fk**, **id_membre_dest_fk**).

Chat (**id_chat**, message_chat, pseudo).

Assister (**id_app**, **id_cours**).

4. Représentations des différents cas d'accès à l'espace de travail :

4.1. Représentation des instances d'accès à l'espace de travail:

Le schéma suivant spécifie le nombre d'instances d'utilisateurs connectés à l'espace de TPs à un moment donné.

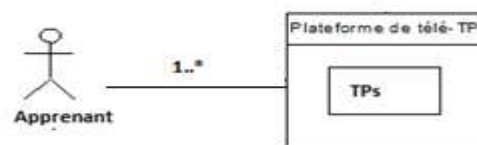


Figure IV.18: instances d'accès à l'espace de travail

4.2. Représentation du problème d'accès :

Plusieurs apprenants demandent l'accès à un TP à un moment donné hors, un seul accès est permis ainsi il y'a concurrence d'accès à un dispositif unique.

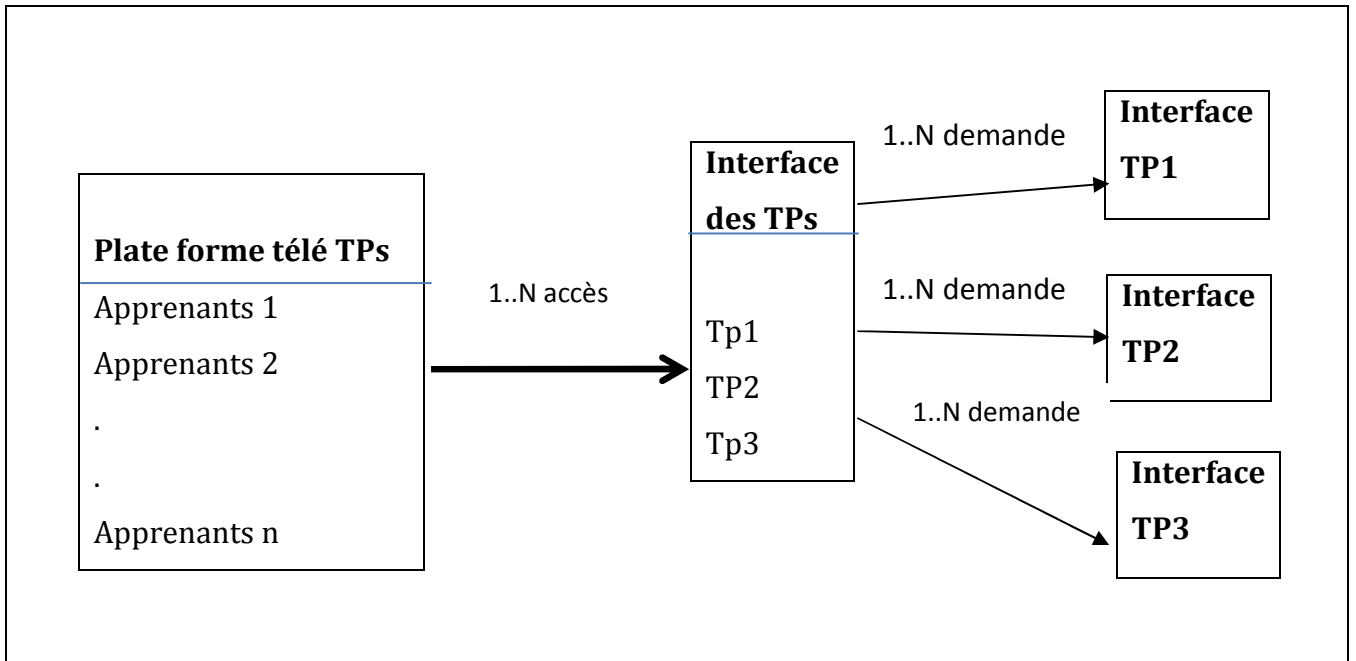


Figure IV.19: instances d'accès aux TPi

4.3. Représentation de la solution au problème d'accès

La solution consiste à intégrer des modules de gestions d'accès aux différentes interfaces TPi.

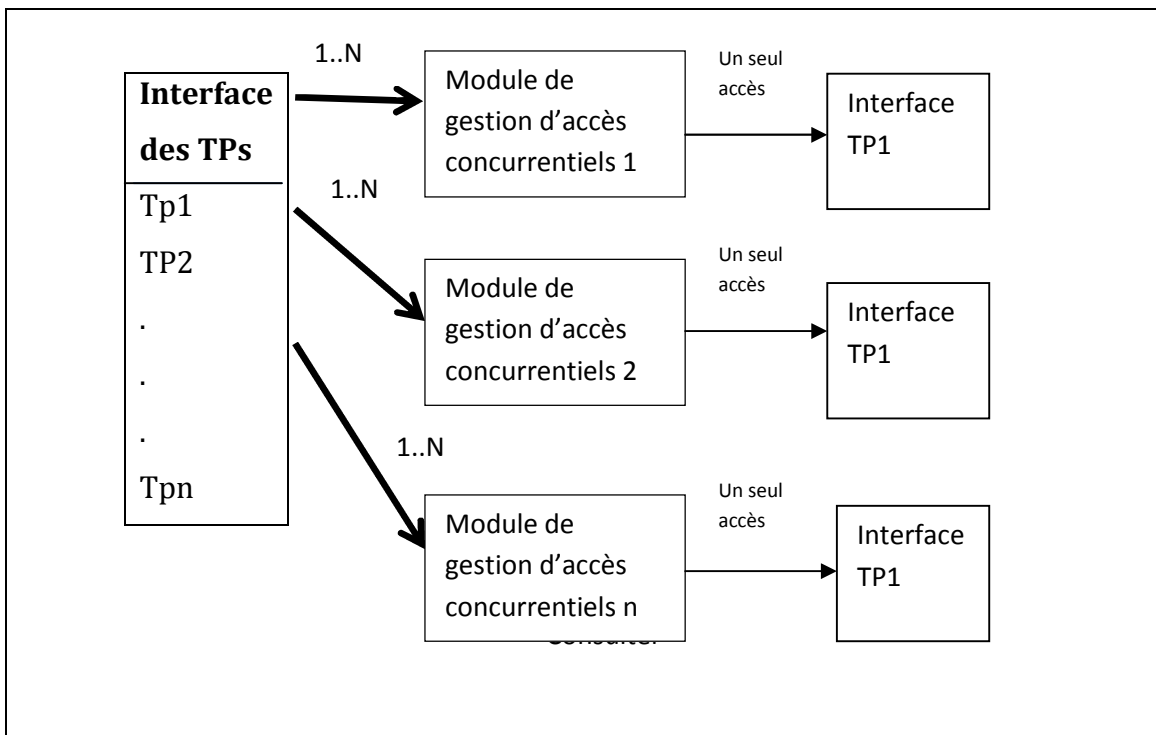


Figure IV.20: gestion d'accès aux TPi

Schéma détaillé du module de gestion d'accès concurrentiel :

Prenons l'exemple de l'interface TP n°1.

Pour assurer la gestion des accès concurrentiels dans notre espace de travail, on a opté pour une synchronisation avec sémaphore et une file d'attente (FIFO).

Pour chaque demande un thread est créé avec la fonction `new()`, puis cette demande est soumise au test suivant :

Si le dispositif est libre et que la file d'attente est vide alors le thread est lancé en faisant appel à la fonction `start()` qui donne l'accès au dispositif manipulant le TP, ainsi il détient le sémaphore (la fonction `acquire()`).

Si le dispositif n'est pas libre, le thread sera mis dans la file d'attente.

Une fois le sémaphore libéré (la fonction `release()`), un signal sera émis au thread de plus haute priorité (la fonction `notify()`) dans la file d'attente qui sera alors lancé.

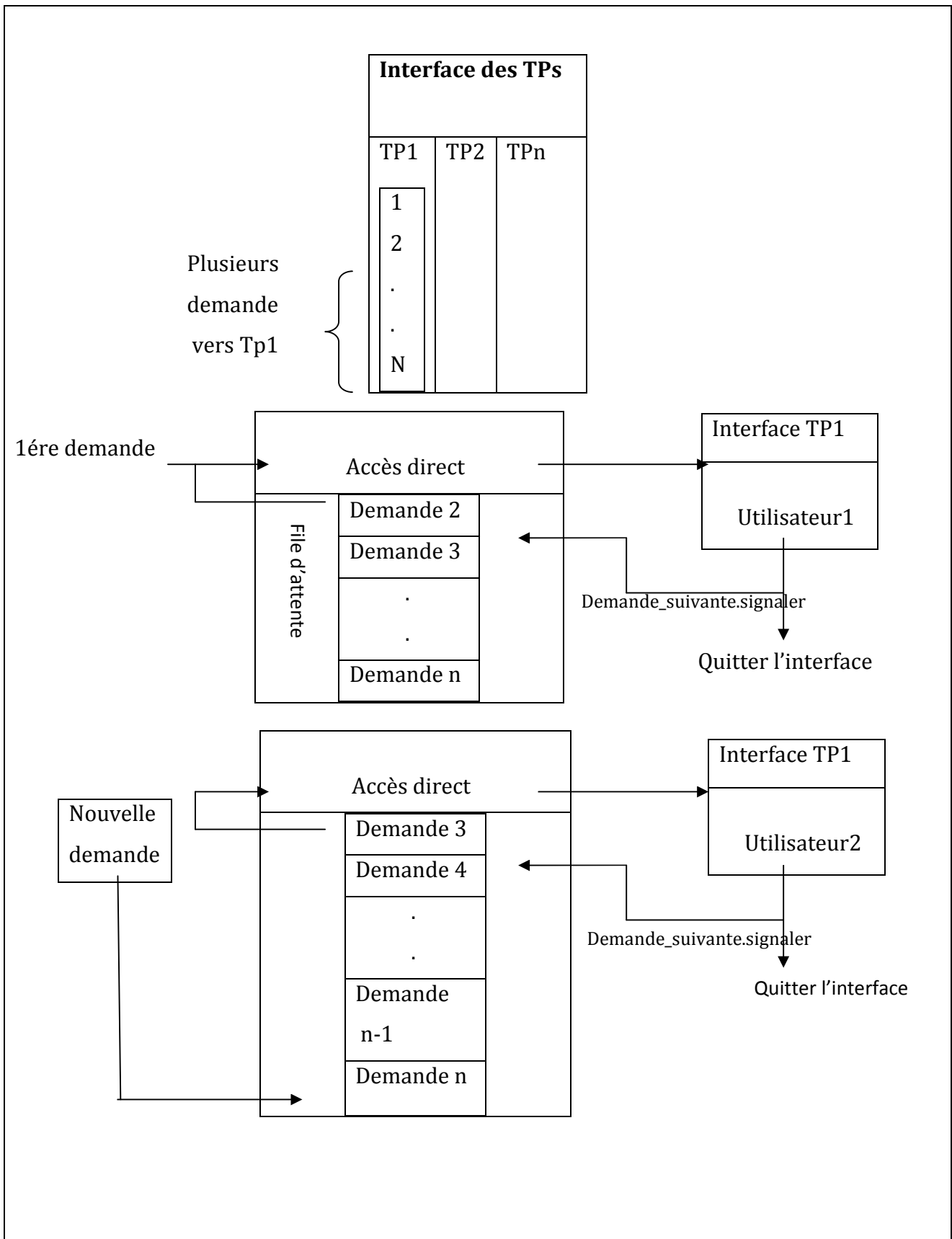


Figure IV.21: module de gestion d'accès aux TPi

5. Conclusion

A l'issue de ce chapitre, nous avons proposé une solution fondée sur une analyse et une conception en utilisant le Langage de Modélisation Unifié UML. Pour cela nous avons :

- Spécifier les divers cas d'utilisations de notre système ;
- Conçu les diagrammes des cas d'utilisations, de séquence et de déploiement ;
- Elaboré un diagramme de classe.

Ainsi qu'une schématisation de la gestion d'accès concurrentiels à notre plate forme.

Le chapitre suivant sera consacré à la mise en œuvre de notre système, en présentant les outils de développement utilisés et les différentes fonctionnalités de notre application a travers différentes interfaces.

Chapitre 5 : Réalisation

1. Introduction :

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement, les outils qui ont servi à la réalisation de notre application, et nous terminerons par la présentation de ses fonctionnalités à travers ses différentes interfaces.

2. L'environnement de développement et l'implémentation :

Afin de réaliser ce projet nous nous sommes appuyés sur les outils suivants :

- **Windows Seven** comme système d'exploitation ;
- **WampServer2.0** qui offre **Apache** comme serveur web, **MySQL** comme serveur de bases de données et **Phpmyadmin** comme interface graphique pour manipuler les bases de données;
- **PHP** pour la programmation des scripts, et **MySQL** pour les différentes requêtes qui font appel à la base de données.
- **Toad For MYSQL 4.6** qui permet d'explorer des bases de données MySQL, d'en gérer les objets, d'exécuter des requêtes et bien plus encore, tout cela graphiquement.
- **JAVA** pour la programmation de l'espace de travail.
- **Eclipse java EE** comme environnement de développement.
- **HTML 5** pour la programmation de la plate forme.

