



# Vers une implémentation d'un Système d'Enseignement à Distance basé sur le filtrage

Boubker Sbihi

► **To cite this version:**

Boubker Sbihi. Vers une implémentation d'un Système d'Enseignement à Distance basé sur le filtrage. La revue RIST :revue de l'information scientifique et technique, 2005, 15 (1). <sic\_00001546>

**HAL Id: sic\_00001546**

**[https://archivesic.ccsd.cnrs.fr/sic\\_00001546](https://archivesic.ccsd.cnrs.fr/sic_00001546)**

Submitted on 18 Sep 2005

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## 1. Introduction

Le mariage de l'informatique et des télécommunications dans le domaine de la formation a permis d'abolir les distances, de disposer des compétences outre que nationales et de faciliter l'accès au savoir. Actuellement et dans le cadre de la mondialisation, il y a émergence du marché mondial de la formation avec l'apparition de nombreuses sociétés internationales qui, à travers les nouvelles technologies de l'information et de la communication, se sont lancées dans le développement de la formation à la demande. En effet, l'enseignement à distance permet également d'optimiser les ressources humaines et contribue à l'ouverture de l'université sur son environnement national et international. Au Maroc, et, afin d'être au diapason des nouvelles mutations dans le domaine de la formation. Cela a poussé la mise en place d'un campus virtuel marocain sur la base des expériences des universités marocaines et à la lumière des expériences d'universités étrangères. Notre collaboration en tant qu'acteurs de l'enseignement supérieur au Maroc a été concrétisée par le projet de Formation Continue en Information Informatisée en Réseau (FORCIIR) [6]. Ce projet a pour objectif le développement de l'enseignement des nouvelles technologies dans les écoles des sciences de l'information en Afrique. Notre collaboration a touché aussi le volet recherche sur l'enseignement à distance [14] afin de modéliser et d'implémenter un système d'enseignement à distance à travers le mécanisme de filtrage ; il a fait également l'objet de certaines de nos recherches [7] [12] [13]. La modélisation d'un système d'enseignement à distance ne peut pas être réalisée selon un point de vue unique en raison des besoins différents et droits d'accès aux informations spécifiques à chaque utilisateur. La technologie objet classique a démontré sa puissance expressionnelle, mais ne permet pas de résoudre les problèmes d'incohérence et de redondance liés à ce type de modélisation. L'introduction de la notion de point de vue dans la modélisation orientée objet des systèmes complexes permet d'élaborer un modèle unique partageable accessible suivant plusieurs points de vue [5]. Par ailleurs, plusieurs solutions d'implémentation des points de vue ont été proposées [3] notamment dans la programmation par sujets [11], la programmation par aspects [2], et enfin la programmation par vues [9]. L'architecture basée sur les modèles MDA (Model-Driven Architecture)[8] conçue et promue par l'OMG afin de séparer les contraintes fonctionnelles des contraintes techniques. D'autres solutions ont utilisé des extensions des modèles UML [4]. L'objectif de ce papier est de modéliser et d'implémenter purement en UML, la notion de point de vue dans un système d'enseignement à distance à l'aide du mécanisme de filtrage afin de cibler différents types de codes de langages orientés objets. Dans la section suivante, nous présentons le système d'enseignement à distance. Dans la section 3, nous illustrerons la visibilité UML puis nous décrirons le mécanisme de filtrage dans la section 4. Ensuite, la section

5 présente l'application du mécanisme de filtrage pour le SED. Finalement, la section 6 conclut cet article en donnant un ensemble de perspectives.

---

## 2. Système d'enseignement à distance

Nous illustrons ci-après les spécifications du Système d'Enseignement à Distance. Un tel système est décentralisé sur plusieurs sites, chaque site étant géré par un responsable de site. La politique du SED est menée par un directeur et un conseil pédagogique qui regroupe le directeur, les responsables de sites, les représentants des enseignants et ceux des étudiants. Pour simplifier, nous considérerons que le SED permet à des étudiants à distance de suivre des cours en temps différé. Parmi les acteurs du SED, nous citons : Les étudiants qui s'inscrivent et suivent des cours à distance, passent des examens, etc. Les enseignants qui produisent et assurent des cours, réalisent un suivi des étudiants, etc. Les responsables de sites qui ajoutent de nouveaux cours dans la liste des cours disponibles, gèrent les inscriptions des étudiants, gèrent les résultats aux examens, etc. Le conseil pédagogique qui fait le bilan régulièrement, définit la stratégie, traite les litiges et les cas particuliers, etc. Le directeur du SED qui analyse les orientations pédagogiques du conseil pédagogique, produit un bilan global de fonctionnement du système, définit la stratégie globale, etc.

