



Université de Fribourg, Suisse

Département d'informatique

Bachelor en informatique de gestion

**ANALYSE D'UN SYSTEME D'INFORMATION ET EXTENSION DE
CELUI-CI PAR DE NOUVELLES FONCTIONNALITES**

Travail de séminaire

Elira Shehu

Varis 7, 1700 Fribourg

Dr. Stefan Hüseemann

Printemps 2008

Table des matières

1	INTRODUCTION	1
1.1	Problématique	1
1.2	Objectifs	1
1.3	Façon de procéder	2
2	L'ANALYSE DES FONCTIONNALITES D'UN SYSTEME D'INFORMATION	4
2.1	La méthode RUP	4
2.1.1	<i>Qu'est-ce que RUP ?</i>	4
2.1.2	<i>L'analyse des besoins et la discipline « Requirements »</i>	7
2.1.3	<i>La discipline « Analysis and Design »</i>	9
2.2	Le langage de modélisation UML.....	11
2.2.1	<i>Qu'est-ce que UML ?</i>	11
2.2.2	<i>Les diagrammes UML de cas d'utilisation</i>	12
2.2.3	<i>Le modèle des cas d'utilisation dans RUP</i>	13
3	LES ORDRES BOURSIERS A COURS LIMITE.....	16
3.1	Définition	16
3.2	Risques	16
3.3	L'intérêt des ordres à cours limité pour les membres de la BSU.....	16
4	ARCHITECTURE DU SYSTEME D'INFORMATION BSU.....	17
5	ANALYSE DU SI BSU ET EXTENSION DU MODELE AVEC UNE NOUVELLE FONCTIONNALITE.....	19
5.1	Software Requirements Specification.....	19
5.1.1	<i>Introduction</i>	19
5.1.1.1	Objectif :	19
5.1.1.2	Cadre :	19
5.1.1.3	Définitions, acronymes et abréviations :	19
5.1.2	<i>Description</i>	20
5.1.2.1	Etude du modèle des cas d'utilisation	20
5.2	Spécification des cas d'utilisation.....	25
5.2.1	<i>Cas d'utilisation : Passer un ordre</i>	25
5.2.2	<i>Cas d'utilisation: Passer un ordre avec limite</i>	26

Table des matières	2
6	IMPLEMENTATION DE LA NOUVELLE FONCTIONNALITE 28
6.1	« Reengineering » dans le cas BSU 28
7	CONCLUSION..... 29
8	REFERENCES 30

1 Introduction

1.1 *Problématique*

Börsenspiel der Schweizer Universitäten (BSU) est une association d'étudiants de l'Université de Fribourg qui organise des simulations boursières destinées à tous les étudiants des universités et des hautes écoles suisses. Son but principal est de familiariser les étudiants avec les marchés financiers et de faire le lien entre la théorie et la pratique en leur proposant de prendre, pendant plusieurs semaines, le rôle d'un gestionnaire de portefeuille fictif.

Le jeu, qui se nomme Portfolio Management Simulation (PMS), se déroule entièrement en ligne, sur le site web de l'association. Les participants, qu'on appellera aussi les joueurs ou les membres, ont la possibilité d'investir leur capital de départ fictif dans un choix de divers produits financiers proposés aux cours réels de la bourse et tenter ainsi de le faire fructifier. Des ordres peuvent être placés à n'importe quel moment de la journée, mais les portefeuilles sont actualisés seulement une fois par jour et au prix de clôture du marché. En outre, l'unique façon de négocier est de passer des ordres à tout prix, autrefois appelés ordres au marché, c.à.d. sans tenir compte de la limite du prix à l'achat ou à la vente.

L'association BSU travaille dans le but d'améliorer continuellement la qualité de ce jeu proposé aux étudiants et dans ce même objectif, elle aimerait élargir la simulation boursière par une nouvelle fonctionnalité et offrir à ses joueurs la possibilité de négocier avec des ordres à cours limités. Ceci rendrait le jeu plus varié et encore plus intéressant et attractif pour les étudiants, en leur permettant de mieux gérer les risques de leurs investissements fictifs. Pourtant, faut-il encore connaître la façon de procéder pour réaliser le projet et les changements dans le système d'information de l'association que celui-ci implique.

1.2 *Objectifs*

Il va de soi que pour apporter avec succès des modifications souhaitées à un système d'information existant, il faut, dans un premier abord, bien être au courant de son état actuel. Un système d'information peut être assez complexe et posséder plusieurs facettes. Il existe, donc, plusieurs façons de voir un tel système et d'autant plus de manières pour le décrire, selon que l'on s'intéresse à l'un ou à l'autre aspect de celui-ci.

Le premier but de ce travail est d'analyser le système d'information de l'association BSU (SI BSU) et de décrire ses fonctionnalités présentes du point de vue de l'utilisateur. Par le mot *utilisateur* on entend aussi bien le joueur que l'administrateur du jeu et donc, les deux cas de figure vont être considérés dans le modèle descriptif.

Le deuxième objectif est de considérer l'utilité et l'usage de la nouvelle fonctionnalité à implémenter dans le système d'information, notamment la négociation avec des ordres à cours limité, et de concevoir le modèle d'un nouveau système d'information agrandi par cette fonctionnalité.

Finalement, les changements nécessaires à apporter dans la pratique aux différents composants de l'architecture du système d'information BSU seront discutés et un prototype visuel illustrant le fonctionnement du jeu avec la nouvelle fonctionnalité intégrée sera proposé.

1.3 Façon de procéder

Pour la conception d'un système d'information on a besoin d'outils pour représenter les différentes facettes de ce qui existe déjà et de ce qui doit être créé par-dessus. C'est la raison pour laquelle on utilise des *modèles*, qui ne sont autre que des représentations visuelles simplifiées de la réalité, reflétant uniquement les aspects essentiels de celle-ci.

Afin d'analyser le système d'information de l'association BSU, on fera appel à l'approche proposée par la méthodologie Rational Unified Process (RUP) et plus spécifiquement par les deux disciplines de cette méthodologie qui sont intéressantes dans le cadre de ce travail : « Requirements » et (dans un moindre degré) « Analysis and Design ».

Plusieurs modèles décrivant chacun un aspect différent, mais de manière cohérente entre eux, sont généralement nécessaires pour examiner le mieux un système d'information et élaborer de manière efficace le cahier des charges, sur le quel on s'appuiera pour construire. Cependant, comme indiqué plus haut (voir *1.2 Objectifs*), on s'intéresse ici à savoir uniquement comment le système d'information BSU est ou sera utilisé. C'est la raison pour laquelle on emploiera un seul modèle indiqué dans RUP : le modèle des cas d'utilisation, basé sur les diagrammes de cas d'utilisation du langage descriptif orienté objet Unified Modelling Language ou UML. Un outil de modélisation pour le langage UML tel que « MagicDraw UML » sera utilisé dans ce but.

En premier lieu, des concepts de base et l'intérêt de la méthodologie RUP et du langage UML, ainsi que la relation existante entre les deux seront présentés.

Deuxièmement, le cas des ordres à cours limité du point de vue financier et l'intérêt d'une nouvelle fonctionnalité dans le jeu PMS permettant aux membres de négocier avec ce type d'ordres seront discutés.

Troisièmement, un aperçu général de l'architecture du système informatique BSU sera donné.

Finalement, une fois les processus qui revèlent du domaine Jeu PMS identifiés et analysés, le modèle des diagrammes UML de cas d'utilisations, élargi par la nouvelle fonctionnalité, sera exposé et des modifications à apporter à l'architecture du système informatique pour réaliser l'implémentation seront suggérés.