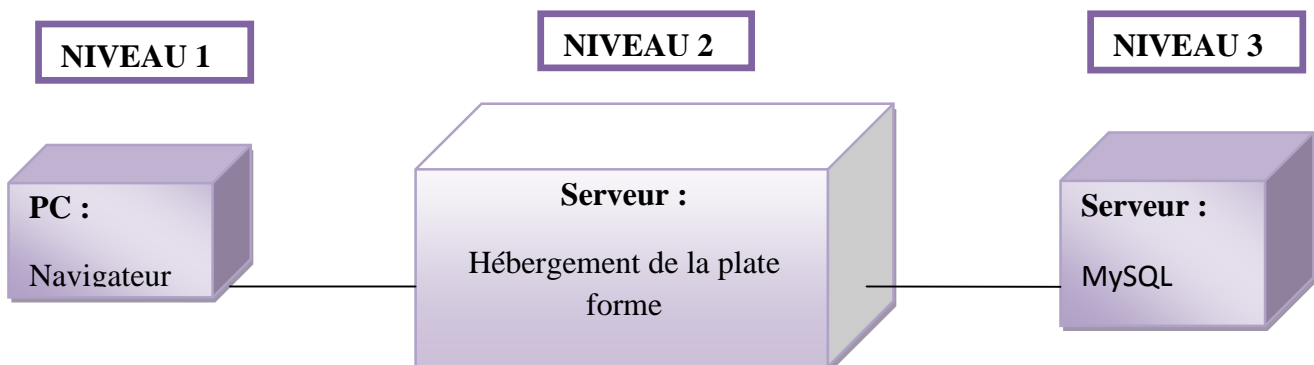


Figure V.1. Schéma de déploiement

2.1. Serveur :

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

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

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

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

L'ancêtre Apache est le serveur libre développé par le National Center for Supercomputing Applications de l'université de l'Illinois .L'évolution de ce serveur s'est arrêtée lorsque le responsable a quitté le NCSA en1994, les utilisateurs ont continué à corriger les bugs et à créer des extensions qu'ils distribuaient sous forme de "patches " (bouts de programmes ajoutés par les utilisateurs des NCSA pour étendre les fonctionnalités d'Apache)d'où le nom "a patchee Server" la version1.0 de Apache a été disponible le 1 Décembre 1995.

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

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

2.2. Serveur de base de données : [Leierer et Stoll, 2000]:

2.2.1. Serveur MySQL

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

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

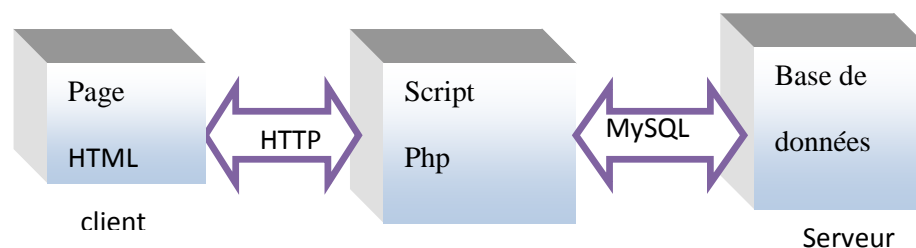


Figure IV.2 : Serveur MYSQL.

2.2.2. Fonctionnalités de MySQL

La liste suivante décrit quelques fonctionnalités importantes de MYSQL :

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

client grâce à l'hôte depuis lequel il se connecte et le nom d'utilisateur qu'il spécifie. Puis, l'associer avec les droits d'utilisation des commandes select, insert, update et delete sur cette base (le serveur vérifie chaque requête émise pour voir si elle est autorisée).

2.3. Les langages utilisés :

2.3.1. Le langage PHP : (Personal Home Page) [Leierer et Stoll, 2000]

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

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

- Le PHP est rapide : compilé en tant que module Apache, les temps d'exécution sont plus courts, surtout sous Unix ;
- simplicité d'écriture de scripts ;
- Le PHP est multi plates formes : il fonctionne sous Windows, Unix....
- Le PHP gère très bien les requêtes SQL : On peut facilement écrire des programmes qui affichent des données extraites de bases SQL, ou qui stockent des données postées par un formulaire dans une table SQL. Le PHP sait communiquer avec presque tous les SGBD (Oracle, MySQL, DB2, SQL Server, Access) mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur les plates formes Unix, Linux, Windows ;
- Le PHP fournit une multitude de fonctions : couvrant presque tous les besoins pour un développeur de sites Internet : prise en charge de XML, génération de PDF (Portable Document Format), création d'images, compression/décompression, statistiques, cryptologie, génération d'email....
- gratuité et disponibilité du code source ;
- intégration au sein de nombreux serveurs Web (Apache, Microsoft IIS, ...) ;
- simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur les plates formes Unix, Linux, et Windows).

➤ Fonctionnement de PHP

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

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

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

2.3.2. Le langage HTML et CSS :

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

Le terme **CSS** est l'acronyme anglais de *Cascading Style Sheets* qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage (une recommandation du W3C) informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

2.3.2.1. HTML 5 et CSS3 :

HTML 5 est une évolution logique du langage HTML, reposant sur les technologies modernes testées (HTML, XHTML, JavaScript, géo localisation) ou plus inintelligible telles que le « Web 2.0 » ou encore la gestion des supports audio et vidéo.

Les travaux sur HTML 5 ont commencé fin 2003 grâce à un groupe de travail indépendant, mais ce n'est qu'en 2007 que le W3C officialise véritablement ce langage en l'intégrant dans

leur groupe de travail. À partir de là s'est fait un gros effort afin de permettre à HTML 5 d'être rétro compatible avec ses ancêtres, ce qui a quelque peu ralenti son développement.

HTML 5 offre nativement une large panoplie de nouveautés et de technologies :

- une grammaire entièrement revue et simplifiée ;
- de nouveaux éléments sémantiques et de nouveaux attributs ;
- la reconnaissance de vidéos ou de sons sans plug-ins ;
- une gestion étendue des formulaires ;
- la possibilité de se localiser géographiquement et de profiter de cette information ;
- la création de dessins ou de graphiques à l'aide de l'élément `<canvas>` ;
- le stockage de données sur son ordinateur pour les exploiter hors ligne ;
- la possibilité d'intervenir sur les éléments en les modifiant à la volée ou en les déplaçant (*drag and drop*), etc.

CSS3 est le dernier standard du langage C, il offre des nouveautés comme la Puissance de sélecteurs, redimensionnement, ombres et effets,... etc.

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

MySQL dérive directement de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel. Il en reprend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MySQL (sélections imbriquées, clés étrangères,...etc.) Le serveur de base de données MySQL est très souvent utilisé avec le langage de création de pages web dynamiques : PHP.

2.3.4. Le langage JAVA :

Java est un langage de programmation a usage général, évolué et orienté objet dont la syntaxe est proche de C. il existe deux type de programme en java : les applets et les applications. Une application autonome (stand alone program) est une application qui s'exécute sous le control directe du système d'exploitation. Une applet est une application qui est chargée par un navigateur et qui est exécutée sous le control d'un plugin de ce dernier.

La source est compilé en pseudo code ou byte code puis exécuter par un interpréteur java :

La Java Virtuel Machine(JVM). Ce concept est a la base de slogan de Sun pour java : WORA

(write once, run anywhere : écrire une fois, exécuter partout). En effet, le byte code, s'il ne contient pas de code spécifique à une plate-forme particulière peut être exécuté et obtenir quasiment les mêmes résultats sur toutes les machines disposant d'une JVM.

Comme la plupart des langages récents, java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées l'une avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractères, flottant, booléen,...).

La sécurité fait partie intégrante du système d'exécution et du compilateur. Un programme java planté ne menace pas le système d'exploitation. Il ne peut pas y avoir d'accès direct à la mémoire.

Les principales différences entre un applet et une application sont :

Les applets n'ont pas de méthode `main ()` : la méthode `main ()` est appelée par la JVM pour exécuter une application.

Les applets ne peuvent pas être testés avec l'interpréteur de commande mais doivent être intégrés à une page HTML, elle est visualisée avec un navigateur disposant d'un plugin qui gère les applets java ou tester avec `applet viewer`.

2.4. Les outils de développement :

2.4.1. EasyPHP

EasyPHP est un utilitaire qui installe et configure automatiquement un environnement de travail complet pour le développement et le test des applications Web. Il regroupe un serveur web Apache, un serveur de base de données MySQL et le langage PHP ainsi que des outils qui facilitent le développement tel l'administrateur de bases de données MySQL, `PhpMyAdmin`.

a. *PhpMyAdmin* :

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

➤ Les fonctions de `PhpMyAdmin` :

- Création et suppression de base de données ;
- Création, modification, copie et suppression de table ;

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

2.4.2. Eclipse 3.5.1

Eclipse est un environnement de développement intégré (IDE) dont le but est de fournir une plateforme modulaire pour permettre de réaliser des développements informatiques. I.B.M est à l'origine du développement d'Eclipse qui est d'ailleurs toujours au cœur de son outil Websphere Studio Workbench (WSW), lui même à la base de la famille des derniers outils de développement en java d'I.B.M.

L'image suivante présente l'interface de travail sous Eclipse.

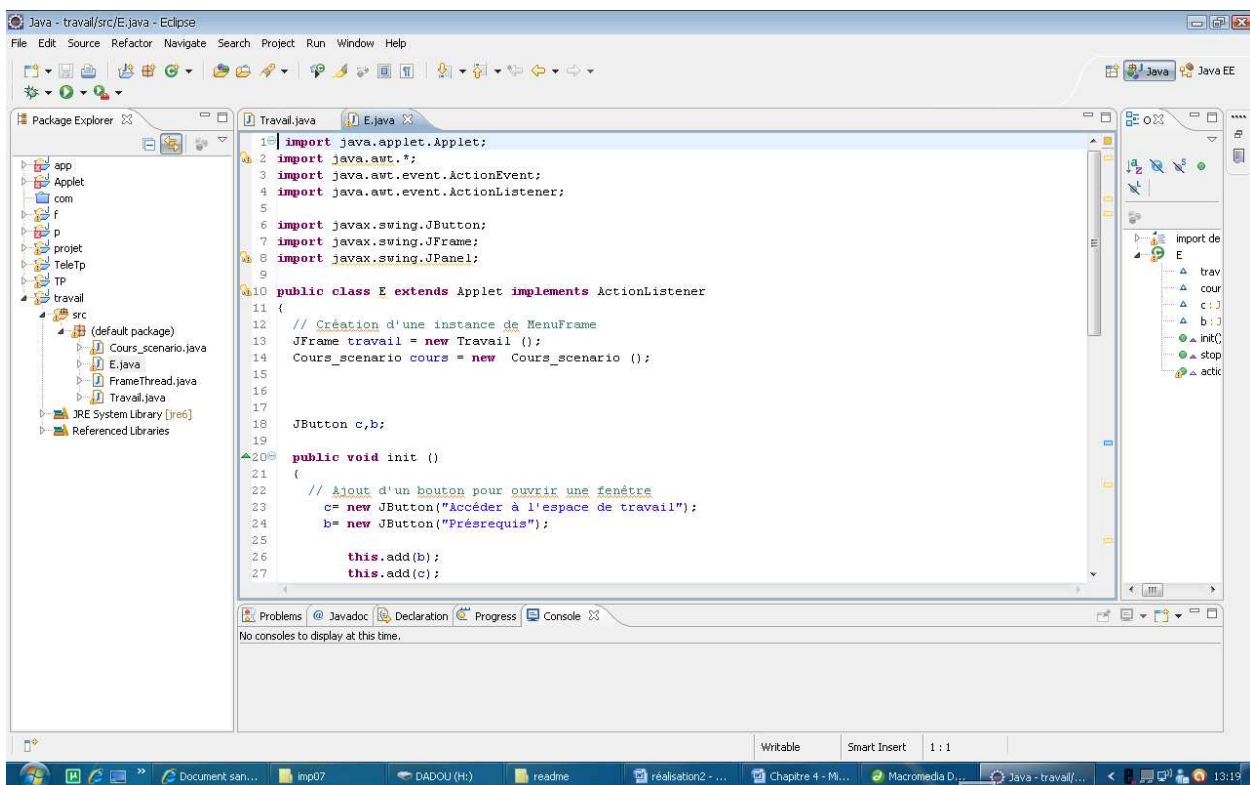


Figure IV.3 : Espace de travail de l'environnement de développement Eclipse

2.5. Autres Outils :

a. Dreamweaver® 8

Sous Windows, Dreamweaver propose une présentation intégrée en une seule fenêtre. Dans l'espace de travail intégré, toutes les fenêtres et tous les panneaux sont rassemblés dans une grande fenêtre d'application. C'est un éditeur de page HTML, convivial et simple à utiliser.