La liste des cours disponibles est décrite dans un catalogue accessible sur Internet. Les inscriptions se font généralement en début d'année. Un cours est caractérisé par un enseignant responsable, un numéro d'identification unique, un titre, un nombre de crédits (équivalence ECTS<sup>1</sup>), un niveau de difficulté, un contenu (transparentes, vidéo, diapositives...) disponible sur le serveur, une liste d'exercices, un indice qui reflète l'importance du domaine traité par le cours, un prix à payer lors de l'inscription (déduit de l'indice et du nombre de crédits affectés au cours), une liste d'étudiants inscrits, et un examen qui se passe via internet. Un exercice est associé à un cours. Il est caractérisé par un niveau de difficulté, une liste d'exercices pré requis et une solution. Un étudiant s'inscrit à des cours en temps différé, dans un site quelconque. Pour chaque cours auquel il est inscrit, il peut envoyer des questions par email à l'enseignant responsable de ce cours qui doit répondre dans les 3 jours. Il peut aussi faire des remarques sur ce cours. Concernant les examens, un étudiant suivant un cours doit s'inscrire à son examen à une date qu'il choisit dans la liste des dates possibles. Dès lors qu'il s'est inscrit, le fait de ne pas passer l'examen équivaut à une note égale à zéro au cours. Un examen d'un cours se déroule en temps limité via Internet, en présence distante du responsable du cours qui peut répondre à des questions de compréhension du sujet.

---

<sup>1</sup> European Credit Transfer System

Pour simplifier, nous considérerons que le SED permet à des étudiants à distance de suivre des cours en temps différé et que parmi les acteurs du SED, nous retenons : les étudiants, les enseignants et les responsables. Afin de réaliser un tel système objet, nous sommes obligés de formuler notre solution selon la norme UML standard de fait dans le domaine de l'orienté objet.

---

### 3. Visibilité l'UML

Le langage de modélisation unifié UML [15] standardisé par l'OMG (Object Management Group) [10] est le résultat de la fusion de plusieurs méthodes orientées objets et est devenu la référence en terme de modélisation objet. Il couvre l'aspect statique et dynamique d'un système selon ses différents diagrammes. Son but est de spécifier, visualiser, construire, et documenter les systèmes informatiques dont les systèmes complexes. Pour cela, il définit neuf diagrammes qui sont subdivisés en des vues statiques (qui représentent "physiquement" le système à modéliser au moyen de diagrammes d'objets, de classes, de cas d'utilisation, de composants et de déploiement) et des vues dynamiques (qui montrent le fonctionnement du système au moyen de diagrammes de séquence, de collaboration, d'états transitions et d'activités). Le diagramme de classes est le modèle de base pour l'implémentation d'un logiciel du moment qu'il peut directement être traduit dans les différents langages de programmation tel que Java ou C++. Le modèle objet qui constitue les briques du logiciel peut être directement déduit à partir du modèle de classes. De plus, il existe plusieurs ateliers de génie logiciel (Rational Rose, Visio,...) qui prennent en charge les diagrammes de classes UML et génèrent ensuite le code des applications orientés objets. Chaque classe UML possède un ensemble d'attributs (état) et un ensemble de méthodes (comportement). Le langage de modélisation UML implémente la notion de visibilité en distinguant trois niveaux de visibilité sur les primitives. La signification et la notation UML de ces visibilités sont définies dans le tableau suivant (Tableau 1) :

Notation	Signification
- <b>privé</b>	La primitive n'est visible que dans la classe
+ <b>public</b>	La primitive est visible par toutes les autres classes
# <b>protégé</b>	La primitive est visible par la classe et ses sous classes

**Tableau 1.** Niveaux de visibilité UML et leur signification

Ces trois niveaux sont insuffisants dans le cas d'un système vu selon différents points de vue, c'est-à-dire contenant des objets qui ne sont pas vus de la même façon par tous les utilisateurs. La distinction de trois niveaux de visibilité est très restreinte pour gérer les points de vue. Aussi, le changement dynamique des visibilités est

impossible durant l'exécution, par exemple un attribut de visibilité publique UML le restera toujours durant l'exécution. De plus, si on associe un niveau de visibilité à un attribut d'un objet dans un système, alors tous les utilisateurs sont obligés de voir cet objet à partir du même angle, indépendamment du point de vue de chacun de ces utilisateurs. Pour palier ce problème, on a défini un mécanisme de filtrage basé sur l'UML donnant la possibilité de changer les visibilités des primitives dynamiquement. Pour remédier aux problèmes de l'incapacité de la visibilité offerte par UML à supporter l'aspect dynamique, nous avons défini un mécanisme de filtrage.