2 L'analyse des fonctionnalités d'un système d'information

2.1 La méthode RUP

2.1.1 Qu'est-ce que RUP ?

RUP ou Rational Unified Process est une des plus célèbres implémentations d'un processus unifié, défini comme une méthode de prise en charge du cycle de vie d'un logiciel et donc du développement, pour les logiciels orientés objets [Wikipedia 2007]. Il a été élaboré par l'entreprise « Rational » qui fut rachetée par IBM en 2003 et dont les fondateurs sont également les auteurs des premières versions d'UML (voir 2.2 *Le langage de modélisation UML*).

RUP est, en d'autres termes, une méthodologie de gestion de projets informatiques ou un processus de génie logiciel. Plusieurs entreprises se sont peu à peu rendues compte au cours des dernières années de l'importance que peut avoir un processus de gestion de projet bien défini et bien documenté. Certaines d'entre elles, se référant au modèle Capability Maturity Model (CMM) et visant à atteindre le niveau 2, 3, ou plus haut, possèdent déjà de précieuses collections de leurs connaissances et des meilleures pratiques, qu'elles transmettent à leurs développeurs sous forme de livres publiés, méthodes de conception, programmes de formations et diverses notes amassées à l'intérieur de l'entreprise lors de différents projets. Pourtant, dans la pratique, ces processus développés à l'interne finissent souvent par prendre la poussière et en étant rarement mis à jour, deviennent obsolètes et ne sont jamais vraiment suivis. D'autres entreprises ne possèdent aucun processus et nécessitent un point de départ pour progresser dans le chemin du développement plus rapide de produits logiciels d'une meilleure qualité. RUP peut aider les deux types d'entreprises en leur fournissant un processus de gestion mature, rigoureux et flexible [Kruchten 2001].

RUP n'est pas seulement une méthode de développement de processus qui a été écrite sur du papier une fois pour toutes. RUP est aussi un logiciel. Il est donc lui-même un produit issu d'un processus. Comme le dit Lee Osterweil, Professeur en Informatique à l'Université de Massachusetts, « *Software processes are software, too.* »

RUP a été conçu et documenté en utilisant le langage UML (voir 2.2 *Le langage de modélisation UML*). Il est distribué sous forme de collection de pages web interactives, de documents types et de guides outils. Par sa modularité et sa forme électronique, RUP peut être configuré, taillé et personnalisé suivant les besoins spécifiques de l'entreprise durant le

processus de développement. En outre, il est en constante évolution et des améliorations sont proposées avec chaque nouvelle version.

La Figure 2-1 montre une page typique dans RUP.

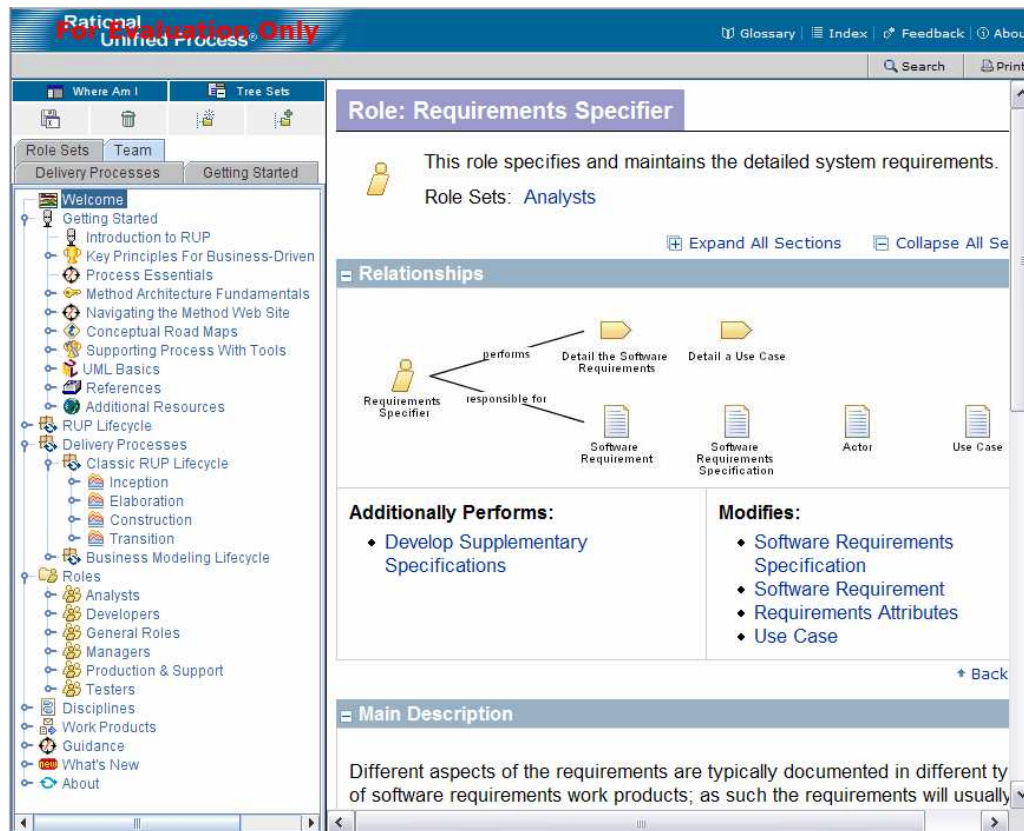


Figure 2-1: Une page de RUP [RUP 2007]

RUP est une méthodologie orientée objet basée sur l'approche itérative et incrémentale, contrairement à des méthodes séquentielles, telles que le modèle en cascade ou Merise. Il possède deux dimensions :

- La dimension horizontale représente le temps et montre les cycles de vie du processus pendant sa durée.
- La dimension verticale représente les disciplines (les workflows) dans le cœur du processus, qui groupent les activités logiquement d'après leur nature.

La première dimension, appelée aussi l'aspect dynamique du RUP, est exprimée en terme de cycles, de phases, d'itérations et de milestones. Les logiciels sont conçus ainsi,

dans une suite d'itérations qui permettent de tester et détecter les problèmes déjà dans les débuts du projet.

La deuxième dimension, représente l'aspect statique du processus et décrit celui-ci en termes de composantes de processus telles que : activités, disciplines, artefacts (produits de travail) et rôles.

La Figure 2-2 montre ces deux dimensions de RUP. La grandeur des zones colorées s'étalant dans le graphique indique le « taux d'activité » de chaque discipline dans les différentes phases du projet.