Figure IV.4 : Espace de travail de Dreamweaver

2.6. Navigateurs :

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

Dans notre cas on a opté pour Google chrome.

✓ Chrome



Chrome est un navigateur web développé par Google basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS. Google Chrome est un navigateur Web rapide, simple et sécurisé, conçu pour le Web d'aujourd'hui.

3. Mode de fonctionnement de l'application (partie TP)

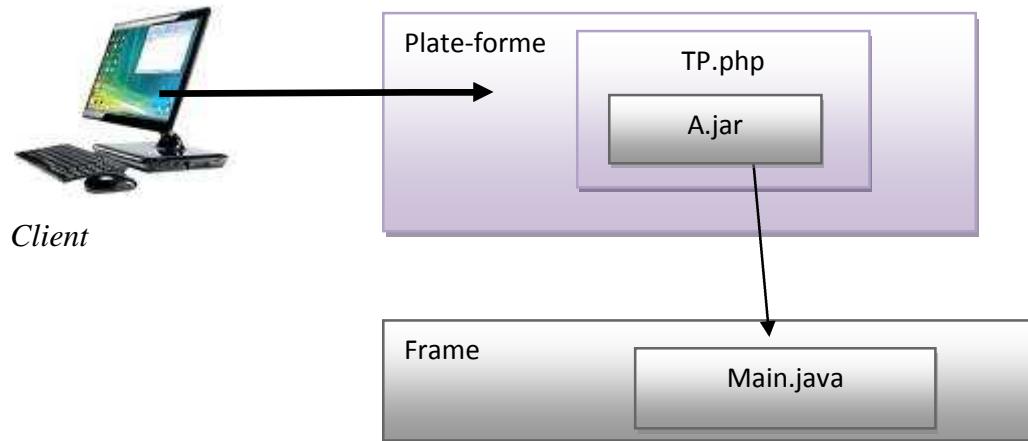


Figure IV.5 : mode de fonctionnement de la partie TP

Pour accéder à l'espace de travail l'apprenant doit cliquer sur le lien TP et le système lui affiche la page web (tp.php) qui contient une applet (A.java). Cette applet contient un lien qui permet d'accéder à l'espace de travail qui offre trois tps. On prend comme exemple le tp de mécanique qui sera lancé par le frame (TravailMecanique.java).

4. Quelques interfaces de notre application :

Pour réaliser une application Web il faut tenir compte de la qualité de l'interface homme/machine et permettre une meilleure adéquation de notre application aux besoins des différents utilisateurs (administrateur, apprenant, formateur) pour faciliter la navigation aux utilisateurs de l'application. Dans ce qui suit nous proposons quelques interfaces de notre Application.

Mais avons-nous précisons que notre plate forme est en responsive design c'est-à-dire qu'elle peut être visualisé sur un ordinateur de bureau une tablette et même un Smartphone.



Figure IV.6 : responsive design

4.1. La page d'accueil :

C'est la première page qui s'affiche lorsqu'un utilisateur saisit l'adresse : « localhost » dans son navigateur localement.

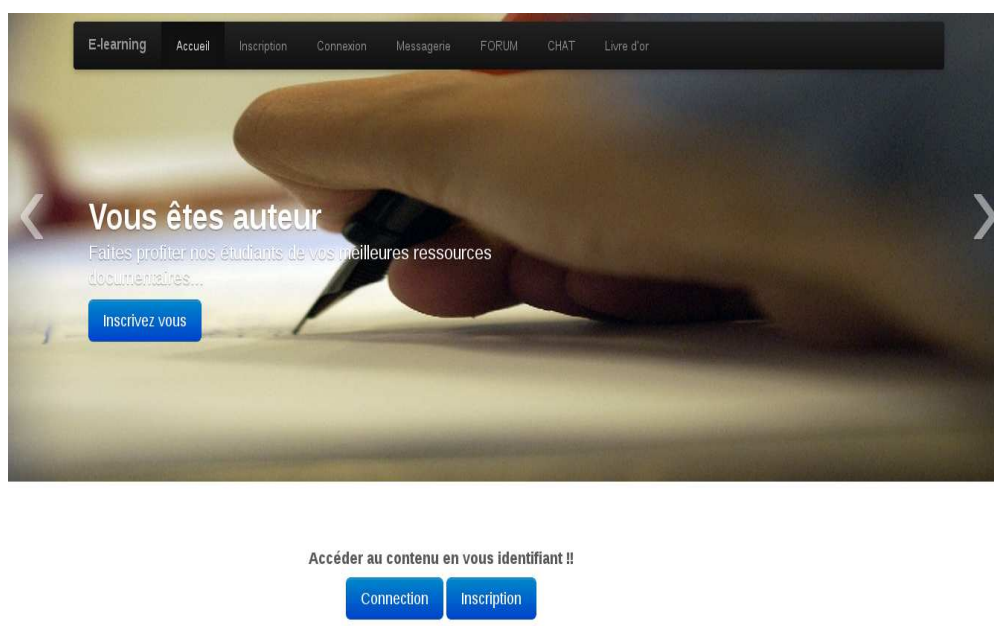


Figure IV.7 : Interface page d'accueil.

La barre de navigation comporte les liens suivants :

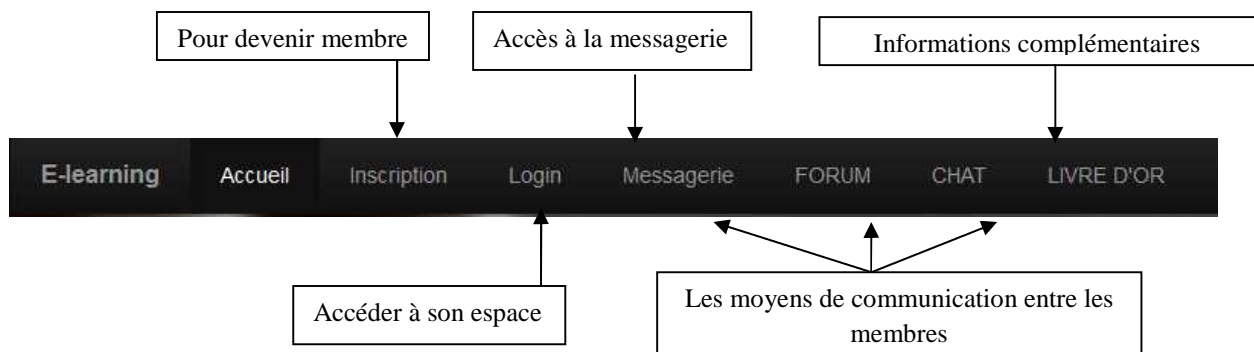


Figure IV.8 : barre de navigation

4.2. Inscription utilisateur :

Si l'utilisateur clique sur « inscription » de la barre de navigation ou sur le bouton « inscrivez-vous » qui s'affiche dans le cadre principale de la page d'accueil, il sera renvoyé à la page suivante pour remplir le formulaire d'inscription.

Devenez membre

Devenez membre

The registration form is titled 'INSCRIPTION' and contains the following fields and elements:

- NOM
- PRENOM
- Masculin (dropdown menu)
- Apprenant (dropdown menu)
- example@examle.com
- Date de naissance
- login
- Password
- Password
- Inscription (button)

Figure IV.9: Interface du formulaire d'inscription.

4.3. Espace auteur :



Figure IV.10: Interface espace auteur.

a. Ajout des cours :

L'auteur clique sur le bouton « cours » pour ajouter un cours une liste avec un bouton « Nouveau cours » qui mène vers un formulaire, l'auteur remplit le formulaire et clique sur ajouter puis remplit le formulaire associé pour l'ajout d'un chapitre constituant le cours.

Ajout cours

Figure IV.11 : Interface formulaire ajout cours.

Figure IV.12 : Interface formulaire ajout chapitre.

4.4. Espace Apprenant

L'espace qui permet aux apprenants d'utiliser les fonctionnalités de la plate forme qui lui sont offertes.

Figure IV.13 : Interface espace Apprenant.

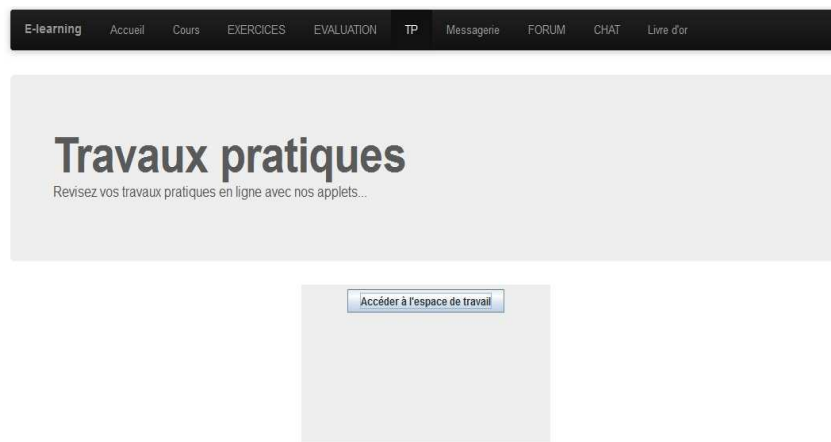
a. Quelques interfaces de déroulement du TP

Figure IV.14.1 : Interface accéder à l'espace de travail du TP.

Une fois cliqué sur le bouton accéder à l'espace de travail, l'apprenant verra l'interface de la figure suivante.



Figure IV.14.2 : Interface de l'espace de travail du TP.

On cliquant sur le bouton TPs, l'apprenant choisit un TP à manipuler.

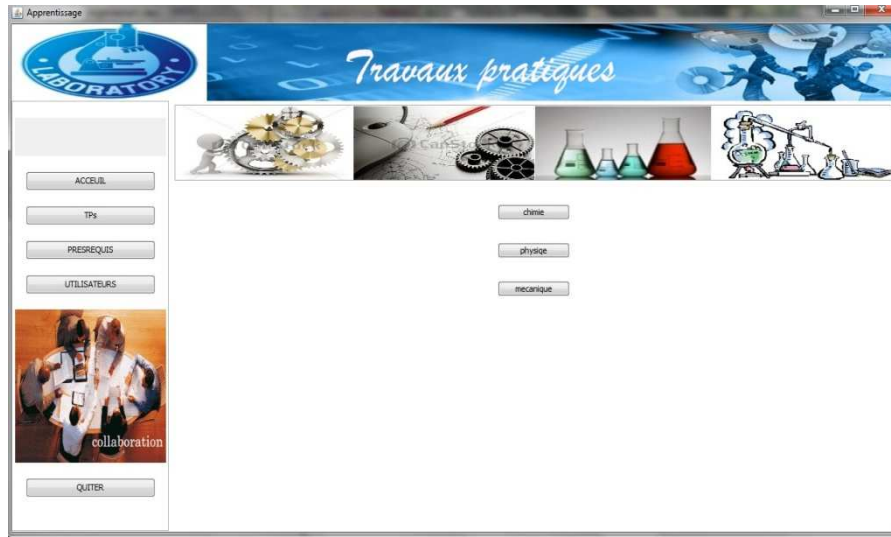


Figure IV.14.3: interface des propositions de TP

L'apprenant choisit par exemple le TP mécanique.

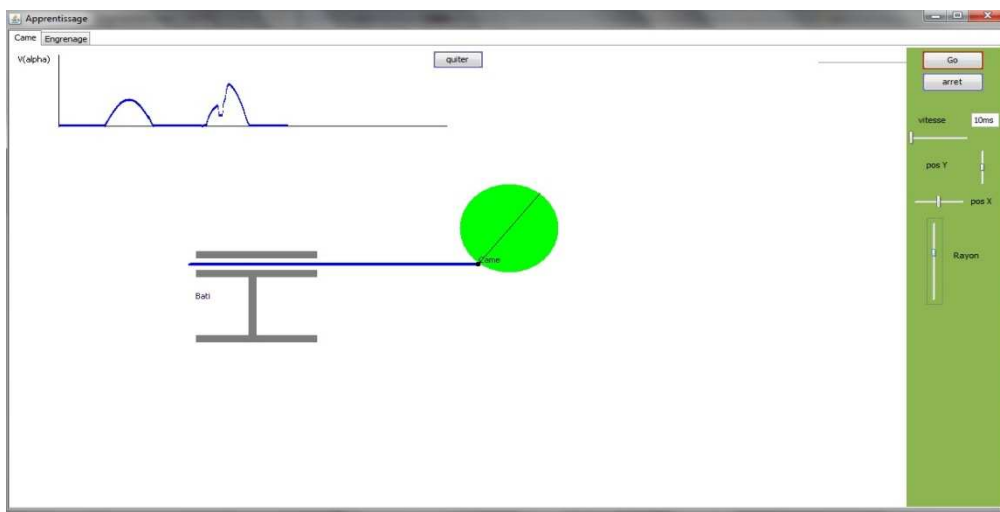


Figure IV.14.4 : interface du TP mécanique

Supposons qu'un autre apprenant demande l'accès au même TP, l'interface suivante s'affichera.