#### 4. Mécanisme de filtrage

Le mécanisme de filtrage [12] [13] permet de filtrer les services offerts par une classe dépendamment du point de vue de l'utilisateur demandant ces services. En effet ce mécanisme, basé sur la visibilité UML, permet de définir une visibilité sur une primitive d'une classe selon des points de vue différents. Il nous permet de changer la visibilité dynamiquement avec le temps et qui peut donner à un attribut la possibilité de lecture à certains points de vue, la modification pour d'autres et masquer sa valeur pour le reste. Dans ce contexte, la notion de point de vue consiste à faire la distinction entre les points de vue des différents utilisateurs vis-à-vis le système à modéliser. Elle apparaît essentiellement au niveau des droits d'accès de chacun de ces utilisateurs aux différentes entités du système. La visibilité d'un utilisateur par rapport à une primitive détermine le niveau d'accès de cet utilisateur sur cette primitive en ce qui concerne la possibilité de son utilisation. Le mécanisme de filtrage cible les primitives comme une unité de travail. Il leur définit une nouvelle visibilité permettant ainsi la gestion des droits d'accès. Dans ce nouveau contexte, un attribut pourra être ou bien inaccessible, accessible en lecture, ou accessible en écriture selon les points de vue. Le même principe est appliqué pour les méthodes puisqu'il interdit ou autorise l'accès selon les points de vues. Ainsi il alloue trois types de visibilité aux attributs et deux types de visibilité pour les méthodes illustrées dans le tableau ci-dessous (Tableau 2) :

Type de Visibilité Attribut	Valeur Visibilité
Ni lecture, ni modification	1
Lecture seule	2
Lecture et modification	3
Type de Visibilité Méthode	Valeur Visibilité
Non accessible	4
Accessible	5

Tableau 2. Types et Valeurs de visibilité pour les primitives

## 5. Modélisation du SED par l'UML

Au cours des dernières années, la venue de l'informatique dans le monde des communications a permis d'assister l'encadrement pédagogique des étudiants dans un contexte d'enseignement à distance. En effet, l'enseignement à distance (EAD) utilisant les nouvelles technologies s'adresse à des étudiants de tous les niveaux.

L'enseignement à distance s'adresse soit à des personnes qui exercent déjà une profession dans un but de perfectionnement (le cas de la formation continue), soit à celles qui se forment dans une spécialité (Formation diplômante).

Cinq paramètres technologiques viennent revendiquer leur place dans l'enseignement actuel, notamment aux niveaux supérieurs et en formation continue: numérisation, information, communication, hypertextualité et interactivité. Parmi les avantages de L'EAD figurent :

Côté étudiant	Côté Enseignant	Côté gouvernement
Travailler à son propre rythme	Eviter les déplacements	Economiser les réorientations
Choisir le temps de travaille (Soir, Matin,..)	Favoriser l'excellence des enseignements	Faciliter les réorientations
Eviter les déplacements	Etre relayé par un tuteur	
Etre accompagné par un tuteur	Nouvelle approche de l'enseignement	
Se former sur place		
S'auto-évaluer		
Nouvelle approche de l'enseignement		
Disponibilité psychologique.		

**Tableau 3.** *Avantages de l'EAD*

La formation en technologies de l'information est une formation professionnelle ouverte aux diplômés de l'enseignement supérieur titulaires d'un Bac + 4 déjà en

activité ou à la recherche d'un emploi. Cette formation théorique et pratique porte sur les techniques de traitement de l'information, et de sa recherche et diffusion à travers Internet. Elle est sanctionnée par un Certificat ou un DESS en Gestion ou en informatique. Nous illustrons les spécifications du Système d'Enseignement à Distance. Un tel système est décentralisé sur plusieurs sites, chaque site étant géré par un responsable de site. La politique du SED est menée par un directeur et un conseil pédagogique qui regroupe le directeur, les responsables de sites, les représentants des enseignants et ceux des étudiants. Pour simplifier, nous considérerons que le SED permet à des étudiants à distance de suivre des cours en temps différé. Parmi les acteurs du SED, nous citons :

- Les étudiants qui s'inscrivent à des cours, suivent des cours à distance, passent des examens, etc.
- Les enseignants qui produisent des cours, assurent des cours, réalisent un suivi des étudiants, etc.
- Les responsables de sites qui ajoutent de nouveaux cours dans la liste des cours disponibles, gèrent les inscriptions des étudiants, gèrent les résultats aux examens, etc.
- Le conseil pédagogique qui fait le bilan régulièrement, définit la stratégie, traite les litiges et les cas particuliers, etc.
- Le directeur du SED qui analyse les orientations pédagogiques du conseil pédagogique, produit un bilan global de fonctionnement du système, définit la stratégie globale, etc.