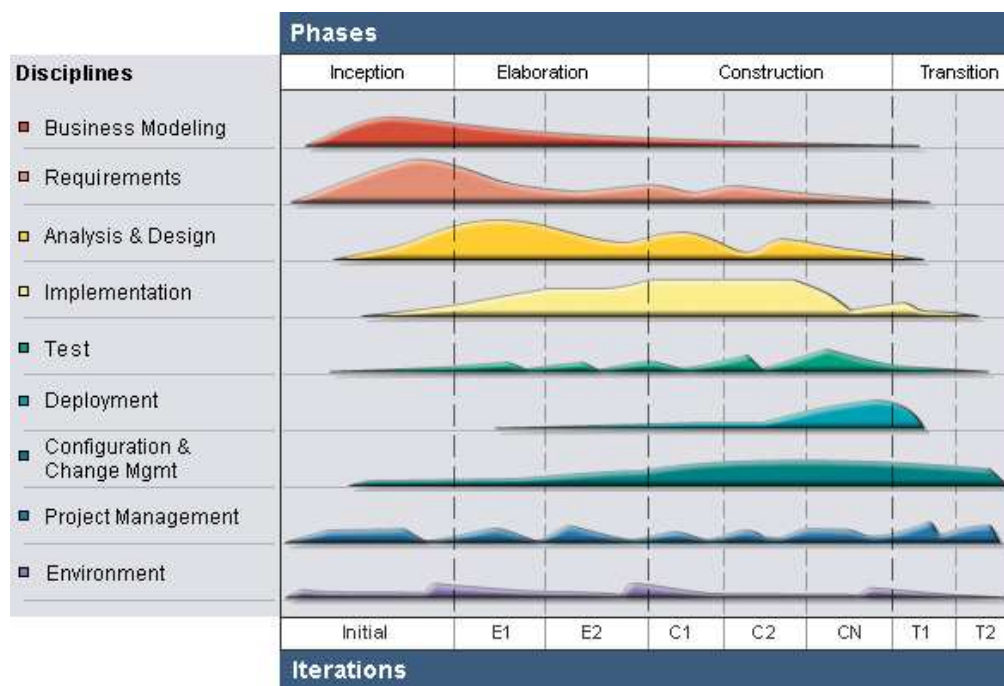


Figure 2-2: Les deux dimensions de RUP [RUP 2007]

Finalement, RUP est un ensemble de principes et de philosophies pour le développement avec succès des produits logiciels. Il rassemble les meilleures pratiques modernes dans le domaine du génie logiciel, appropriées à une vaste gamme de projets ou entreprises, telles que : le développement de manière itérative, la gestion des exigences, l'usage d'architectures basées composantes (orientées objet), la modélisation visuelle des logiciels, le contrôle continu de la qualité, etc.

2.1.2 L'analyse des besoins et la discipline « Requirements »

Qu'un projet informatique ait pour but de développer un nouveau système d'information ou d'améliorer un système existant, il requiert une exécution d'activités de tout genre, partant des activités en lien avec la conception jusqu'à celles ayant trait à l'implémentation, aux tests ou à la mise en œuvre du SI.

Un projet bien réalisé est non seulement un projet réalisé dans les délais et dans le budget accordé dès le départ, mais surtout un projet livrant un produit qui satisfait les besoins des utilisateurs finaux, c'est-à-dire un système d'information ayant toutes les fonctionnalités exigées par eux. La majorité des projets informatiques dans le monde sont voués à l'échec total ou partiel, selon leur degré de satisfaction de ces trois contraintes, et une des principales raisons pour leurs échecs est le manque de clarté sur les vrais besoins des utilisateurs [Standish Group 2001]

Ceci indique qu'il n'est pas toujours très aisé de savoir ce que l'utilisateur final exige en réalité, pour différentes raisons :

- Les utilisateurs ne sont pas ou très peu consultés durant le projet.
- Les développeurs ne comprennent pas bien le fonctionnement des processus de l'entreprise client.
- Les utilisateurs n'ont pas exactement conscience de leurs propres besoins.
- Les besoins sont mal exprimés ou documentés.
- Les besoins peuvent changer au fur et à mesure du projet, mais ils ne sont pas pris en compte par les développeurs, etc.

Afin de mener un projet à bien, il est nécessaire de porter une grande part de l'attention à l'analyse des besoins et que cela soit un domaine pris en charge par des analystes du business (*business analysts*), plutôt que par des programmeurs. Ce domaine est souvent référencé en anglais comme *Requirements Engineering* ou encore *Requirements Specifications*. Il s'agit là d'un processus pour l'analyse, la description et la validation des exigences envers un système d'information et dont le résultat est une spécification des besoins [IEEE Std. 830 – 1993].

Les tâches du domaine *Requirements Engineering* sont pris en charge dans RUP par l'une des disciplines concernant la conception des SI, la discipline « Requirements ».

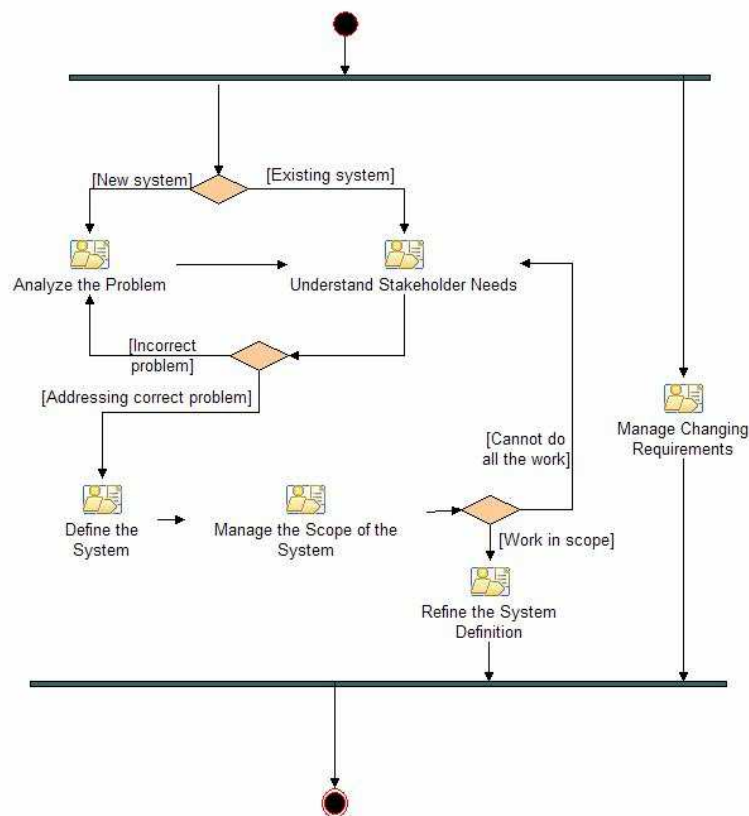


Figure 2-3: Le flux de travail de la discipline « Requirements » [RUP 2007]

« Requirements » propose une suite d'activités à exécuter, distribue les rôles aux responsables pour les activités et met à disposition des modèles de produits de travail qui doivent être fournis au fur et à mesure. Son but est d'expliquer comment découvrir les besoins des parties prenantes et les transformer en produits de travail qui fourniront des spécifications détaillées sur ce que le système à construire doit être capable de faire.

La Figure 2-3 montre le diagramme d'activités de ce processus ou son workflow qui sert de ligne de conduite générale. Dans ce diagramme, on voit un point de départ et un point de finalisation représentés par des cercles, entre les quels une suite d'activités et des points de décisions se situent dans un ordre logique indiqué par des flèches.

Quelques tâches dans cette discipline sont :

- Etablir un vocabulaire commun entre les utilisateurs finaux et les réalisateurs du projet et arriver à un consensus sur ce que le système doit faire.
- Détailler les spécifications du logiciel.
- Détailler un plan de gestion des exigences.

- Définir les frontières du système logiciel.
- Identifier les acteurs et ce qu'on appelle les « cas d'utilisation » ou les façons dont le logiciel pourra être utilisé. (voir 2.2.2 *Les diagrammes UML de cas d'utilisation*)
- Structurer le modèle des cas d'utilisations.
- Développer une vision générale du logiciel du point de vue des utilisateurs, etc.

La discipline « Requirements » est étroitement liée à d'autres disciplines, notamment à « Analysis and Design », à laquelle elle fournit les principales entrées, à « Test » qui évalue le système d'après la spécification des exigences, à « Configuration and Change Management » qui lui fournit le mécanisme du contrôle des changements des besoins, etc.