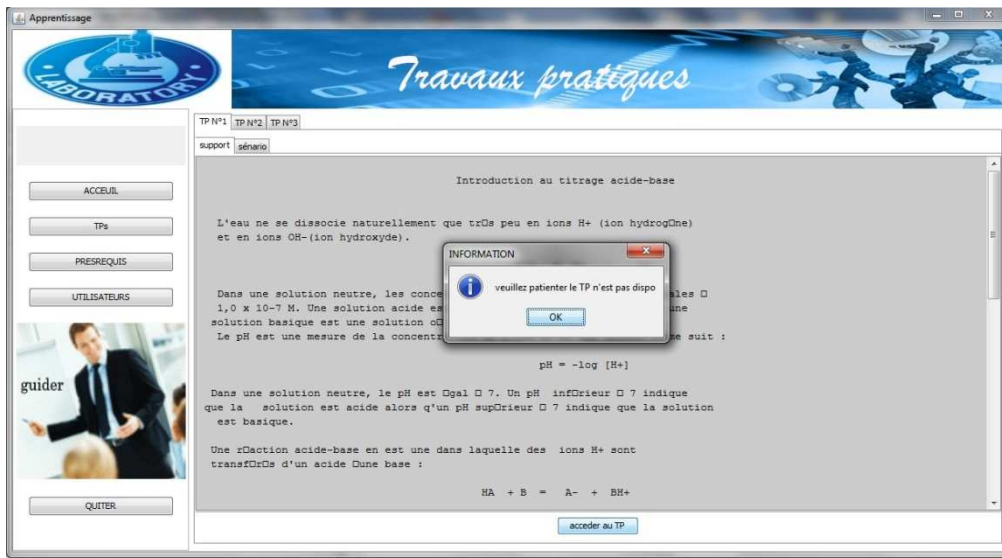


Figure IV.14.5 : mise en attente

L'apprenant 1 quitte le TP.

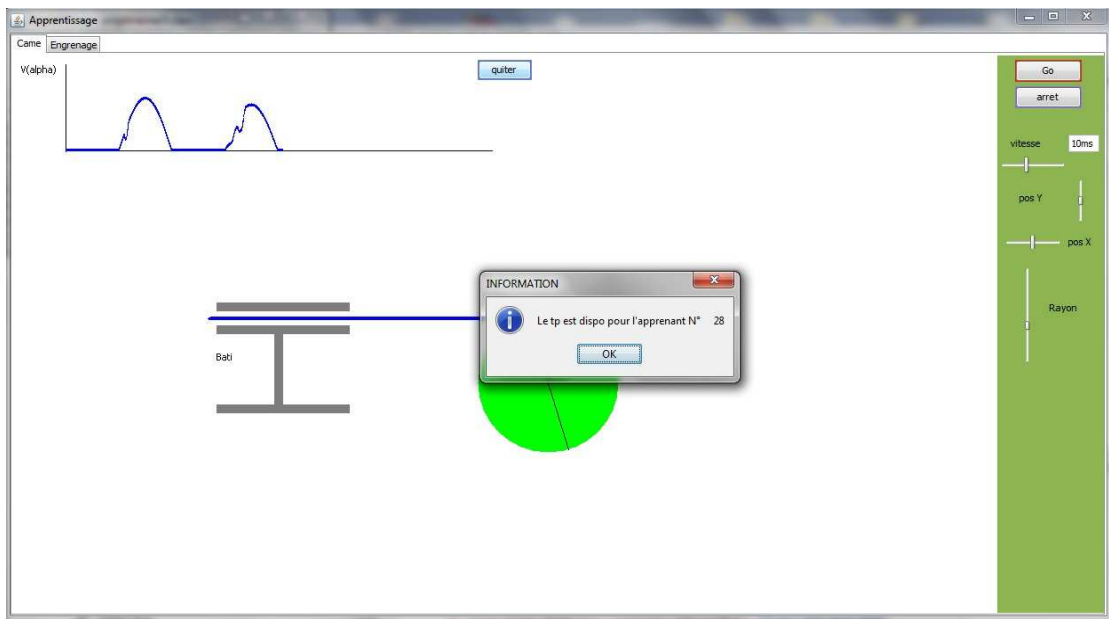


Figure IV.14.6: libération du TP

Et on cliquant sur le bouton ok la figure IV.14.4 sera affichée pour l'apprenant 2.

5. Conclusion :

Dans ce chapitre, nous avons présenté l'environnement de développement et l'implémentation de notre application. Nous avons également présenté quelques interfaces relatives à la gestion et le suivi des apprenants dans notre plate forme de formation à distance.

Conclusion générale

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.

L'objectif fixé dans notre projet était de concevoir et surtout de gérer les problèmes d'accès concurrentiels à un dispositif de TP. C'est pour cela que nous avons commencé par un état de l'art sur les téléTP afin d'en situer l'état actuel en la matière. Ensuite nous avons abordé les différents problèmes d'accès concurrentiels pour mieux les comprendre et ensuite proposer une solution avec les threads et sémaphores de sorte à atteindre notre objectif.

Bilan et Perspectives

La plateforme que nous avons réalisée offre des outils de base nécessaires pour approcher au maximum les méthodes traditionnelles de l'enseignement : un espace public (partagé) et des espaces personnalisés pour les différents acteurs (Apprenants, Formateurs, Administrateur, Auteur), des outils de communication, des dispositifs de TP et surtout de l'interactivité pour simuler au mieux la pédagogie réelle.

Même si nous avons pu proposer une solution afin de remédier aux problèmes d'accès concurrentiels mais les problèmes liés aux TéléTP sont encore présent. Ainsi il serait intéressant d'élargir nos objectifs en essayant de résoudre les problèmes de sécurité du matériel et des humains, la restitution en temps réel et à distance des événements et la mise à disposition d'outils d'édition de scénarios de télé-TP pour les tuteurs.

Annexes

Annexe UML

I. Introduction :

La notation UML est une fusion des notations de Booch, OMT et OOSE et d'autres notations. Les concepteurs de cette notation ont recherché avant tout la simplicité. Les symboles embrouillés, redondants ou superflus ont été éliminés en faveur d'un meilleur rendu visuel.

UML n'est pas une notation fermée : elle est générée, extensible et configurable par l'utilisateur UML, aussi elle ne cherche pas la spécification à outrance, il n'y a pas une représentation graphique pour tous les concepts imaginables ; en cas de besoins particuliers, des précisions peuvent être apportées au moyen de mécanisme d'extension et de commentaire textuel.

Dans cette annexe nous présentons la notation UML, ainsi que son extension pour le web utilisée dans notre mémoire, ainsi que sa schématisation dans l'AGL. (Pour plus de détails sur UML, consulter les livres traitant ce langage dont une liste figure en bibliographie).

II. Les briques de base d'UML :

La terminologie d'UML inclut trois sortes de briques :

- Des éléments.
- Des relations.
- Des diagrammes.

III. Elements d'UML :

1. Les éléments structurels :

Les éléments structurels sont représentés par des noms dans les modèles UML. Ce sont les parties les plus statiques d'un modèle : ils représentent des éléments conceptuels ou physiques.

1.1. La classe :

Elle représente un ensemble d'éléments qui partagent les mêmes attributs, les mêmes opérations, les mêmes relations et les mêmes sémantiques.

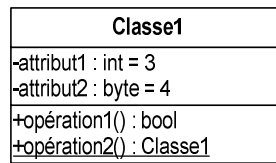


Figure 01 : Classes

1.2. Les classes-associations :

Il est possible de représenter une association par une classe pour ajouter, par exemple, des attributs et des opérations dans l'association. Une classe de ce type, appelée **classe- associative** ou **classe-association**, possède à la fois les caractéristiques d'une classe et d'une association, et peut à ce titre participer dans d'autres relations dans le modèle.

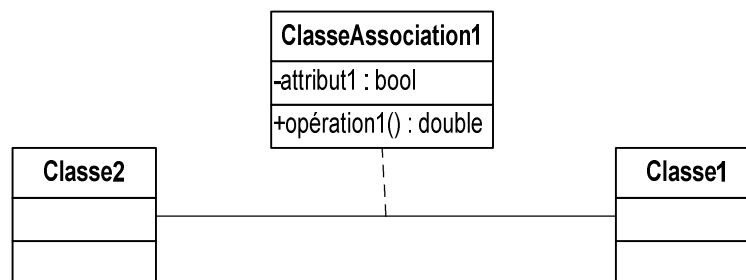


Figure 02 : Classe d'association

1.3. Les cas d'utilisation :

Un cas d'utilisation est la description d'une séquence d'actions exécutées par un système pour conduire à un résultat qui peut être constaté par un acteur particulier .Il sert à structurer les éléments comportementaux d'un modèle et est réalisé par une collaboration comme le montre la **Figure 03**, un cas d'utilisation est représenté par une ellipse en trait plein qui, en règle générale, contient seulement son nom.



Figure 03 : Cas d'utilisation

2. Les éléments comportementaux :

Les éléments comportementaux représentent les parties dynamiques des modèles UML. Ce sont les verbes du modèle et ils représentent son comportement dans le temps et dans l'espace.

2.1 Les interactions :

Une interaction est un comportement qui comprend un ensemble de messages échangés au sein d'un groupe d'éléments, dans un contexte particulier, pour atteindre un but bien défini. Le comportement d'un ensemble d'objets ou celui d'une opération individuelle peut être précisé par une interaction. Cette dernière implique un certain nombre d'éléments, y compris des messages, des séquences d'actions (comportement induit par un message) et des liens (relations entre des éléments). Comme le montre la **Figure 04**, un message est représenté par une ligne fléchée, qui indique le nom de son opération.

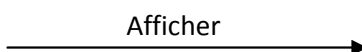


Figure 04 : Message

3. Les éléments de regroupement :

Les éléments de regroupements représentent les parties organisationnelles des modèles UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé. Il existe un seul type fondamental d'éléments de regroupement : le « paquetage ».

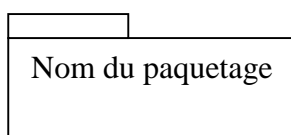


Figure 05 : Paquetage

4. Les éléments d'annotation :

Les éléments d'annotation représentent les parties explicatives des modèles UML. Ce sont les commentaires qui peuvent accompagner tout élément dans un modèle, à des fins de description, d'exploitation et de remarque. Il existe un type fondamental d'éléments d'annotation appelé « note » qui est simplement un symbole utilisé pour représenter les contraintes et les commentaires rattachés à un élément ou un ensemble d'éléments.

Comme le montre la **Figure 06**, une note est représentée par un rectangle écorné qui contient un commentaire textuel ou graphique

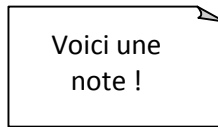


Figure 06 : Note

IV. Relation dans UML :

1. Dépendance :

C'est une relation sémantique entre deux éléments selon laquelle un changement apporté à l'un (élément indépendant) peut affecter la sémantique de l'autre (élément dépendant) comme le montre la **Figure 07**, une dépendance est représentée par une ligne en pointillés qui peut être fléchée ; elle comprend parfois une étiquette.

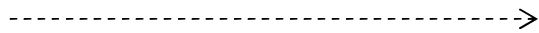


Figure 07 : Dépendance

2. Association :

Une association est une relation structurelle qui décrit un ensemble de liens, un lien constituant une relation entre différents objets .L'agrégation est un type particulier d'association, qui représente une relation structurelle entre un tout et ses parties. Comme le montre la **Figure 08**, une association est représentée par une ligne qui peut être fléchée ; elle comprend parfois une étiquette et souvent d'autres décorations, comme la multiplicité et les noms de rôles.

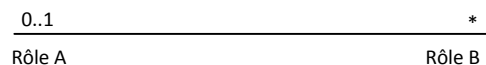


Figure 08 : Association

3. Généralisation :

Une généralisation est une relation de spécialisation selon laquelle les attributs de l'élément spécialisé (l'enfant) peuvent se substituer aux attributs de l'élément généralisé (le parent). De cette manière, l'enfant partage la structure et le comportement du parent. Comme le montre la **Figure 09**, une relation de généralisation est représentée par une flèche dont le trait est plein et dont la pointe creuse est dirigée vers le parent.



Figure 09 : Généralisation

4. Réalisation :

La réalisation est une relation sémantique utilisée principalement soit pour indiquer qu'une interface est réalisée par une classe, soit pour indiquer qu'un cas d'utilisation est réalisé par une collaboration d'objets. On la représente par :

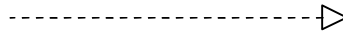


Figure 10 : Réalisation

V. Extensibilité :

UML comporte des mécanismes qui permettent d'étendre la syntaxe et la sémantique du langage.

1. Stéréotypes :

Les stéréotypes représentent de nouveaux éléments de modélisation, ils constituent un moyen de classer les éléments de la modélisation et facilitent l'élaboration du méta modèle d'UML.