La liste des cours disponibles est décrite dans un catalogue accessible sur Internet. Les inscriptions se font généralement en début d'année. Un cours est caractérisé par un enseignant responsable, un numéro d'identification unique, un titre, un nombre de crédits (équivalence ECTS<sup>1</sup>), un niveau de difficulté, un contenu (transparentes, vidéo, diapositives...) disponible sur le serveur, une liste d'exercices, un indice qui reflète l'importance du domaine traité par le cours, un prix à payer lors de l'inscription (déduit de l'indice et du nombre de crédits affectés au cours), une liste d'étudiants inscrits, et un examen qui se passe via internet. Un exercice est associé à un cours. Il est caractérisé par un niveau de difficulté, une liste d'exercices pré requis et une solution.

Un étudiant s'inscrit à des cours en temps différé, dans un site quelconque. Pour chaque cours auquel il est inscrit, il peut envoyer des questions par email à l'enseignant responsable de ce cours qui doit répondre dans les 3 jours. Il peut aussi faire des remarques sur ce cours. Concernant les examens, un étudiant suivant un cours doit s'inscrire à son examen à une date qu'il choisit dans la liste des dates possibles. Dès lors qu'il s'est inscrit, le fait de ne pas passer l'examen équivaut à une note égale à zéro au cours. Un examen d'un cours se déroule en temps limité via Internet, en présence distante du responsable du cours qui peut répondre à des questions de compréhension du sujet.



## 8 Boubker Sbihi

A travers un extrait du système SED, pour simplifier, nous considérerons que le SED permet à des étudiants à distance de suivre des cours en temps différé. Parmi les acteurs du SED, nous citons :

- Les étudiants qui s'inscrivent à des cours, suivent des cours à distance, passent des examens, etc.
- Les enseignants qui produisent des cours, assurent des cours, réalisent un suivi des étudiants, etc.
- Les responsables de sites qui ajoutent de nouveaux cours dans la liste des cours disponibles, gèrent les inscriptions d'étudiants, gèrent les résultats aux examens, etc.

Pour la réalisation d'un tel système on est obligé de formuler notre solution selon la notation UML qui est la norme dans le domaine de l'orienté objet qui est l'UML.

Le langage de modélisation unifié UML standardisé par l'OMG (Object Management Group) [10] est le résultat de la fusion de plusieurs méthodes orientées objets et est devenu la référence en terme de modélisation objet. Il couvre l'aspect statique et dynamique d'un système selon ses différents diagrammes. Son but est de spécifier, visualiser, construire, et documenter les systèmes informatiques dont les systèmes complexes. Pour cela, il définit dix diagrammes pour représenter les différents points de vue de la modélisation

Les dix diagrammes d'UML sont subdivisés en :

- Des vues statiques, qui représentent "physiquement" le système à modéliser au moyen de diagrammes d'objets, de classes, de cas d'utilisation, de composants, de déploiement et d'architecture.
- Des vues dynamiques, qui montrent le fonctionnement du système au moyen de diagrammes de séquence, de collaboration, d'états transitions et d'activités.

La notion de vue d'UML est un moyen mis à la disposition du concepteur pour structurer une conception : cas d'utilisation, logique, composants, déploiement.

Le diagramme de classes est le modèle de base pour l'implémentation d'un logiciel du moment qu'il peut directement être traduit dans les différents langages de programmation tels que Java, C++ ou C#. Le modèle objet qui constitue les briques du logiciel à réaliser peut être directement déduit à partir du modèle de classes correspondant. De plus, il existe plusieurs ateliers de génie logiciel (Rational Rose,...) qui prennent en charge les diagrammes de classes UML et génère par la suite le code y relatif. Après une modélisation UML qui commence par un diagramme de cas d'utilisation suivi d'un diagramme de séquences et d'un diagramme d'activités, le diagramme de classe purement UML du SED se présente dans la figure suivant (figure 2):