Les principaux acteurs de « Requirements » sont les *System Analysts* ou *Business Analysts*. L'un des artefacts qui nous intéresse particulièrement dans le cadre de ce travail est le *Modèle des Cas d'Utilisation*, dont on parlera dans la section suivante de ce chapitre.

2.1.3 La discipline « Analysis and Design »

« Analysis and Design » est une autre discipline dans RUP qui groupe des activités en relation avec la conception d'un système d'information. Elle est étroitement liée avec la discipline « Requirements » dans le sens qu'elle explique comment transformer les produits de travail de l'analyse des besoins en produits de travaux spécifiant le design du logiciel que le projet a pour but de développer. On passe, donc, d'un point de vue utilisateur, à un point de vue architecture, qui servira par la suite à l'implémentation réelle du logiciel.

Une représentation graphique du workflow de ce processus est donnée par la Figure 2-4.

Les objectifs principaux de cette discipline sont :

- Transformer les spécifications des besoins en un design du système à construire.
- Développer une architecture robuste pour le système.
- Adapter le design afin de le faire correspondre à l'environnement d'implémentation.

Ces objectifs peuvent être atteints en suivant les différentes tâches prévues par cette discipline, dont :

- L'analyse de l'architecture
- Le design des classes
- Le design de la base de données

- Le design de l'interface utilisateur
- Le design des éléments d'évaluation et de test
- L'identification des services
- L'analyse et le design des opérations.
- La création des prototypes pour l'interface utilisateurs, etc.

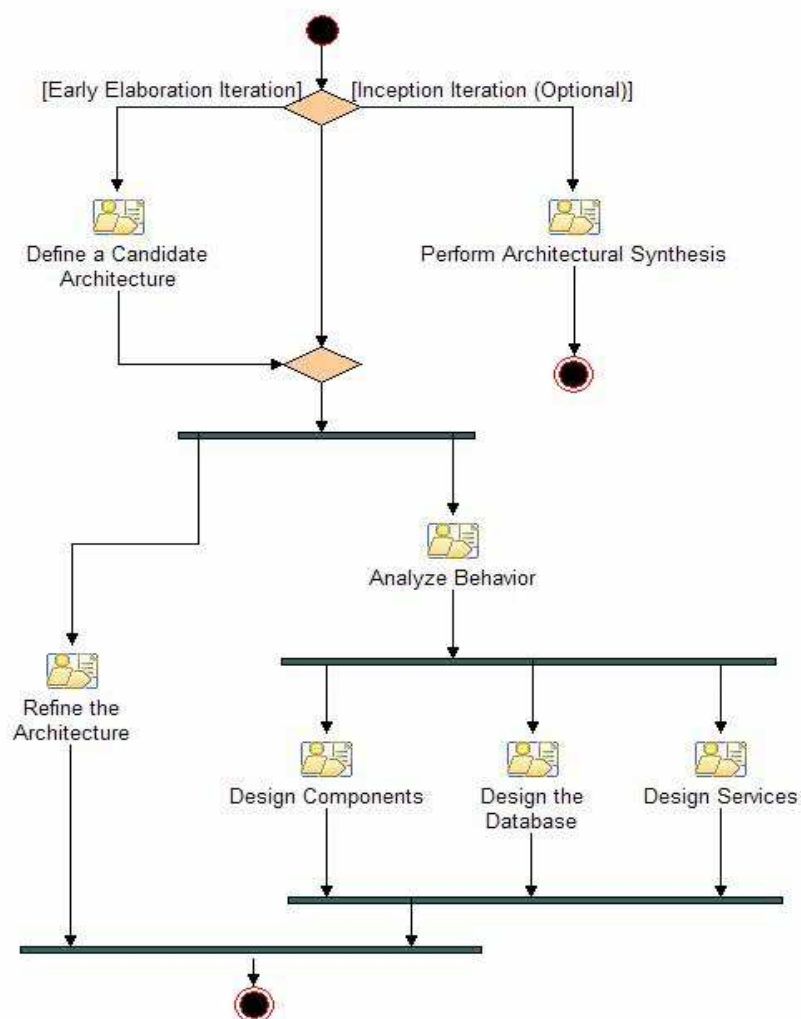


Figure 2-4: Le flux de travail dans la discipline « Analysis and Design » [RUP 2007]

2.2 Le langage de modélisation UML

2.2.1 Qu'est-ce que UML ?

UML, *Unified Modelling Language*, est un langage de modélisation orienté objet dont l'objectif initial était de permettre aux informaticiens de représenter un système logiciel et son utilisation prévue dans l'entreprise, afin d'améliorer la qualité des applications informatiques qu'ils développaient. [Morley 2006, p.17]. Ses origines se trouvant vers le milieu des années '90, UML s'est développé par versions successives jusqu'à devenir un standard de fait et ouvert, géré et normalisé par l'OMG.¹ La dernière spécification UML à ce jour est la version 2.1.1 et dans le cadre de ce travail on fera référence simplement à UML 2. Il a été fondé et influencé par des noms importants dans le courant de l'orienté objet, tels que J. Rumbaugh, G. Booch et I. Jacobson, parmi d'autres.

Plus concrètement, UML est un ensemble de notations graphiques ou modèles, qui s'appuient sur une syntaxe appelée méta-modèle. Le méta-modèle régit l'utilisation des éléments UML et assure que tous les outils de génie logiciel orientés UML puissent présenter une cohérence et autoriser l'interopérabilité de ses modèles, c'est-à-dire leur échange entre ateliers. Les modèles, quant à eux, sont des ensembles de diagrammes UML qui ont été élaborés pour modéliser ou représenter différents aspects d'un système d'information. Que ce soit pour communiquer des idées et des solutions entre les gestionnaires de projet, les utilisateurs finaux et les développeurs ou pour décrire en détails la composition et le mode de fonctionnement d'un produit logiciel à l'attention des programmeurs, UML se prête à une utilisation aussi bien au niveau conceptuel (système d'information) qu'au niveau logiciel (système informatique).²

Pour l'analyse d'un système d'information, le recours à la modélisation visuelle est d'un apport majeur, parce qu'il aide à réduire les difficultés de compréhension liées à la complexité du système. Les différents éléments hétérogènes qui sont tissés ensembles pour former un système d'information – les informations, les acteurs, les processus, les technologies – nécessitent plusieurs modèles pour leur représentation, suivant la perspective qu'on adopte.

¹ *Object Management Group*, consortium ouvert d'entreprises, ayant pour mission d'élaborer des standards qui facilitent l'interopérabilité dans le monde objet, c'est-à-dire la communication et l'échange de systèmes construits dans des environnements différents. (www.omg.org)

² Si on devait considérer la différence entre un système d'information et un système informatique, le premier ce serait l'ensemble de toutes les personnes, des machines et des logiciels impliqués dans l'acquisition, le

Les treize types de diagrammes UML existants peuvent être utilisés pour accomplir cette tâche. Ils peuvent être employés pour décrire aussi bien la structure statique (les composantes) d'un système étudié, que sa structure dynamique (son comportement). Selon le point de vue que l'on adopte, certains modèles et diagrammes peuvent paraître plus intéressants que d'autres. Ainsi, par exemple, les diagrammes d'activités utilisés pour décrire des flots de contrôle, peuvent être utiles aussi bien pour l'utilisateur final qui veut connaître son rôle dans le système, que pour le développeur qui construira le logiciel d'après la logique exigée. Par contre, des diagrammes de déploiement ou de timing, intéressent exclusivement les informaticiens.