Ils s'appliquent principalement aux classes et rendent possible l'identification d'une typologie de classe souvent nécessaire lorsqu'on manipule un grand nombre de classes.

Le nom du stéréotype est indiqué entre guillemets.

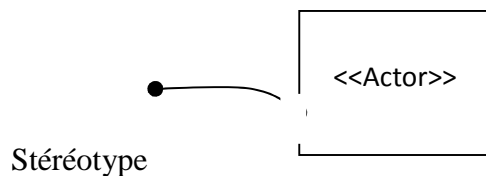


Figure 11 : Schéma représentatif d'une classe stéréotypée

- 2. Les contraintes:** Une contrainte est une note ayant une valeur sémantique particulière pour un élément de la modélisation, elle s'écrit entre accolades {}, elle peut concerner plusieurs éléments

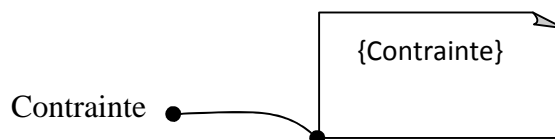


Figure 12 : Schéma représentatif d'une contrainte

VI. Diagrammes dans UML :

Un diagramme est une représentation graphique d'un ensemble d'éléments qui constituent un système. La plupart du temps, il se présente sous la forme d'un graphe connexe où les sommets correspondent aux éléments et les arcs aux relations. Les diagrammes servent à visualiser un système sous différentes perspectives et sont donc des projections d'un système.

1. Le diagramme de classes:

Le diagramme de classes est un diagramme structurel qui exprime d'une manière générale la structure statique d'un système en termes de classes, d'interfaces et de collaborations, ainsi que leurs relations.

2. Le diagramme d'objets :

Le diagramme d'objets appelé aussi diagramme d'instance représente aussi une structure statique et montre des objets et des liens.

Les notations retenues pour les diagrammes d'objets sont dérivées de celles des diagrammes des classes ; les éléments qui sont des instances sont soulignés.

Pour le nom de l'objet on peut le désigner sous trois (03) formes :

- Nom de l'objet : désignation directe et explicite du nom d'objet
- Nom de l'objet : **nom de la classe**: Désignation incluant le nom de la classe.
- Nom de la classe : désignation anonyme d'un objet d'une classe donnée.

3. Le diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation représente les cas d'utilisation, les acteurs et les relations entre eux.

Un acteur représente un rôle joué par une personne ou une chose qui interagit avec un système, il est représenté sous la forme de personnage et déclenche des cas d'utilisation. Il existe quatre (04) catégories d'acteurs à savoir :

- Les acteurs principaux : ce sont les personnes qui utilisent les fonctions principales du système.
- Les acteurs secondaires : ce sont des personnes qui effectuent des fonctions secondaires du système.
- Les matériels externes : Ce sont des dispositifs matériels nécessaires pour être utilisés.

- Les autres systèmes : ce sont des systèmes avec lesquels le système doit interagir. On peut schématiser l'acteur par la figure suivante :



Figure 13 : Représentation d'un acteur

UML définit trois (03) types de relation pour le diagramme de cas d'utilisation.

- a. La relation de communication** : elle est signalée par une flèche entre l'acteur et le cas d'utilisation. Comme la montre la figure suivante :

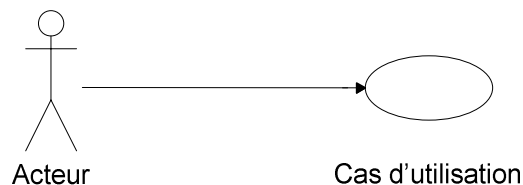


Figure 14 : Représentation d'un déclenchement d'un cas d'utilisation par un acteur

- b. La relation d'utilisation** : Une relation d'utilisation entre cas d'utilisation signifie qu'une instance du cas d'utilisation source comprend également le comportement décrit par le cas d'utilisation destination.

La figure suivante montre une relation d'utilisation :



Figure 15 : Représentation de la relation d'utilisation ou d'inclusion

- c. Relation d'extension** : Une relation d'extension entre cas d'utilisation signifie que le cas d'utilisation source étend le comportement du cas d'utilisation destination.

La figure suivante montre une relation d'extension.

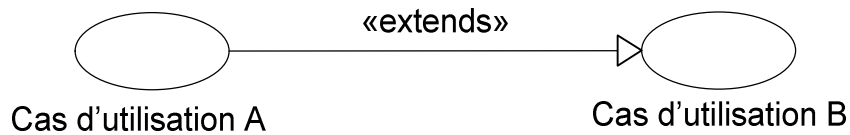


Figure 16 : Représentation de la relation d’extension.

4. Le diagramme de séquences :

Le diagramme de séquences montre les interactions entre objets selon un point de vue temporel. La représentation du contexte des objets se concentre sur l’expression des interactions.

Un objet est matérialisé par un rectangle et une barre verticale appelée ligne de vie des objets. Les objets, communiquent en échangeant des messages représentés au moyen de flèches orientées, de l’émetteur du message vers le destinataire. L’ordre des messages est donné par leur position sur l’axe vertical.

Les autres diagrammes UML sont :

- Le diagramme des activités qui décrit le comportement d’une opération en termes d’actions.
- Le diagramme de collaboration qui est une représentation spatiale des objets, des liens et des interactions.
- Le diagramme de composants qui décrit les composants physiques d’une application.
- Le diagramme de déploiement qui décrit les composants sur les dispositifs matériels.
- Le diagramme d’états transitions qui décrit le comportement d’une classe en terme d’états.

VII. L’extension d’UML pour le web :

1. Description :

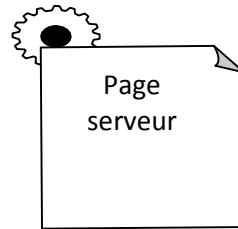
L’extension d’UML pour le Web définit un ensemble de stéréotype, d’étiquettes et de contraintes, qui rend possible la modélisation Web.

2. Stéréotypes :

2.1. classe

2.1.1. Page serveur « Server page »

Icône



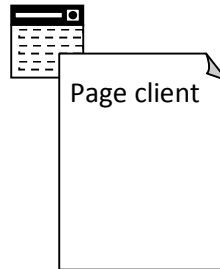
Description : Représente une page Web possédant des scripts qui interagissent avec des ressources serveur telles que les bases de donnée, ces scripts sont exécuté par le serveur.

Contraintes : Les pages serveur ne peuvent avoir de relation qu'avec des objets sur le serveur.

Etiquettes : Moteur de script qui peut être un langage ou le moteur qui doit être utilisé pour exécuter ou interpréter cette page.

2.1.2. Page client « client page »

Icône



Description : Une instance d'une page client est une page Web formatée en HTML. Les pages clients peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts de la page web.

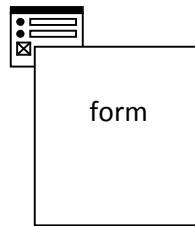
Contrainte : Aucune.

Etiquettes

- Titre (Title) : Titre de la page tel qu'il est affiché par le navigateur.
- Base (Base) : URL de base pour dé-référencer les URL relatives.
- Corps (Body) : ensemble des attributs de la balise <body>, qui définit des caractéristiques par défaut du texte et de l'arrière-plan.

2.1.3. Formulaire

Icône



Description : Une classe stéréotypée « form » est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML <form>, les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zone de saisie, zone de texte, boutons d'option.).

Un formulaire n'a pas d'opérations, puisqu'il peut les encapsuler. Toute opération qui interagit avec le formulaire appartient à la page qui la contient.

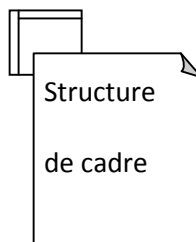
Contraintes : Aucune.

Etiquettes

GET ou POST – Méthodes utilisées pour soumettre les données à l'URL de l'attribut action de la balise HTML <form>.

2.1.4. structure de cadres « frameset »

Icône



Description : Une structure de cadres est un conteneur de plusieurs pages Web. La zone d'affichage rectangulaire est divisée en cadres rectangulaires inscrits. A chaque cadre peut être associé un nom unique de cible « Target ».Le contenu d'un cadre peut être une page Web ou une structure de cadre.

Une classe stéréotypée « frameset » est directement associé à une structure de cadre de page Web par la balise HTML < frameset >.

Une structure de cadre est une page client qui peut posséder des opérations et des attributs.

Contraintes : Aucune

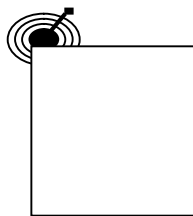
Etiquettes

Rangées (rows) : valeur de l'attribut rows de la balise HTML <framset>. C'est une chaîne de pourcentages séparés par des virgules, définissant les hauteurs relatives des cadres.

Colonnes (cols) : valeur de l'attribut cols de la balise HTML <frameset>. C'est une chaîne de pourcentages séparés par des virgules, définissant les largeurs des cadres.

2.1.5. cible « target »

Icône



Description : Une cible est une zone nommée dans la fenêtre du navigateur dans laquelle des pages Web peuvent être affichées. Le nom de la classe stéréotypée est celui de la cible. Habituellement, une cible est le cadre d'une structure de cadre définie dans une fenêtre ; cependant, une cible peut être une toute nouvelle instance de navigateur : une fenêtre. Une association « targeted link » spécifie la cible où une page Web doit être affichée.

Contraintes : Pour chaque client du système le nom de la cible doit être unique. Par conséquent sur un même client, il ne peut exister qu'une seule instance d'une même cible.

Etiquettes : Aucune.

2.1.6. Objet Java Script « Java Script object »

Icône : Aucune.

Description : Sur un navigateur compatible Java Script, il est possible de simuler des objets personnalisés à l'aide de fonctions Java Script. Les objets Java Script ne peuvent exister que dans le contexte de pages client.

Contraintes : Aucune.

Etiquettes : Aucune

2.2. Association

2.2.1. Lien « link »

Icône : Aucune

Description : Un lien est un pointeur d'une page client vers une autre page. Dans un diagramme de classes, un lien est une association entre une page client et une autre page client ou une page serveur. A un lien correspond une balise ancre HTML.

Contraintes : Aucune.

Étiquettes : Paramètres (paramètres) : liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

2.2.2. Lien cible « targeted link »

Description : Similaire à une association lien. Un lien cible est un lien dont la page associée est affichée dans une cible. A un lien cible correspond une balise ancre HTML, dont l'attribut target prend la valeur de la cible.

Icône : Aucune.

Contraintes : Aucune.

Étiquettes : Paramètres (Paramètres) : liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

Nom de la cible (target name) : nom de la cible ou la page vers laquelle pointe le lien qui doit être affichée.

2.2.3. Contenu de cadre « frame content »

Icône : Aucune.

Description : Une association contenue de cadre est une association d'agrégation qui traduit l'appartenance d'une page ou d'une cible à un cadre.

Une association contenue de cadre peut aussi pointer vers une structure de cadre, aboutissant dans ce cas, à des cadres imbriqués.

Contraintes : Aucune.

Étiquettes : Rangée (Row) : entier qui indique la rangée du cadre dans la structure de cadre auquel appartient la page, ou la cible associée.

Colonne (Col) : entier qui indique la colonne du cadre dans la structure de cadre auquel appartient la page, ou la cible associée.

2.2.4. soumet « submit »

Icône : Aucune.

Description : submit est une association qui se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur, par l'intermédiaire de pages serveur, pour qu'il les traite. Le serveur Web traite la page serveur, qui accepte et utilise les informations du formulaire.

Contraintes : Aucune.

Étiquettes : Paramètres (parameters) : une liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

2.2.5. Construit « build »

Icône : Aucune.

Description : La relation « build » est une relation particulière qui fait le pont entre les pages client et les pages serveur. L'association « build » identifie quelle page serveur est responsable de la création d'une page client. C'est une relation orientée, puisque la page client n'a pas connaissance de la page qui est à l'origine de son existence.

Une page serveur peut construire plusieurs pages client, en revanche, une page client ne peut être construite que par une seule page serveur.

Contraintes : Aucune.

Étiquettes : Aucune

2.2.6. Redirige « redirect »

Icône : Aucune

Description : Une relation « redirect », est une association unidirectionnelle avec une autre page web, peut être dirigée à partir d'une page client ou serveur ou vers une page client ou serveur.

Contraintes : Aucune.

Étiquettes

Délai (delay) : délai que doit observer une page client avant de rediriger vers la page destination. Cette valeur correspond à l'attribut Content de la balise <META>

2.3. Attribut

2.3.1. Élément de saisie « input element »

Icône : Aucune.

Description : Un élément de saisie correspond à la balise <input> d'un formulaire HTML.. Les étiquettes associées à cet attribut stéréotypé, correspondent aux attributs de la balise <input>. Les attributs obligatoires de la balise HTML <input> sont renseignés de la manière suivante : l'attribut name prend la valeur du nom de l'élément de saisie et l'attribut value prend celle de sa valeur initiale.

Contraintes : Aucune.