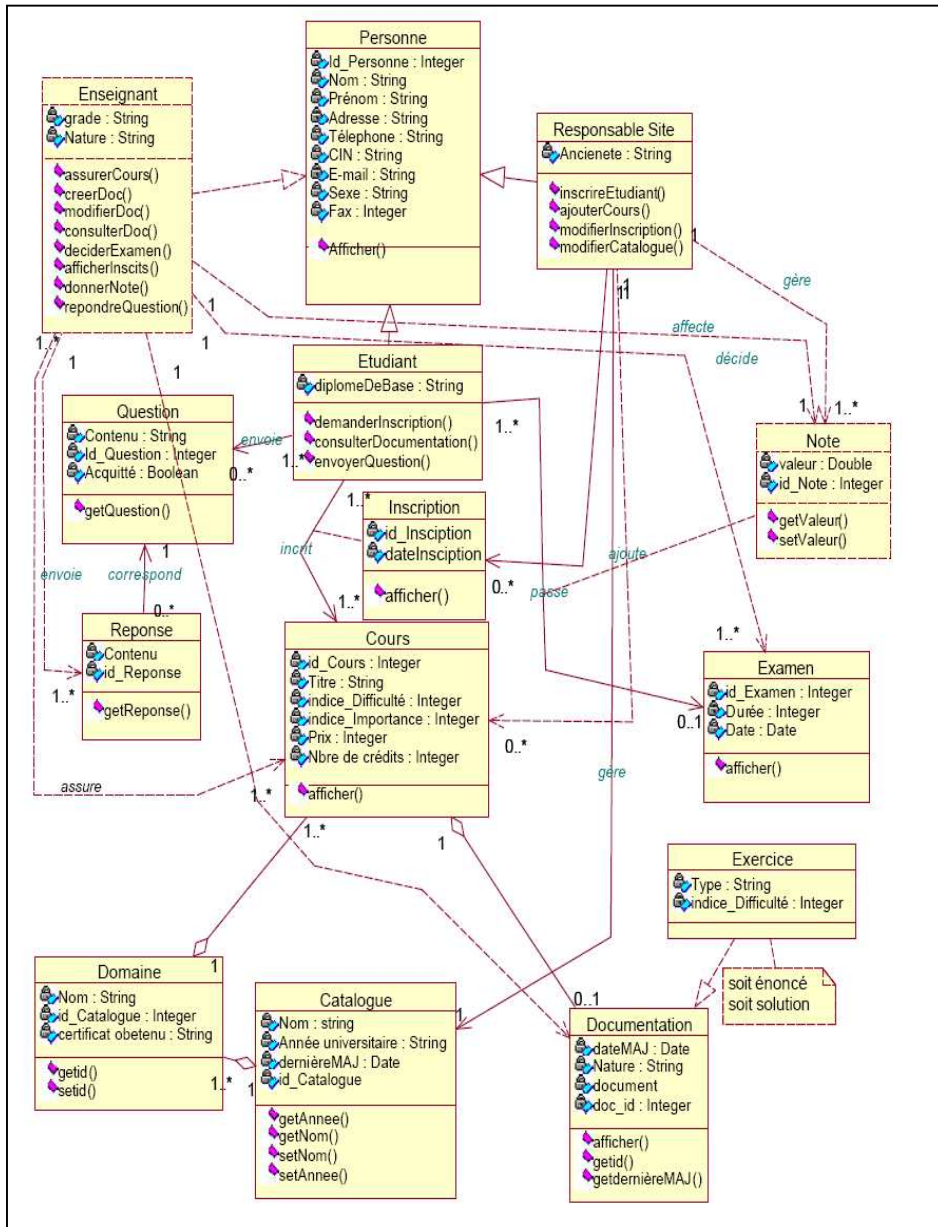


Figure 2. Diagramme de classes purement UML du SED

## 6. Modélisation du SED par l'UML et le filtrage

Le mécanisme de filtrage propose une solution fine que celle présentée par la norme UML puisqu'il définit un filtrage sur les attributs et les méthodes en se basant sur la visibilité de ce dernier. Pour bénéficier de sa réutilisabilité, le code de ces méthodes a été implémenté dans une classe appelée classe de filtrage et peut être réutilisé grâce à l'héritage dans les classes descendantes. Un tel mécanisme n'est pas offert à ce jour par les langages de programmation à objets. La figure 3 illustre une partie du diagramme de classes UML concernant les cours pour comprendre la philosophie du mécanisme de filtrage. Dans cette figure, la classe COURS se compose des classes (Cours\_A\_Suivre, Cours\_A\_Enseigner, Cours\_A\_Gerer). Ces dernières héritent de la classe COURS\_FILTRAGE qui contient la partie publique aux points de vue et les méthodes du mécanisme de filtrage.