2.2.2 Les diagrammes UML de cas d'utilisation

Comme il a été dit dans le point précédent, un type de diagramme est utilisé pour représenter une facette du système d'information, d'après une perspective donnée. Dans le cadre de ce travail, nous voulons décrire le système d'information BSU d'après le point de vue des utilisateurs (joueurs et administrateurs) et dans ce but, nous allons privilégier un type de diagramme proposé par UML que l'on appelle le diagramme de cas d'utilisation.

Un *cas d'utilisation* est défini comme une façon spécifique d'utiliser le système. Il est composé d'un ensemble d'actions déclenché par un acteur externe et qui produit un résultat identifiable. Ce regroupement d'actions ne correspond pas à une structure implémentée ou à implémenter directement du système, mais plutôt à une séquence perçue comme logique par l'utilisateur. [Morley 2006]

Un modèle UML de cas d'utilisation recense tous les cas d'utilisations d'un système d'information identifiés et montre ceux-ci en relation avec les acteurs ou avec d'autres cas d'utilisation. Chaque cas est ensuite éclairé par une description textuelle où figurent : l'objectif du cas, les acteurs associés, les actions exprimées sous formes de scénarios, les règles de gestion, ainsi que les références des documents utilisés ou produits durant les activités du cas.

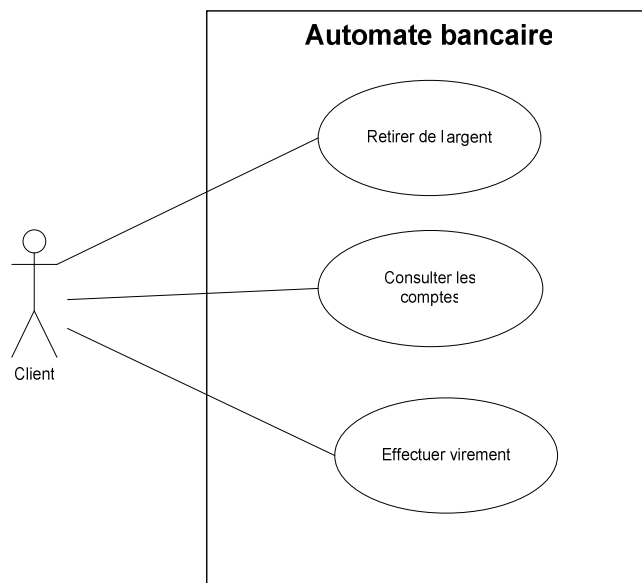


Figure 2-5: Exemple d'un diagramme simplifié de cas d'utilisation

Les cas d'utilisation peuvent être groupés dans des diagrammes selon les critères librement choisis par le gestionnaire. Par exemple, il peut créer un diagramme pour chaque acteur identifié ou encore un diagramme regroupant des cas d'utilisation d'un domaine précis.

La Figure 2-5 montre un exemple de modélisation simple d'un guichet bancaire automatique. Elle illustre les trois façons dont la borne peut être utilisée (pour retirer de l'argent, pour consulter les comptes ou pour effectuer un virement), l'acteur (le client de la banque) et les associations entre l'acteur et les cas d'utilisation (l'acteur utilise le système d'information). Des exemples plus complets et détaillés sont présentés dans les pages qui suivent (voir 5 *Analyse du SI BSU et extension du modèle avec une nouvelle fonctionnalité.*), où la modélisation du système d'information BSU est discutée.

2.2.3 Le modèle des cas d'utilisation dans RUP

RUP est basé UML non seulement en ce qui concerne sa conception, mais aussi en ce qui concerne l'approche qu'il propose. Les diagrammes UML se trouvent dans un grand nombre de produits de travail ou *artefacts* de diverses disciplines de RUP. Comme il est montré dans la Figure 2-6, RUP se base spécialement sur les cas d'utilisation. Le modèle UML des cas d'utilisation est un produit de travail à fournir par la discipline « Requirements » et à utiliser par toutes les autres disciplines en charge d'activités d'analyse, de design ou de test.

Comme il est indiqué dans RUP, le modèle de cas d'utilisation (Use-Case Model) est un modèle des fonctionnalités et de l'environnement du système et il sert de contrat entre le

client et les développeurs. Le modèle est sensé être utilisé par différentes personnes impliquées dans le projet de manières différentes:

- Le client approuve le modèle des cas d'utilisation. A ce moment, on sait que le système est vraiment ce que le client veut. Le modèle peut aussi être utilisé pour discuter avec le client durant le développement.
- Les utilisateurs potentiels s'en servent pour mieux comprendre l'utilisation du système final.
- L'architecte logiciel utilise le modèle pour identifier les fonctionnalités significatives du point de vue architecture.
- Les designers utilisent le modèle comme base pour leur travail.
- Le manager l'utilise pour planifier et suivre la modélisation des cas d'utilisation et le design qui suit.
- Les testeurs se réfèrent aux fonctionnalités indiquées dans le modèle des cas d'utilisations pour tester le système.
- Les rédacteurs de la documentation se basent sur ce modèle pour écrire le guide de l'utilisateur du système.
- Les développeurs de futures versions utilisent le modèle pour comprendre la version actuelle.

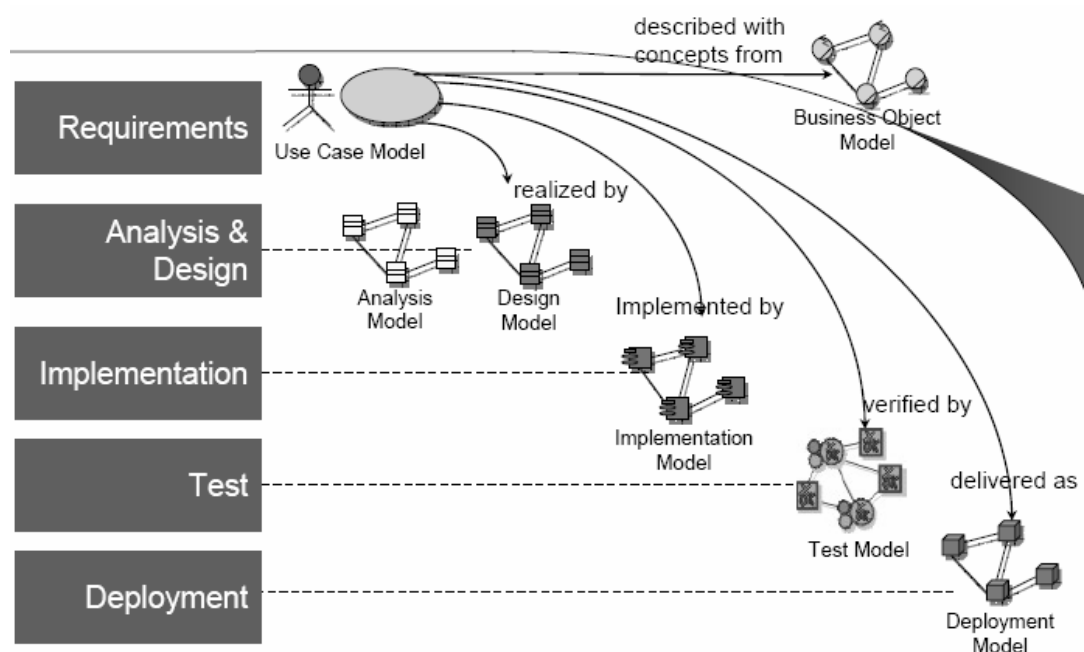


Figure 2-6: RUP est basé sur les cas d'utilisation [Jacobson 1999]

Le modèle des cas d'utilisation est un produit de travail en soi, mais il peut être inclus dans un autre produit de travail appelé *Software Requirements Specification* ou spécification des exigences du logiciel. Celui-ci est également fourni par la discipline « Requirements » et son but est de recenser et d'organiser toutes les exigences du système ou d'une partie du système, que ce soit des exigences fonctionnelles (les fonctionnalités) ou non-fonctionnelles (performance, sécurité, ... etc.). La spécification, est utilisée tant par les designers et les implémenteurs que par les gestionnaires de projet ou par les testeurs.