Étiquettes

Type (Type) : Le type de l'élément de saisie : texte, numérique, mot de passe, case à cocher, bouton d'option, bouton SUBMIT ou bouton RESET.

Taille (size) : définit la largeur visible allouée à l'écran en caractères.

Longueur Max (Maxlength) : nombre maximal de caractères que peut saisir l'utilisateur.

2.3.2. Sélection d'éléments « select element »

Icône : Aucune.

Description : Contrôle de saisie employé dans les formulaires, il permet à l'utilisateur de sélectionner une ou plusieurs valeurs dans une liste. La plupart des navigateurs restituent ce contrôle par une liste d'options ou une liste déroulante.

Contraintes : Aucune.

Étiquettes

- **Taille (Size) :** définit le nombre d'éléments qui doivent être affichée simultanément.
- **Multiple (Multiple) :** valeur booléenne qui indique que plusieurs éléments peuvent être sélectionnés conjointement.

2.3.3. Zone de texte « texte area element »

Icône : Aucune.

Description : C'est un contrôle de saisie, employé dans les formulaires, qui permet l'écriture de plusieurs lignes de texte.

Contraintes : Aucune.

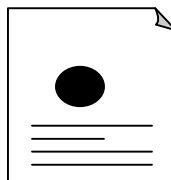
Etiquettes

- **Ligne (rows)** : nombre de lignes de texte visibles.
- **Colonnes (cols)** : largeur visible du texte en largeurs de caractères moyennes.

2.4. composant

2.4.1. page web « web page »

Icône



Description : Un composant page est une page web. Il peut être requis d'après son nom par un navigateur. Un composant page peut contenir des scripts client ou serveur.

Le plus souvent, le composant page est un fichier texte accessible au serveur Web, mais il peut également être un module compilé, chargé et exécuté par le serveur web. Dans les deux cas, le serveur Web produit, à partir du composant page, un document au format HTML, qui est renvoyé en réponse à la requête du navigateur.

Contraintes : Aucune

Etiquettes

Chemin (path) : chemin requis pour spécifier la page Web sur le serveur web. Cette valeur doit être relative au répertoire racine du site de l'application web.

3. Règles de cohérence sémantique

• Réalisation de composant :

En principe, les composants pages Web peuvent réaliser les classes stéréotypées « serveur page », « client page », « form », « JavaScript object », « clientScript object », « frameset » et « target ». Quand l'environnement de développement associé (ASP ou JSP) est en place, les pages web peuvent réaliser, au lieu des pages « server page », selon le cas, des classes stéréotypées « ASP page » ou « JSP page ».

• Généralisation

Tous les éléments de modélisation impliqués dans une même généralisation doivent être du même stéréotype.

• Association

Une page client peut avoir au plus une relation « build » avec une page serveur, mais une page serveur peut avoir plusieurs relations « build » avec différentes pages client. En plus de la combinaison standard d’UML, les combinaisons de stéréotypes présentées au Tableau ci-dessous sont permises.

| De \ A | « client page » | « server page » | « frameset » | « target » | « form » |
|-----------------|---|---|---|-----------------|------------------|
| « client page » | « link » « redirect » « targeted link » | « link » « redirect » « targeted link » | « link » « redirect » « targeted link » | « dependenc » | « aggregatio n » |
| « server page » | « build » « redirect » | « redirect » | « build » « redirect » | | |
| « frameset » | « frame content » | | « frame content » | « frame conte » | |
| « target » | | | | | |
| « Form » | « aggregated by » | « submit » | | | |

Tableau 1 : Combinaisons valides d’associations de stéréotypes.

VIII. Conclusion :

Cette annexe a proposé un survol et une présentation brève des principaux concepts de modélisation UML ainsi que leur notation.

Annexe B

Introduction

Qu'est ce qu'un thread ?

Système multi-tâches (multi-thread) : exécution concurrente de plusieurs tâches (Thread) sur un même processeur

Le système passe par un ordonnanceur de tâches (scheduler) qui attribue des temps d'exécution aux différentes tâches à tour de rôle selon une politique définie. L'activation d'une tâche est cyclique et dure un temps relativement bref, ce qui donne l'illusion du parallélisme.

Généralement, une application (java par exemple) correspond à un processus qui lui peut contenir un ou plusieurs threads.

Les threads en Java

Un programme Java (lancé via la JVM) lance un processus de base qui contient plusieurs threads: le thread principal (celui qui exécute le code à partir du main) et d'autres pour, par exemple, la gestion du "garbage collector".

En java, il est possible de développer ses propres threads afin de permettre la réalisation de plusieurs tâches en parallèle (de façon concurrentes).

Création et démarrage d'un thread.

Il existe, en Java, deux techniques pour créer un thread. Soit en dérivant un thread de la classe `java.lang.Thread`, Soit en implémentant l'interface `java.lang.Runnable`. Le choix de l'une ou l'autre des techniques dépend du contexte de l'implémentation. Par exemple, on choisit d'implémenter l'interface plutôt que la dérivation si notre thread doit hériter d'une autre classe (Ex/ cas de la classe montre vue en TD).

Créer un thread en dérivant de la classe `java.lang.Thread`.

La dérivation de la classe `java.lang.Thread` passe par au moins 1) la définition d'un constructeur et 2) la surcharge de la méthode `public void run()` héritée de `java.lang.Thread`.

```
public class FirstThread extends Thread {
private String threadName;          /** Un attribut propre à chaque thread : nom du
Thread */

public FirstThread(String threadName) {
```

```

this.threadName = threadName;
/** start est la méthode qui permet de lance le thread (héritée de java.lang.Thread)
**/

this.start() ; ---> ligne indispensable !!

}

/** Le but d'un tel thread est d'afficher 500 fois
 * son attribut threadName. Notons que la méthode
 * <I>sleep</I> peut déclencher des exceptions.
 */

public void run(){
try {
    for(int i=0;i<500;i++) {
        System.out.println("Thread nommé : " + this.threadName + " - itération : " + i);

/**/ sleep est une methode de la classe java.lang.Thread , qui met en veille le thread
pendant la durée indiquée par le paramètre en millisecondes***/

        Thread.sleep(30);
    }
} catch (InterruptedException exc) {
exc.printStackTrace();
}
}
}

```

On obtient ainsi une nouvelle classe FirstThread, qui permet de créer des objet tâches (Thread) et qui exécutent le code décrit dans la méthode run.

Comment lancer le thread, une fois créée?

Attention ! Il ne suffit pas pour lancer le thread, d'appeler la méthode run avec un objet instance de la classe thread créé.

Exemple (à ne pas faire)

```

....
    FirstThread t= new FirstThread ("First")
    t.run ()
.....

```

Ceci a pour conséquence d'exécuter la méthode (dans le thread courant) comme n'importe quelle autre méthode java.

Afin de lancer effectivement le Thread, il faut passer par la méthode public void start() héritée de la classe java.lang.Thread. Cette méthode permet de récupérer les ressources

nécessaires à l'exécution du thread, puis l'active (comme thread séparé du thread dans lequel il a été appelé). La méthode start, appelle la méthode run définie dans le Thread lancée.

```
/** Premier test de classe de thread en utilisant la
 * technique qui consiste à dériver de la classe Thread.
 */

/** Programme qui lance deux threads.
 * Chacun d'eux possède une données qui lui est
 * propre.
 */

static public void main(String argv[]) {
    FirstThread thr1 = new FirstThread("Toto");
    FirstThread thr2 = new FirstThread("Tata");
}
}
```

Remarque : avec cette technique on comme

- avantage :
 - autant d'objets que de threads, chaque thread a sa propre identité
 - possibilité d'avoir plusieurs classes différentes qui dérivent de java.lang.Thread, avec une méthode run spécifique à chacune
- inconvénients
 - le lien d'héritage (unique en java), est déjà utilisée pour Thread,. Il ne peut pas être utilisé pour une autre dérivation (Ex. La classe montre)

Créer un thread en implémentant l'interface java.lang.Runnable.

La seconde technique consiste donc à implémenter l'interface java.lang.Runnable. L'interface Runnable est en fait très simple dans le sens où elle ne définit qu'une unique méthode : la méthode *run*.

Pour créer un thread avec cette technique, on réutilise aussi la classe java.lang.Thread, mais pas en dérivation. Cette classe sera utilisée pour instancier un Thread, en utilisant le constructeur de Thread admettant l'interface Runnable comme paramètre .

```
package java.lang;

public class Thread extends Object {
    private Thread theThread = null;

    // ... du code ...

    public thread (Runnable theThread){
        this.theThread = theThread;
    }
}
```

```

    }

    // ... du code ...

    public void run() {
        if (this.theThread != null) {
            theThread.run();
        }
    }

    // ... du code ...
}

```

Par ce biais, un thread est initialisé avec l'objet implémentant l'interface Runnable, et la fonction start invoquera la méthode run (de l'objet de type Runnable) passé en paramètre au constructeur du Thread.

```

/** Second test de classe de thread en utilisant la
 * technique qui consiste à implémenter l'interface
 * java.lang.Runnable
 */

public class SecondThread implements Runnable {
    /** Un attribut partagé par tous les threads */
    private int counter;

    /** Démarrage de cinq threads basés sur un même objet */
    public SecondThread (int counter) {
        this.counter = counter;

        // On démarre cinq threads sur le même objet
        for (int i=0;i<5;i++) {
            (new Thread(this)).start() ;
        }
    }

    /** c'est la que s'établit le lien entre l'objet Runnable et le thread*/

    }

    /** Chaque thread affiche 500 fois un message. Un
     * unique compteur est partagé pour tous les threads.
     * Il y a cinq threads. Le dernier affichage devrait
     * donc être "Valeur du compteur == 2499".
     */

    public void run(){
        try {
            for(int i=0;i<500;i++) {
                System.out.println("Valeur du compteur== " + counter++);

                Thread.sleep(30);
            }
        }
    }
}

```

```

    }
    } catch (InterruptedException exc) {
exc.printStackTrace();
}
}

}

```

```

/** Le main créer un unique objet sur lequel vont se
 * baser cinq threads. Il vont donc tous les cinq se
 * partager le même attribut.
 */
static public void main(String argv[]) {
    SecondThread p1 = new SecondThread(0);
}
}

```

Remarque

Plusieurs Thread peuvent ainsi être initialisés avec un même objet. Nécessité de synchroniser les accès à l'objet dans certaines situations (cf. Synchronisation). Exemple d'implémentation d'une classe Clock permettant d'afficher l'heure graphiquement. Pour cela, on a besoin de dissocier le calcul de l'heure (comptage du temps écoulé toutes les secondes), de son affichage graphique (2 threads : le thread courant qui gère l'affichage et un thread que l'on crée pour changer le contenu à afficher à chaque seconde).

Cette classe dérive de la classe *java.awt.Label*. Cet objet label peut être ajouté à une interface graphique. En plus, cette classe fournit la méthode *run* requise par l'interface *Runnable*. Il ne reste plus qu'à lancer un objet de thread sur l'objet label pour pouvoir changer son contenu à chaque seconde.

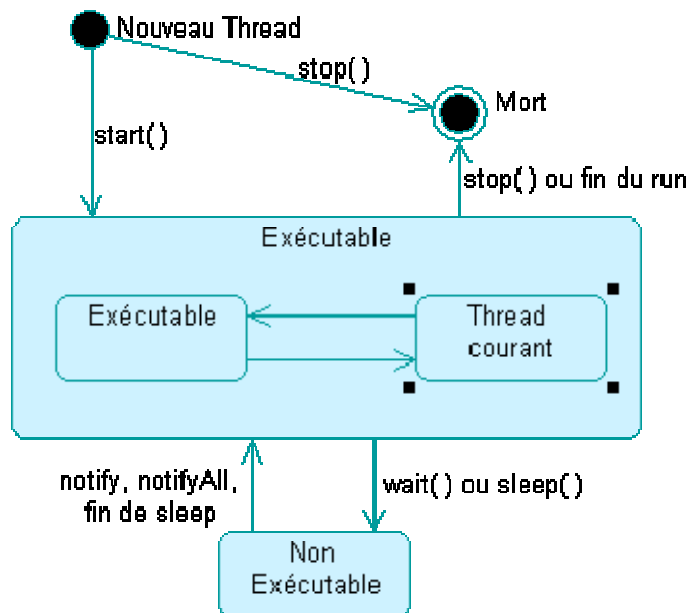
```

import java.util.*;
import java.text.*;
import java.awt.*;

public class Clock extends Label implements Runnable {
    private DateFormat timeFormat = DateFormat.getTimeInstance();
}

```


Un thread peut passer par différents états, tout au long de son cycle de vie. Le diagramme suivant illustre ces différents stades ainsi que les différentes transitions possibles.



Comme on le voit sur le schéma, un thread passe par les états : initial, transitions entre exécutable et non exécutable, et fin (mort).

A sa création, un thread est par défaut dans son état initial. Il faut invoquer la méthode *start* pour lui permettre de passer dans un état exécutable. Le thread passe à l'état mort soit à la fin de l'exécution de la méthode *run()*, soit à l'exécution de *stop()*. Un thread à l'état mort ne peut plus redémarrer.