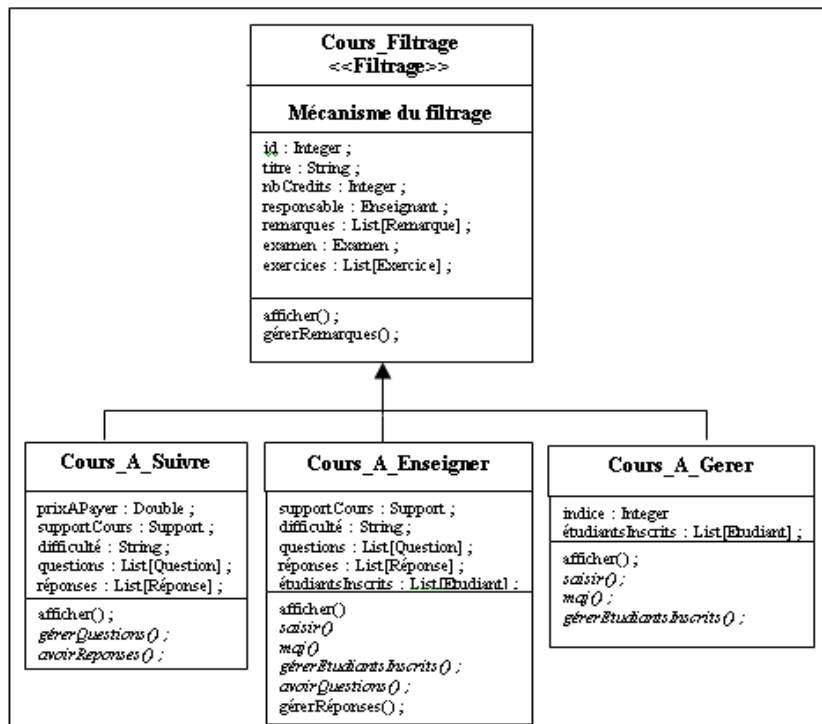


Figure 3. Diagramme de classe correspondant à la classe Cours en UML

Pour la gestion des points de vue, nous définissons un nouveau point de vue qui est celui de l'administrateur des points de vue. Il a la possibilité d'ajouter, modifier, supprimer les autres points de vue d'une manière dynamique. L'administrateur des points de vue doit cependant être muni d'un mot de passe pour le faire. Il possède comme valeurs de visibilité 3 pour chaque attribut et 5 pour chaque méthode. Les attributs communs aux points de vue doivent être définis dans la classe Cours\_Filtrage afin de bénéficier de la réutilisation du code et d'éviter les problèmes de conflits et de mise à jour. Pour assurer l'implémentation de ce mécanisme, nous déclarons l'ensemble des primitives privées UML dans les différentes classes. Le choix de cette visibilité découle du besoin d'interdire l'accès direct à ces dernières. Pour filtrer l'accès à ces primitives, nous ajoutons à la classe de filtrage un certain nombre de méthodes dotées d'une visibilité publique UML et prenant comme paramètres les points de vue et leurs mots de passe. Ces méthodes sont : AccederLectureAttribut, AccederEcritureAttribut et InvokerMéthode. Le tableau suivant (Tableau 4) présente le dictionnaire des primitives de l'exemple SED.

N° et Nom de la classe		N° et Nom de la primitive		N° du point de vue (*)			
				0	1	2	3
1	Cours_Filtrage	1	id	3	2	2	3
		2	Tigre	3	2	2	3
		3	nbCredits	3	2	2	3
		4	Responsable	3	2	2	3
		5	Remarques	3	2	2	3
		6	examen	3	2	3	3
		7	exercices	3	2	3	3
		8	supportCours	3	3	3	3
		9	Difficulté	3	2	2	3
		10	questions	3	2	3	3
		11	réponses	3	3	3	3
		12	étudiantsInscrits	3	2	2	3
		13	afficher();	5	5	5	5
		14	gérerRemarques()	5	4	5	5
2	Cours_A_Suivre	1	prixAPayer	3	2	2	3
		2	Etudiantafficher()	5	5	4	4
		3	gérerQuestions()	5	5	4	4
		4	avoirReponses()	5	5	4	4
3	Cours_A_Enseigner	1	Enseignantafficher()	5	4	5	4
		2	Enseignantsaisir()	5	4	5	4
		3	Enseignantmaj()	5	4	5	4
		4	avoirQuestions()	5	4	5	4
		5	gérerRéponses()	5	4	5	4
4	Cours_A_Gerer	1	Indice	3	4	4	5
		2	Responsablesaisir()	5	4	4	5
		3	Responsablemaj()	5	4	4	5
		4	Responsableafficher()	5	4	4	5