3 Les ordres boursiers à cours limité

3.1 Définition

Une des modalités à connaître lorsqu'on souhaite passer un ordre boursier réside dans le type d'ordre. Il existe, entre autres, le type d'ordre boursier dit *à cours limité*. Ce type d'ordre permet de spécifier à l'avance le prix qu'on souhaite payer à la négociation d'un titre boursier. Par exemple, si le cours d'un titre est CHF 60 et on place un ordre à cours limité à CHF 55, ça voudrait dire qu'on aimerait acheter ce titre seulement au prix inférieur ou égal à CHF 55. Si le cours du titre baisse à ce seuil, on achètera le titre. Sinon, l'achat n'aura pas lieu. Le même principe peut s'appliquer aussi à la vente d'un titre boursier.

L'ordre à cours limité ne garantit pas que l'exécution ait lieu. Par contre, en cas d'exécution en totalité, l'investisseur connaît à l'avance ses gains ou ses pertes.

3.2 Risques

L'ordre à cours limité a toutefois un écueil important : le risque de fractionnement. En effet, en fixant une limite à un ordre, on diminue sa probabilité d'être exécuté. Plus la limite fixée est élevée, moins l'ordre a de chance d'être exécuté en une seule fois. Ainsi, il se peut que l'ordre soit exécuté sur plusieurs fois, en multipliant les frais de courtage [Edu Bourse 2007].

3.3 L'intérêt des ordres à cours limité pour les membres de la BSU

Il est pratique pour l'investisseur d'utiliser le cours limité quand il place un ordre après la fermeture du marché financier. Ce genre d'ordre est exécuté le lendemain et un titre peut avoir un prix d'ouverture sérieusement différent du prix de clôture. Dans le jeu PMS, l'exécution des ordres des joueurs se fait une seule fois par jour au cours de clôture. Celui-ci peut varier fortement du prix en vigueur au moment du placement de l'ordre et peut créer des surprises. Pour cette raison, ce serait intéressant pour les membres PMS d'avoir la possibilité de passer des ordres à cours limité pour mieux pouvoir gérer leurs portefeuilles fictifs. La simulation boursière PMS gagnerait en attractivité et les joueurs auraient la possibilité de négocier de façon plus proche de la réalité boursière.

4 Architecture du système d'information BSU

Le SI BSU actuel est basé sur un ensemble de serveurs et de PCs connectés au réseau de l'Université de Fribourg. Dans le modèle simplifié montré par la Figure 4-1, on distingue les composantes suivantes :

- Le site web BSU : Il constitue la plateforme visible pour les joueurs et l'endroit où se déroule le jeu. Il est composé d'une partie publique codée en HTML et Javascript et d'une partie sécurisée dynamique, réservée aux membres et codée en ASP. Le site web se trouve sur un serveur web IIS situé sur un serveur virtuel géré par l'Université de Fribourg.
- Le Programme Access PMS : C'est un programme basé sur « MS Access '97 » ; les administrateurs BSU s'en servent pour configurer le jeu au début de la simulation et pour actualiser les cours des titres, ainsi que les portefeuilles des joueurs durant le jeu. Ce programme est accessible par le PC appelé BSUPC04 ou par le serveur physique appelé BSUPC03.
- La base de donnée BSU : Basée sur un serveur SQL qui se situe sur le serveur physique, elle contient toutes les informations importantes sur le jeu PMS et elle est reliée au Programme Access PMS, par lequel elle est mise à jour et auquel elle fournit les données nécessaires durant le jeu.
- L'unité de communication avec la bourse Suisse, Swiss Exchange (SWX) : C'est un ensemble de programmes rédigés dans le langage Java, qui servent à importer les cours réels des titres boursiers. Ils sont initiés de façon automatique par des automates (*scheduled tasks*) sous Windows.
- L'entrepôt de données BSU : Basé sur le programme Microstrategy, c'est un outil d'analyse des données de la simulation boursière, que les membres BSU peuvent consulter par le site web. Il communique avec la base de données BSU par un processus ETL (extracting, transforming, loading). La configuration se fait à partir de Microstrategy Desktop, installé sur le serveur BSUPC03.

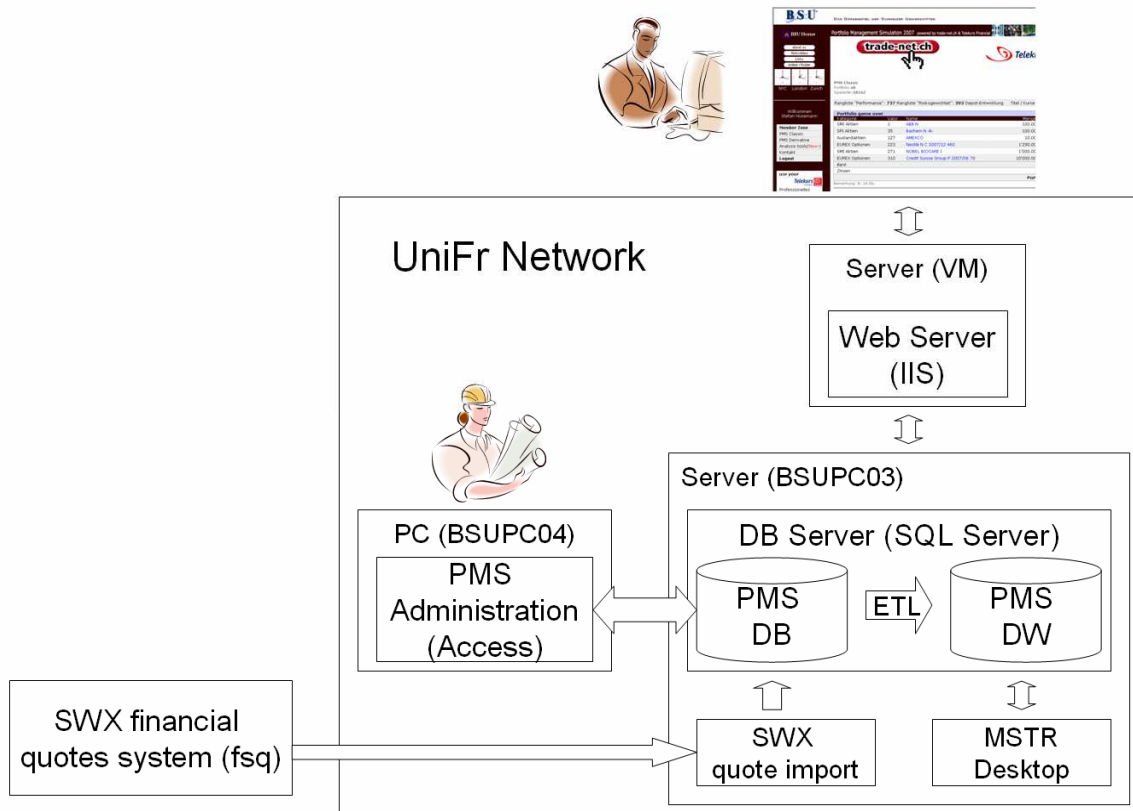


Figure 4-1: Architecture des composantes du système d'information BSU [Hüsemann 2007]

5 Analyse du SI BSU et extension du modèle avec une nouvelle fonctionnalité.

Dans ce chapitre, la description des fonctionnalités existantes et celles souhaitées pour le système d'information BSU sera présentée à l'aide du modèle du produit de travail « Software Requirements Specification » de la discipline « Requirements » qu'on trouve dans RUP (voir chapitre 2.2.3 *Le modèle des cas d'utilisation dans RUP*). Les diagrammes des cas d'utilisation ont été élaborés avec le programme « MagicDraw UML 12.0 ». La structure du document présenté ci-dessous a été simplifiée et modifiée par rapport au modèle original.