L'état exécutable du thread lui permet de recevoir le processus et de s'exécuter un certain temps (quelques millisecondes) puis de laisser la main à un autre thread exécutable.

- Dans les systèmes préemptifs (systèmes actuels): Le thread n'a aucune action à faire pour laisser la main à un autre thread.
- Dans les système non préemptifs (plus rare actuellement) : le thread doit appeler la méthode *yield()* pour céder la main à un autre thread.

Un thread passe d'un état exécutable à un état non exécutable sous l'invocation de l'une de ces méthodes :

- *Thread.sleep(long durée)*, qui suspend le thread durant quelques instants, où
- les méthodes *wait* de la classe *java.lang.Object*. (cf. synchronisation)

Le passage à un état exécutable à nouveau, à partir d'un état non exécutable se fait soit :

- après l'écoulement du temps indiqué dans la méthode *sleep*
- soit après la réception d'une notification d'un autre thread (*notify()*, *notifyAll()*)

Démarrage, suspension, reprise et arrêt d'un thread.

Pour gérer l'exécution des threads, on dispose de différentes méthodes dont celles ci-dessous.

- **public void start()** : cette méthode permet de démarrer un thread. En effet, si on invoque la méthode *run* (au lieu de *start*), le code s'exécute bien, mais aucun

nouveau thread n'est lancé dans le système. Inversement, la méthode *start*, lance un nouveau thread dans le système dont le code à exécuter démarre par le *run*.

- **public void suspend()** : cette méthode permet d'arrêter temporairement un thread en cours d'exécution.
- **public void resume()** : celle-ci permet de relancer l'exécution d'un thread, au préalable mis en pause via *suspend*. Attention, le thread ne reprend pas au début du *run*, mais continue bien là où il s'était arrêté.
- **public void stop()** : cette dernière méthode permet de stopper, de manière définitive, l'exécution du thread. Une autre solution pour stopper un thread consiste à simplement sortir de la méthode *run*.

Gestion de la priorité d'un thread.

On peut, en Java, agir sur la priorité des threads. Sur une durée déterminée, un thread ayant une priorité plus haute recevra plus fréquemment le processeur qu'un autre thread. Il exécutera donc, globalement, plus de code.

La priorité d'un thread va pouvoir varier entre 0 et 10. Mais attention, il n'est en aucun cas garanti que le système hôte saura gérer autant de niveaux de priorités. Des constantes existent et permettent d'accéder à certains niveaux de priorités : MIN_PRIORITY (0) - NORM_PRIORITY (5) - MAX_PRIORITY (10).

L'exemple de code qui suit, lance trois threads supplémentaires. Chacun d'eux se voit affecter une priorité différente.

Quelque soit le thread considéré; celui-ci exécute un code très simpliste : il incrémente indéfiniment un compteur. Malgré cela, au bout d'un certain temps le thread initial stoppe les trois autres et l'on regarde le nombre d'incrémentations réalisé par chacun d'entre eux.

```
public class ThreadPriority extends Thread {
    private int counter = 0;

    public void run() {
        while(true) counter++;
    }

    public int getCounter() { return this.counter; }

    public static void main(String args[]) throws Exception {
        ThreadPriority thread1 = new ThreadPriority();
        ThreadPriority thread2 = new ThreadPriority();
        ThreadPriority thread3 = new ThreadPriority();

        thread1.setPriority(Thread.MAX_PRIORITY);
        thread2.setPriority(Thread.NORM_PRIORITY);
        thread3.setPriority(Thread.MIN_PRIORITY);

        thread1.start(); thread2.start(); thread3.start();
        Thread.sleep(5000);
        thread1.stop(); thread2.stop(); thread3.stop();

        System.out.println("Thread 1 : counter == " + thread1.getCounter());
        System.out.println("Thread 2 : counter == " + thread2.getCounter());
    }
}
```

```
System.out.println("Thread 3 : counter == " + thread3.getCounter());
}
}
```

Le tableau suivant montre les variations du nombre de constructions, en fonction des priorités affectées.

Au terme de votre analyse, on peut bien voir que la priorité de threads est un mécanisme de contrôle efficace.

| Thread 1 | Thread 2 | Thread3 |
|----------------------|----------------------|----------------------|
| Thread.NORM_PRIORITY | Thread.NORM_PRIORITY | Thread.NORM_PRIORITY |
| 250 752 409 | 256 697 243 | 251 964 807 |
| Thread.MIN_PRIORITY | Thread.NORM_PRIORITY | Thread.MAX_PRIORITY |
| 11 104 663 | 20 673 290 | 1 164 460 398 |

Gestion d'un groupe de threads.

Une autre possibilité intéressante consiste à regrouper différents threads. Dans un tel cas, on peut invoquer un ordre sur l'ensemble des threads du groupe, ce qui peut dans certains cas sérieusement simplifier le code.

Pour inscrire un thread dans un groupe, il faut que le groupe soit initialement créé.

Pour ce faire, il vous faut instancier un objet de la classe ThreadGroup.

Un fois le groupe créé, on peut attacher des threads à ce groupe, via un constructeur de la classe Thread.

```
public Thread(ThreadGroup group, String name)
```

```
ou public Thread(Runnable target,String name)
```

Par la suite, un thread ne pourra en aucun cas changer de groupe.

Une fois tous les threads attachés à un groupe, on peut alors invoquer les méthodes de contrôle d'exécution des threads sur l'objet de groupe. Les noms des méthodes sont identiques à la classe Thread : *suspend()*, *resume()*, *stop()*, ...

Synchronisation de threads et accès aux ressources partagées.

Lorsque qu'on lance une JVM (Machine Virtuelle Java), on lance un processus. Ce processus possède plusieurs threads et chacun d'entre eux partage le même espace mémoire. En effet, l'espace mémoire est propre au processus et non à un thread. Cette caractéristique est à la fois un atout et à la fois une contrainte. En effet, partager des données pour plusieurs threads est, par définition, relativement simple. Par contre les choses peuvent se compliquer sérieusement si la ressource (les données) partagée est accédée en modification : il faut synchroniser les accès concurrents.

Pour mieux comprendre les choses, imaginons que deux threads cherchent à modifier (à incrémenter) l'état d'un attribut statique, de type entier, nommé *MaClasse.shared*. Seul un

thread à la fois peut exécuter du code, mais n'oublions pas que les systèmes les plus modernes sont préemptifs ! De plus une instruction telle que *MaClasse.shared = MaClasse.shared + 1*; se traduit en plusieurs instructions en langage machine. Imaginez qu'un premier thread évalue l'expression *MaClasse.shared + 1* mais que le système lui hôte le cpu, juste avant l'affectation, au profit d'un second thread. Ce dernier se doit lui aussi d'évaluer la même expression. Le système redonne la main au premier thread qui finalise l'instruction en effectuant l'affectation, puis le second en fait de même. Au final de ce scénario, l'entier aura été incrémenté que d'une seule et unique unité. De tels scénarios peuvent amener à des comportements d'applications chaotiques. Il est donc vital d'avoir à notre disposition des mécanismes de synchronisation.

Notions de verrous.

L'environnement Java offre un premier mécanisme de synchronisation : les verrous (locks en anglais). Chaque objet Java possède un verrou et seul un thread à la fois peut verrouiller un objet. Si d'autres threads cherchent à verrouiller le même objet, ils seront endormis jusqu'à que l'objet soit déverrouillé. Cela permet de mettre en place ce que l'on appelle plus communément une section critique.

Pour verrouiller un objet par un thread, il faut utiliser le mot clé **synchronized**. En fait, il y a deux façons de définir une section critique :

- soit on synchronise un ensemble d'instructions sur un objet, dans ce cas on utilise l'instruction *synchronized*.

```
Synchronized(object){  
    // Instructions de manipulation d'une  
    // ressource partagée.  
}
```

- soit on synchronise directement l'exécution d'une méthode pour une classe donnée, en utilisant le qualificateur *synchronized* sur la méthode considérée.

```
Public synchronized void meth(int param)           {  
    // Le code de la méthode synchronisée.  
}
```

Afin de mieux comprendre les choses, nous allons étudier un petit exemple : celui-ci va permettre à plusieurs threads de tracer des pixels en parallèle. Mais attention, pour que le programme se déroule normalement, il faut que l'appel à la méthode de tracé soit synchronisé. Sinon, il y aura des possibilités pour que certains pixels soient sautés.

L'interface graphique de l'applet présente deux boutons et une zone de dessin. Les deux boutons permettent de lancer le tracé dans la zone de dessin. Mais l'un des boutons lance le tracé sans synchronisation alors que l'autre le fait en synchronisant les threads. Pour localiser le code mettant en œuvre la synchronisation, regardez le code de la méthode *run*.

```
import java.applet.* ;
import java.awt.* ;
import java.awt.event.* ;

public class Synchronized extends Applet
    implements ActionListener, Runnable {
    private Panel pnlButtonBar = new Panel() ;
    private Button btnStartNormal = new Button(« Non synchronisé ») ;
    private Button btnStartSynchronized = new Button(« Synchronisé ») ;
    private Canvas cnvGraphic = new Canvas() ;
    private Label lblStatus = new Label(« Choisissez un mode d'exécution ») ;

    private boolean drawMode = false ;
    private int x, y ;

    public void init() {
        pnlButtonBar.setLayout(new FlowLayout()) ;
        pnlButtonBar.add(btnStartNormal) ;
        pnlButtonBar.add(btnStartSynchronized) ;
        btnStartNormal.addActionListener(this) ;
        btnStartSynchronized.addActionListener(this) ;

        this.setLayout(new BorderLayout()) ;
        this.add(pnlButtonBar, BorderLayout.NORTH) ;

        cnvGraphic.setBackground(Color.white) ;
        this.add(cnvGraphic, BorderLayout.CENTER) ;

        this.add(lblStatus, BorderLayout.SOUTH) ;
    }

    public void plot() {
        this.cnvGraphic.getGraphics().drawLine(this.x,this.y,this.x,this.y) ;
        if (++this.x >= 300) { this.x=0 ; this.y++ ; }
    }

    public void run() {
        while(this.y < 100) {
            if (drawMode) {
                synchronized(this) { this.plot() ; }
            } else {
                this.plot() ;
            }
        }
    }

    public void actionPerformed(ActionEvent event) {
        this.drawMode = (event.getSource() == btnStartSynchronized) ;
    }
}
```

```
this.cnvGraphic.repaint() ;
this.x = this.y = 0 ;

(new Thread(this)).start() ;
(new Thread(this)).start() ;
(new Thread(this)).start() ;
}
}
```

Attendre l'accès à une ressource.

Mais poser des verrous ne suffit pas toujours. Dans certains cas, vos threads doivent patienter (pour ne pas consommer trop de temps CPU) avant qu'une ressource ne soit disponible. Pour gérer ces cas, l'environnement Java propose aussi un support pour pouvoir contrôler l'activité de vos threads. Il vous est possible de stocker (un tableau d'éléments de type *java.lang.Object*) des threads endormis sur un objet dans le but de la manipuler (attention, cela n'a rien à voir avec les verrous).

Tout le support nécessaire à cette gestion est fourni dans la classe *Object*. Celle-ci propose notamment quatre méthodes permettant d'endormir un thread, ainsi que de le réveiller. Pour de plus amples informations, vous pouvez toujours consulter [la documentation de l'API du JDK](#).

Pour endormir un thread sur le moniteur, il vous faut utiliser la méthode *wait*. Plusieurs prototypes vous sont fournis afin de pouvoir attendre indéfiniment soit durant un délai maximal. Il est vrai que la méthode *Thread.sleep* permet aussi de faire patienter un thread, mais il sera alors impossible de faire reprendre l'activité du thread avant la fin du timeout. Par contre, la méthode *Object.wait* le permet.

Pour réveiller des threads endormis, vous pouvez utiliser les méthodes *notify* et *notifyAll*. Soit vous choisissez de réveiller un unique thread endormi sur un objet sur lequel il faut se synchroniser, soit vous décidez de tous les réveiller.

Exemple de producteurs/consommateurs.

Pour mettre en œuvre un exemple de synchronisation un peu évolué, nous allons considérer un cas d'école : un exemple de producteurs/consommateurs. Dans un tel cas, une ressource partagée est utilisée par des threads qui produisent et par des threads qui consomment. Mais attention, il est hors de question de consommer si rien n'a été produit.

Pour implémenter notre objet partagé, nous allons coder une pile (Stack en anglais). On considère une pile bornée, sur laquelle on ne peut donc pas empiler indéfiniment. Pour empiler des données supplémentaires, il faudra attendre que des threads consommateurs aient dépilé les anciennes données.

Dans ce cas, les choses sont subtiles. L'objet de synchronisation est clairement la pile. Nous allons donc, au gré de l'exécution du programme, endormir des threads sur cet objet. Mais deux types de threads seront à considérer : ceux qui produisent et ceux qui consomment.