Tableau 4. Extrait du dictionnaire des primitives

(\*) 0 : Administrateur; 1 : Etudiant ; 2 : Enseignant; 3 : Responsable

Comme exemple d'utilisation de ces méthodes, on peut utiliser pour accéder en lecture à un attribut selon un point de vue spécifique, à titre d'exemple, l'attribut multi-points de vue « Remarques » dans la classe Cours par l'administrateur, la méthode **AccederLectureAttribut** qui retourne la valeur de la primitive et qui possède comme paramètres : Point de Vue, Nom Classe, Nom Primitive et le privilège. L'appel de cette méthode est : **AccederLectureAttribut** ("Administrateur", "Cours\_filtre", "Remarques", "mot\_passe");

## 7. Conclusion

L'approche présentée dans cet article a pour intérêt d'intégrer le concept de point de vue dans le diagramme de classes UML et spécialement le système d'enseignement à distance. Cette approche a pour but de compléter l'UML et d'automatiser par la suite la génération du code multivues. Le travail présenté dans cet article fait partie d'un projet plus vaste qui a pour travaux qui restent à entamer : la définition de la notion de composant multivues, l'élaboration d'une base de patrons supportant l'approche par points de vue, l'ajout d'un autre diagramme de la modélisation dynamique d'UML

## 8. Bibliographie

- [1] : AspectJ-Oriented Programming (AOP) for Java. <http://www.aspectj.org>.
- [2] : BOURAQADIN, LEDOUX.T. « The point on the programming by aspects ». *Technique and Computer Sciences*, vol 20, page 505 à 528, Hermès, 2001.
- [3] : BARDOU.D, « survey of languages to prototypes, of the delegation mechanism, and of its report to the notion of point of view ». Thesis of doctorate, computer specialty, University Montpellier 2, April 1998.
- [4] : CLARKE.S. « Extending standard UML with model composition semantics ». *Science of Computer Programming*, Elsevier Science, 2002.
- [5] : COULETTE.B, KRIOUILE.A, MARCAILLOU.S, « The approach by points of view in the development oriented object of the complex systems ». *The object*, vol 2, n°4, p. 13-20, 1996.
- [6] : FORCIIR : <http://www.ebad.ucad.sn/forciir/>
- [7] : HAIR.A, SBIHI.B, BELANGOUR.A, U\_VBOOM: Unified Analysis and Design Process Based on the Viewpoint Concept. *Revue RIST*, Volume 14, N°1,2004.

[8] : MDA Guide, version 1.0, version 2.0, document OMG : <http://www.OMG.org/mda>

[9] : LEBLANC.P, Implementation of the UML Testing Profile and Production of Executable Test Cases , *White Paper Telelogic*, [www.telelogic.com](http://www.telelogic.com) , 2004.

[10] : Object Management Group : <http://www.omg.org>

[11] : OSSHER.H, KAPLAN.M, HARRISON.W, KATZ.A AND KRUSKAL.V. « Specifying subject oriented composition ». *To appear in Theory and Practice of Object Systems (TAPOS)*, 2(3), 1996. Special issue on Subjectivity in Object-Oriented Systems, 1996.

[12] : SBIHL.B, KRIOUILE.A, ETTALBI.A, COULETTE. B. « Toward a generation of code multi-targets for the VBOOM method : Approche by filtering, » *The International Conference on Software Engineering Research and Practice (SERP'03)*, Las vegas, USA 2003.

[13] : SBIHL.B, La modélisation par les réseaux de pétri du mécanisme de filtrage UML; WONTIC'05, *Workshop sur les Technologies de l'Information et de la communication*, Kénitra, Maroc, 24- 25 juin 2005

[14] : SBIHL.B. , HAIR.A, ETTALBI.A. «The integration of the points of view notion in UML». *Act The IADIS Applied Computing 2004 conference*, Lisbon, Portugal 2004.

[15] : Unified Modeling Language 2.0, 2004, <http://www.omg.org/uml/>

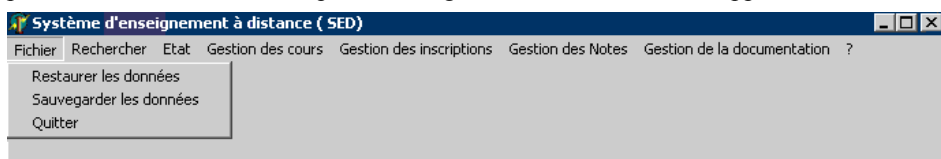
[16] : U2 Partners, UML, version 2.0, <http://www.U2-Partners.org>

---

## 9. Figures du système

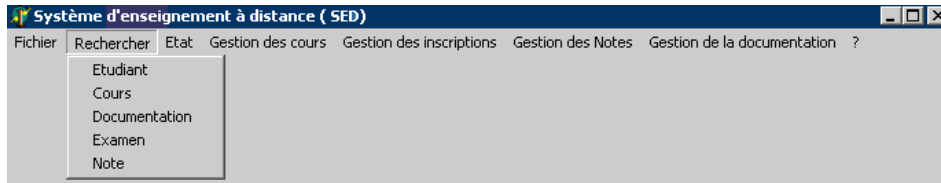
- **Menu de l'application pour la persistance**

Ce menu déroulant intégrera les fonctions « classiques » d'un logiciel gérant la persistance de fichier, telles que la sauvegarde, la restauration, la suppression, etc.