5.1 Software Requirements Specification

5.1.1 Introduction

5.1.1.1 *Objectif :*

Le but de ce document est de recenser et de décrire les fonctionnalités existantes d'une partie du système d'information BSU et celles qu'on aimerait avoir dans le futur, ainsi que d'autres exigences non-fonctionnelles envers lui.

5.1.1.2 *Cadre :*

Cette spécification contient le modèle des cas d'utilisation qui représente principalement les fonctionnalités liées aux activités de négociation dans le jeu PMS et dans une moindre mesure celles liées à l'inscription des membres, à la mise en place et à l'actualisation du jeu.

5.1.1.3 *Définitions, acronymes et abbréviations :*

Admin – espace administrateur

Administrateur – toute personne autorisée à administrer le jeu

Automate – Tâche programmée dans Windows qui lance des demandes telnet vers la SWX pour l'importation des cours des titres, à l'aide des programmes en Java.

Calcul du jeu – importation des cours de clôture des titres dans le programme PMS et actualisation des portefeuilles des membres.

Member zone – Partie du site web BSU où les membres peuvent négocier.

Négociation – passation d'un ordre boursier dans le jeu PMS

Période – un jour ouvrable de la bourse pendant le jeu PMS

Classic – la version classique du jeu, où l'on peut négocier des actions, des obligations, des options, des devises et des fonds.

Derivative – la version dérivative du jeu PMS, où seulement des options peuvent être négociées.

Joueur – toute personne ayant un compte dans la base de donnée BSU.

5.1.2 Description

5.1.2.1 Etude du modèle des cas d'utilisation

En prenant en considération le système d'information BSU du point de vue de l'utilisateur, trois acteurs ont été identifiés : **l'Administrateur, le Joueur et l'Automate**.

En outre, quatre types de fonctionnalités recensées ont été regroupés sous forme de cas d'utilisations dans quatre paquets principaux : **Admin, Calcul, Inscriptions et Négociation**.

Les diagrammes du modèle des cas d'utilisation sont donnés par les Figures Figure 5-1, Figure 5-2, Figure 5-3 et Figure 5-4. Pour la spécification des cas d'utilisation on se réfère au chapitre 5.2. *Spécification des cas d'utilisation*. Une liste des noms des cas d'utilisation avec une brève description pour chacun est donnée ci-dessous :

Admin :

1. (UC1) Actualiser le site web (par l'Administrateur)
2. (UC2) Mettre le jeu en place (par l'Administrateur au début de la simulation)
3. (UC3) Configurer connection vers la SWX
4. (UC26) Importer les cours de SWX (par l'Automate)

Calcul :

1. (UC4) Corriger erreurs (une *généralisation*¹ des cas d'utilisation UC5 et UC6, associés à l'Administrateur)

¹ Une relation généralisation/spécialisation entre les cas d'utilisation en UML est indiquée par une flèche avec le triangle vide. Dans le cas présent, cela veut dire que les cas d'utilisation « Corriger erreurs de transactions » et « Corriger erreurs de cours » sont tous les deux des cas spécialisé du cas « Corriger erreurs ».

2. (UC5) Corriger erreurs de transactions
3. (UC6) Corriger erreurs de cours (en cas de fausses importations depuis la SWX)
4. (UC7) Calculer periode (une généralisation des cas d'utilisation UC8 et UC9, associés à l'Administrateur qui déclenche ce processus sur le Programme Access PMS)
5. (UC8) Calculer période actuelle (lancer le calcul des portefeuilles pour la dernière période écoulée)
6. (UC9) Calculer plusieurs périodes (recalculer depuis la période souhaite, en cas de modifications de transactions passées ou de cours anciens)
7. (UC10) Calculer classements (un cas d'utilisation qui peut être *utilisé* par le cas d'utilisation « Calculer période ».

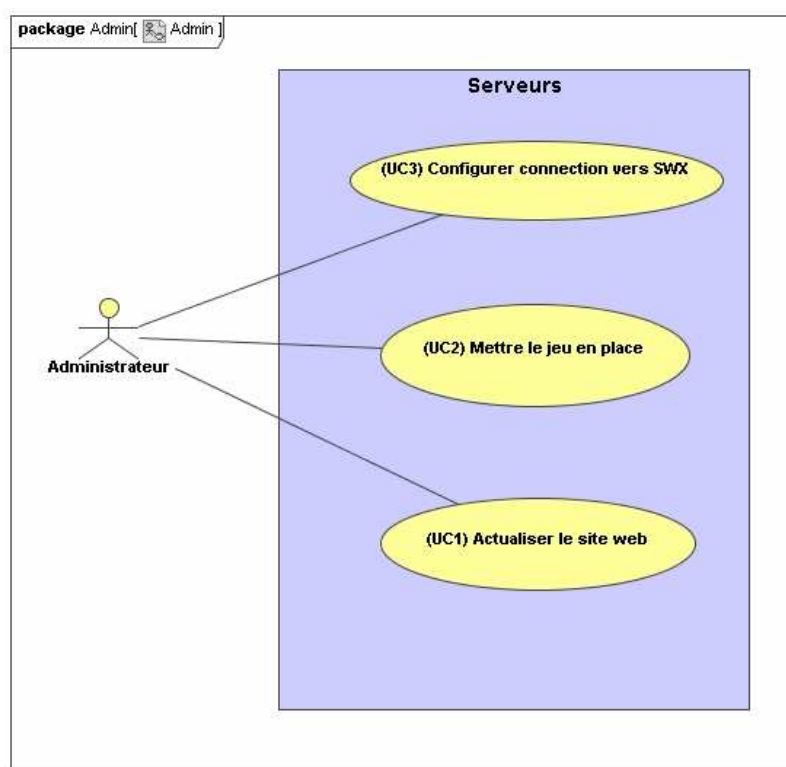


Figure 5-1: Le diagramme « Admin » du paquet **Admin**. Ici, le cas d'utilisation « Configurer connection vers la SWX » appartient au paquet **Calcul**.

Inscriptions :

1. (UC11) Inscription par le joueur (sur le site web à partir du formulaire d'inscription)
2. (UC12) Récupérer le mot de passe (sur le site web, mot de passe envoyé automatiquement par e-mail)

3. (UC13) Réactiver le compte (les comptes des membres sont généralement désactivées à la fin du jeu et peuvent être réactivées pour un nouveau jeu à travers le site web)
4. (UC14) Inscription par l'admin (l'Administrateur peut toujours inscrire un nouveau compte de joueur dans la base de données BSU par l'interface administrateur sur Access.
5. (UC15) Modifier le statut du compte (l'Administrateur peut désactiver ou bloquer l'accès à un compte à tout moment par l'interface administrateur).