Imaginons le scénario suivant : un thread empile une donnée. Il va donc utiliser une méthode pour réveiller un éventuel consommateur. Mais qu'est ce qui garantit que le thread réveillé ne sera pas un autre producteur ? Si c'était le cas, nous aboutirions à des cas d'inter-blocage : le programme n'évoluerait plus, mais ne terminerait pas. Il nous faudra donc utiliser la méthode *notifyAll* pour réveiller les threads. Il faut alors garantir que tous les threads réveillés qui n'ont pas accès à la ressource se rendorment rapidement. D'où le code des méthodes *push* et *pop* de la classe suivante.

```
public class Pile {
    private int array [];
    private int size, index;

    /** Un constructeur de pile */
    public Pile() {
        this (5);
    }

    /** Un autre constructeur de pile qui prend la taille de cette dernière */
    public Pile(int size) {
        this.size = size;
        this.index = 0;
        this.array = new int[size];
    }

    /** Renvoie true si la pile est vide */
    public boolean isEmpty() { return index == 0; }

    /** Renvoie true si la pile est pleine */
    public boolean isFull() { return index == size; }

    /** Cette méthode synchronisée permet de dépiler une valeur */
    public synchronized int pop () {
        try {
            while (isEmpty()) {
                System.out.println("Consommateur Endormi");
                wait();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        int val = array[--index];
        notifyAll();
        return val;
    }

    /** Cette méthode synchronisée permet d'empiler une valeur */
    public synchronized void push(int value) {
        try {
            while (isFull()) {
                System.out.println("Producteur Endormi");
                wait();
            }
        }
    }
}
```

```

    } catch(Exception e) {
        e.printStackTrace();
    }

    array[index++] = value;
    notifyAll();
}
}

```

Maintenant que notre ressource partagée est prête, il ne nous reste plus qu'à coder nos producteurs et nos consommateurs. Dans les deux cas, ces deux types (classes) de composants partagent tous certaines caractéristiques : ils travaillent tous sur la même pile et dans les deux cas, cadencer les choses via un `Thread.sleep` pourra permettre une bonne lisibilité des résultats sur la console. Nous utilisons ici l'héritage pour définir le tronc commun à tous nos threads. La classe de base se nomme *Fonctionneur*, et elle est de plus abstraite : nous ne voulons pas permettre d'instancier un quelconque objet de ce type là (de plus, nous n'avons pas d'implémentation de la méthode *run*).

```

abstract class Fonctionneur implements Runnable {
    /** La pile partagée par tous nos threads */
    private static Pile p = new Pile(5);
    /** Un délai d'attente pour chaque thread */
    protected long sleepTime;

    /** Un constructeur par défaut. Pas de délai d'attente */
    public Fonctionneur () {
        this (0);
    }

    /** Un constructeur qui prend un délai d'attente pour le thread */
    public Fonctionneur (long sleepTime) {
        this.sleepTime = sleepTime;
        (new Thread(this)).start();
    }

    /** Permet de pouvoir récupérer la pile */
    public Pile getPile() { return p; }
}

```

Nous pouvons maintenant dériver de cette classe nos deux types de threads : les producteurs (classe *Producteur*) qui vont produire des valeurs entières et donc les empiler (tant que cette dernière n'est pas pleine) et les consommateurs (class *Consommateur*) qui vont lire des valeurs entières sur la pile (tant que cette dernière n'est pas vide).

```

public class Producteur extends Fonctionneur {

    private int value = 0;

    public Producteur() { super(); }

    public Producteur(long sleepTime) {
        super(sleepTime);
    }
}

```

```

    }

    public void run() {
        while (true) {
            try {
                Thread.sleep ((long)Math.random()*this.sleepTime);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            System.out.println (this + " empile " + value);
            getPile().push(value++);
        }
    }
}

public class Consommateur extends Fonctionneur {

    public Consommateur() { super(); }
    public Consommateur(long sleepTime) {
        super(sleepTime);
    }

    public void run() {
        while (true) {
            try {
                Thread.sleep ((long)Math.random()*this.sleepTime);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            System.out.println (this + " depile " + getPile().pop());
        }
    }
}

```

Maintenant il ne reste plus qu'à coder une classe de démarrage afin d'initier le démarrage des threads à synchroniser sur la pile.

```

public class Start {
    static public void main (String args[]) {
        // Démarrage de deux Producteurs
        new Producteur(1000);
        new Producteur(1000);
        // Démarrage de deux Consommateurs
        new Consommateur(1000);
        new Consommateur(1000);

        while (true) { // Mise en veille du thread principal
            try {
                Thread.sleep (10000L);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

```
}  
}
```

Conclusion

Nous venons donc de voir qu'en Java, il est possible de créer des threads. Deux techniques vous sont proposées. Chacune d'entre elle ayant des avantages et des inconvénients (nous avons dans ce chapitre, volontairement un peu utilisé les deux techniques). De plus nous avons vu qu'il été possible de contrôler, plus ou moins, finement l'exécution de vos threads. Il existe même la notion de groupes de threads permettant une gestion en lots de threads.

Mais comme dans tout langage, s'il n'était pas possible de synchroniser les accès concurrents aux ressources partagées, cette solution serait trop souvent inutile. En conséquence, la notion de verrous (sections critiques) et un mécanisme de synchronisation (mise en sommeil et réveil de threads) vous sont proposés.

Bibliographie

Bibliographie

[Azzi, 1995]. AZZI, R. Environnement de développement de simulateurs pédagogiques de procédés industriels. Thèse de doctorat en informatique. Lyon : INSA de Lyon, 1995, 212p

[Akkouche I. 1996]. Télé-enseignement: formation professionnelle à distance et formation coopérant. Thèse de doctorat, INSA de Lyon.

[Arpaia, P., Baccigalupi, A., Cennamo, F. & Daponte, P. (2000)]. A Measurement Laboratory on Geographic Network for Remote Test Experiments. *IEEE Transactions on Instrumentation and Measurement*, 49(5), 992-997.

[Ben, 2007] Benmohamed H « ICTT@Lab: Un environnement informatique pour la génération et l'exécution de scénarios de télé-TP ». Thèse de doctorat de l'INSA de Lyon.

[Ben, 2004] Benmohamed H, Lelevé A, Prevot P «Remote Laboratories: New Technology and Standard Based Architecture», première conférence internationale sur les technologies d'information et de communication(ICTTA 04)(Information & Communication Technologies: from Theory to Applications), Damas, Syrie, Avril 2004.

[Buitargo, 1999]. BUITARGO, G. C. Simulation et contrôle Pédagogique : Architectures Logicielles Réutilisables. Thèse de doctorat en informatique. Grenoble : Université Joseph Fourier-Grenoble I, 1999, 269p.

[Chen et al., 1999]. CHEN, S. H., RAMAKRISHNAN, V., HU, S. Y., ZHUANG, Y., KO, C. C. , CHEN, B. M. Development of remote Laboratory Experimentation through Internet. Proceedings of Symposium on robotics and control, juillet 1999, Hong Kong, Chine, pp. 756-760.

[Coop, 2002] Cooper M « Remote controlled experiments for teaching over the Internet : a comparison of approaches developed in the PEARL Project» Conférence ASCILITE(Australian Society for Computers in Learning in Tertiary Education), Auckland, Nouvelle Zélande, du 08 au 12 Décembre 2002.

[Deniz et al., 2003]

DENIZ, D. Z., BULANCAK, A. , ÖZCAN, G. A novel approach to remote laboratories. 33d ASEE/IEEE Frontiers in Education Conference, 5-8 Novembre 2003, Boulder, CO, pp. T3E8 à T3E12.

[Dixon et al., 2002]. DIXON, W. E., DAWSON, D. M., COSTIC, B.T., QUEIROZ, M. S. D. A MATLAB-Based control Systems laboratory Experience for undergraduate Students; Towards Standardisation and Shared Ressources. *IEEE Transactions on Education*, 2002, vol. 45, n°3, pp. 218-126.

[Lelevé, 2004] Lelevé A, Benmohamed H, Prevot P, Benadi M «Generic e-lab platforms and e-learning standards », actes de la conférence (CALIE 04)(Computer Aided Learning In Engineering Education) Grenoble, France du 16 au 18 Février 2004.

[Lelevé1, 2005] Lelevé A, Benmohamed H, Prevot P «Implémentation d'une chaîne d'édition générique pour télé-TPs ».actes de la conférence sur la robotique pédagogique.

[Lelevé2, 2003] Lelevé A, Benmohamed H, Prevot P, Meyer C « Remote Laboratory Towards an integrated training system ». Quatrième conférence internationale sur l'éducation et la formation basées sur les technologies de l'information (ITHET03)(Information Technology Based Higher Education and Training) Marrakech, Maroc du 7 au 9 juillet 2003 .

[Lelevé3, 2002] Lelevé A, Meyer C, Prevot P « Télé-TP: premiers pas vers une modélisation » Actes du Symposium on Technology of Information and Communication in education for engineering and industry, Lyon, p. 203-211.

[Lelevé4, 2003] Lelevé A, Prevot P, Subai C, Noterman D, Guillemot M. « Toward remote laboratory platforms with dynamic scenarios » , multi-conférence en systémique, cybernétique et informatique (SCI2003), Juillet 2003 ,Orlando, Floride, USA

[Lelevé, A., Benmohamed, H. & Prévôt, P. (2005, mai)]. *Implémentation d'une chaîne d'édition générique pour télé-TP.* Communication au 8ème colloque francophone de robotique pédagogique, Le Mans.

[Lelevé, A., Benmohamed, H. & Prévôt, P. (2004, septembre)]. *Sharing a System between Simultaneous Learners in Remote Laboratories.* Communication au 2nd IFAC Workshop on Internet Based Control Education (IBCE), Grenoble.

[Leroux, 2002] Leroux P « Machines partenaires des apprenants et des enseignants, Etude dans le cadre d'environnements supports de projets pédagogiques » Mémoire présenté pour l'obtention de l'Habilitation à Diriger des Recherches en Informatique(HDR),LIUM Université du Maine, France.

[Moura, 2007] Moura C « MDEduc : conceiving and implementing a language-oriented approach for the design of automated learning scenario » Thèse de doctorat de l'Université des sciences et technologies de Lille, 2007.

[Nedic *et al.*, 2003]

NEDIC, Z., MACHOTKD, J. , NAJHLSK, A. Remote laboratories versus virtual and real laboratories. 33rd ASEE/IEEE Frontiers in Education Conference, 5-8 Novembre 2003, Boulder, CO, États-Unis d'Amérique, pp. T3 E1 à T3 E6.

[Paqu, 2002] Paquette G « L'ingénierie pédagogique. Pour construire l'apprentissage en réseau ». Presses de l'Université du Québec.

[Paqu, 2004] Paquette G « Instructional engineering for learning objects repositories networks », 2nd International Conference on Computer Aided Learning in Engineering Education. Grenoble (France), p 25-36.

[Perin, 1996]. PERIN, J. -P.M.A.R.S. Un modèle opérationnel de conception de simulation pédagogiques. Thèse de doctorat en informatique. Grenoble : Université Joseph Fourier – Grenoble I, 1996, 276p.

[Ram, 2011] Mohamed Ramdane, Rachid Ahmed-Ouamer. Un Modèle Générique pour les Travaux Pratiques à Distance. Laboratoire de Recherche en Informatique LARI, Département d'Informatique Université Mouloud Mammeri de Tizi-Ouzou, 15000 Tizi-Ouzou, Algérie.

[Remi 2006] Remigiusz J, Godziemba-Maliszewski M, Majkowski A «A proposal of virtual laboratory structure» Conférence IMTC Sorrento, Italie 24-27 Avril 2006

[Saad et al., 2001]. SAAD, M., SALIAH-HASSANE, H., HASSAN, H., EL-GUETIOUI, Z. , CHERIET, M. A Synchronous Remote Accessing Control laboratory on the internet. International Conference on Engineering Education, 6-10 Août, 2001, Oslo, Norvège, pp. 8D1-30 à 8D1-33.

[Sad1, 2006] Sadou M, Ahmed-Ouamer R « Support aux activités individuelles et collectives en e-learning. » Colloque Euro Méditerranéen et Africain d'Approfondissement sur la Formation A Distance : CEMAFORAD 3, Sousse, Tunisie 11-13 nov. (2006), 12 p.

[Sad2, 2006] Sadou M, Ahmed-Ouamer R « Support au tutorat en ligne. » Colloque International sur l'Optimisation et les Systèmes d'Information » : COSI'06, Alger, Algérie 11-13 juin (2006), p. 162-173

[Learn]

<http://www.demos.fr/lexique.asp>

[Moodle]

<http://www.moodle.org>

[Papert]

<http://fr.wikipedia.org/wiki/Micromonde>

[SCORM]

<http://www.adlnet.org/>

[Spiral]

<http://spiral.univ-lyon1.fr/>

[WebCT]

<http://www.webct.com/>

Le Tannenbaum système