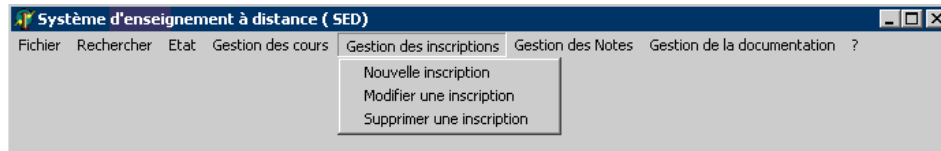
- **Menu de recherche**

Ce menu déroulant intégrera les fonctions de recherches, telles que un étudiant, un cours, une documentation, un examen ou une note.



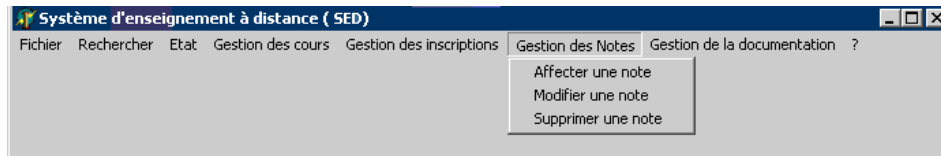
- **Menu des inscriptions**

Ce menu déroulant constituera le point d'entrée principal dans le logiciel, puisque qu'il gère les inscriptions



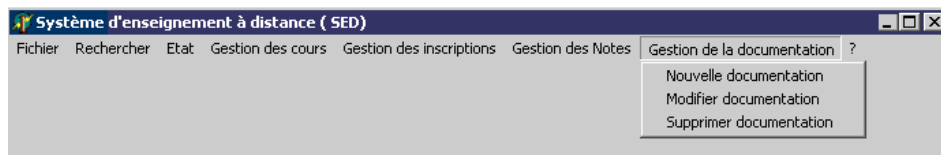
- **Menu de la gestion des notes**

Ce menu déroulant constituera le point d'entrée principal dans le logiciel pour les notes des étudiants dans les différents examens



- **Menu de la documentation**

Ce menu déroulant gèrera la documentation en ajoutant, modifiant et supprimant les documentations



- **Menu d'ajout d'étudiant**

Le bouton Nouvelle Etudiant de cette fenêtre de dialogue permettra l'accès à la fenêtre suivante pour ajouter un étudiant s'il ne figure pas parmi la liste des étudiants déjà existant

**Form**

### Ajout d'un étudiant

*Caractéristique Etudiant*

**Nom** merouani      **Prénom**      **Sexe** Masculin

**Adresse** 11,rue merouani casa      **Téléphone** 0154465656      **FAX** 0154546

**Diplome de base** Ingénieur d'état

**E-mail** amerouani p@yahoo.fr

**CIN** LB565

**Observation**  
Est ce que cette formation est nécessaire pour un ingénieur en Génie Logiciel?

Ajouter Etudiant

- **Menu d'ajout d'un cours**

Ce menu permettra l'accès aux outils de gestions des cours : Ajout/Modification/Suppression d'un cours d'un domaine déterminé.

**Ajout d'un cours**

### Ajout d'un cours à partir d'un domaine

**Catalogue souhaité** Catalogue 2005      **Domaine souhaité** Unix

*Caractéristique*

**Titre** Administration Réseau sous UNIX

**indice de difficulté (x/10)** 5      **indice d'importance(x/10)** 10

**Prix (En EURO)** 100

**Nombre de crédits** 11

**Remarque**  
Cette formation nécessite une connaissance :  
-de la programmation Cshell  
-Comandes UNIX  
-Architecture Client Serveur

Ajouter Cours

- **Sous menu de recherche**

Ce sous menu permettra la recherche d'un étudiant par numéro



Système d'enseignement à distance ( SED)

## Recherche d'un etudiant

Numéro Etudiant:

Nom Etudiant:

Numéro	Nom	Adresse	Email	Note finale
40	Adil manari	24 rue imouraz rabat	manari@yahoo .fr	12,45