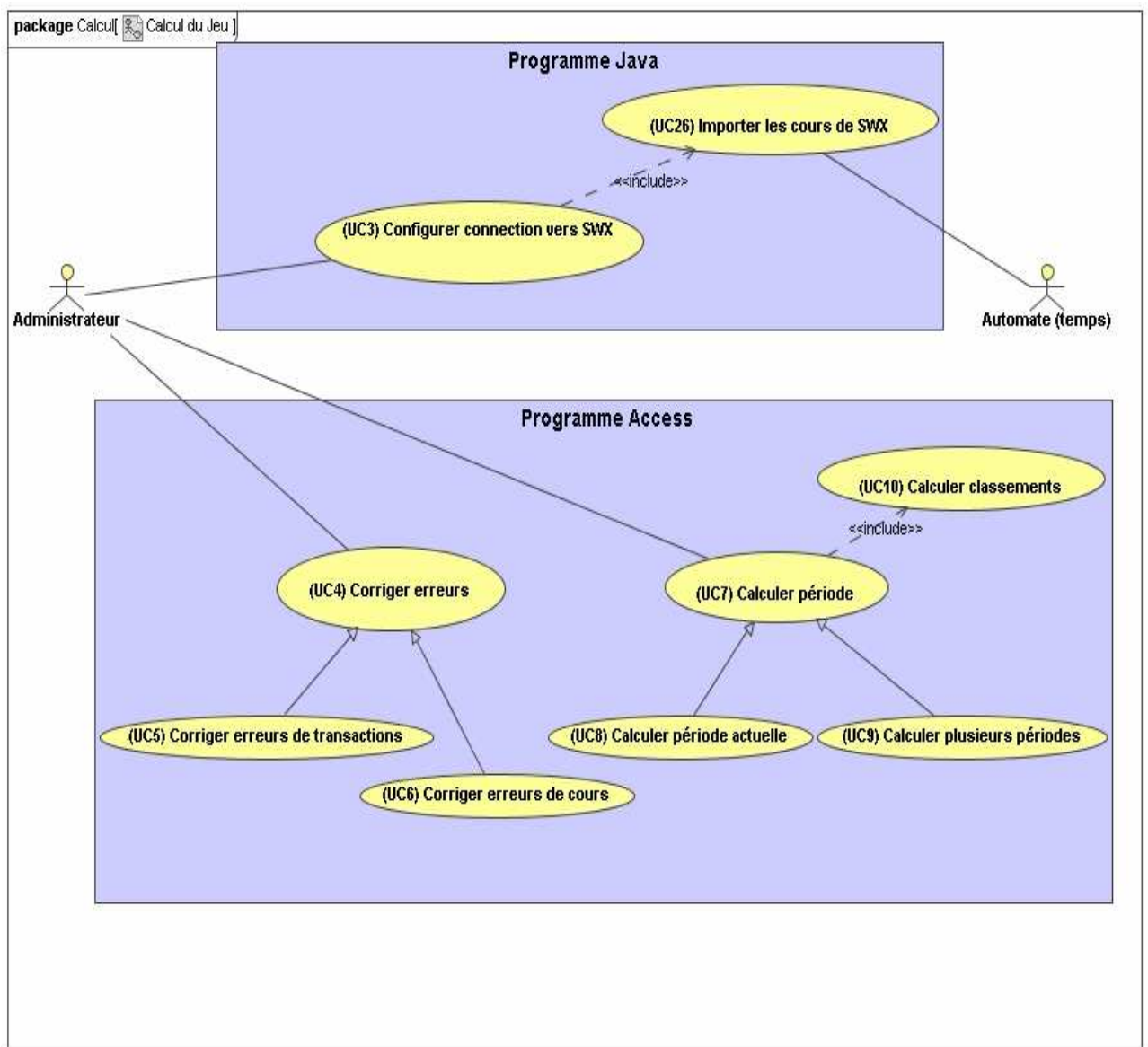


Figure 5-2: Le diagramme « Calcul du Jeu » du paquet **Calcul** avec les deux sous-systèmes où on repère les cas d'utilisation : Programme Java et Programme Access (Programme Access PMS)

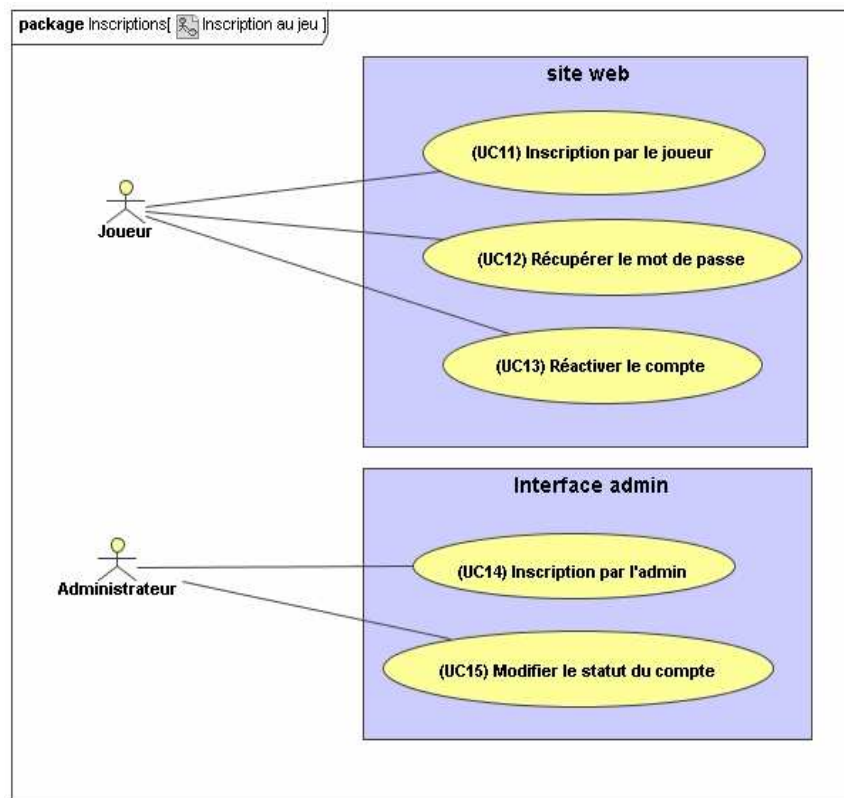


Figure 5-3: Le diagramme « Inscription au jeu » où on distingue les deux sous-systèmes « site web » et « interface admin » avec les cas d'utilisation leur appartenant.

Négociation :

1. (UC16) Consulter le portefeuille (par le Joueur sur le Member zone, comme tous les cas d'utilisation qui suivent)
2. (UC17) Consulter le développement du dépôt
3. (UC18) Consulter le classement (la liste des joueur d'après leur classement est affichable dans la zone sécurisée ; ce cas d'utilisation est *spécialisé* en deux autres cas : le 18 et le 19)
4. (UC19) Consulter classement Classic (pour la version Classic du jeu PMS il y a un classement séparé de celui de la version Derivative ; ce cas est en outre spécialisé en deux autres cas : le UC20 et le UC21)
5. (UC20) Consulter classement Derivative
6. (UC21) Consulter classement Total
7. (UC22) Consulter classement Sharp-Ratio (un classement calculé en utilisant la formule du Sharp-Ration.

8. (UC23) Consulter titres disponibles (des tables avec les noms des titres s'affichent sur le site web)
9. (UC24) Passer un ordre (passer un ordre boursier à tout prix – fonctionnalité existante ; ce cas d'utilisation comprends un point d'extension, qui permet de l'élargir vers un nouveau cas d'utilisation : le cas UC25)
10. (UC25) Passer un ordre avec limite (passer un ordre boursier à cours limité – nouvelle fonctionnalité qu'on souhaite avoir dans le futur ; ce cas d'utilisation est une *extension* du cas UC24 et fait appel au services du cas UC23.)

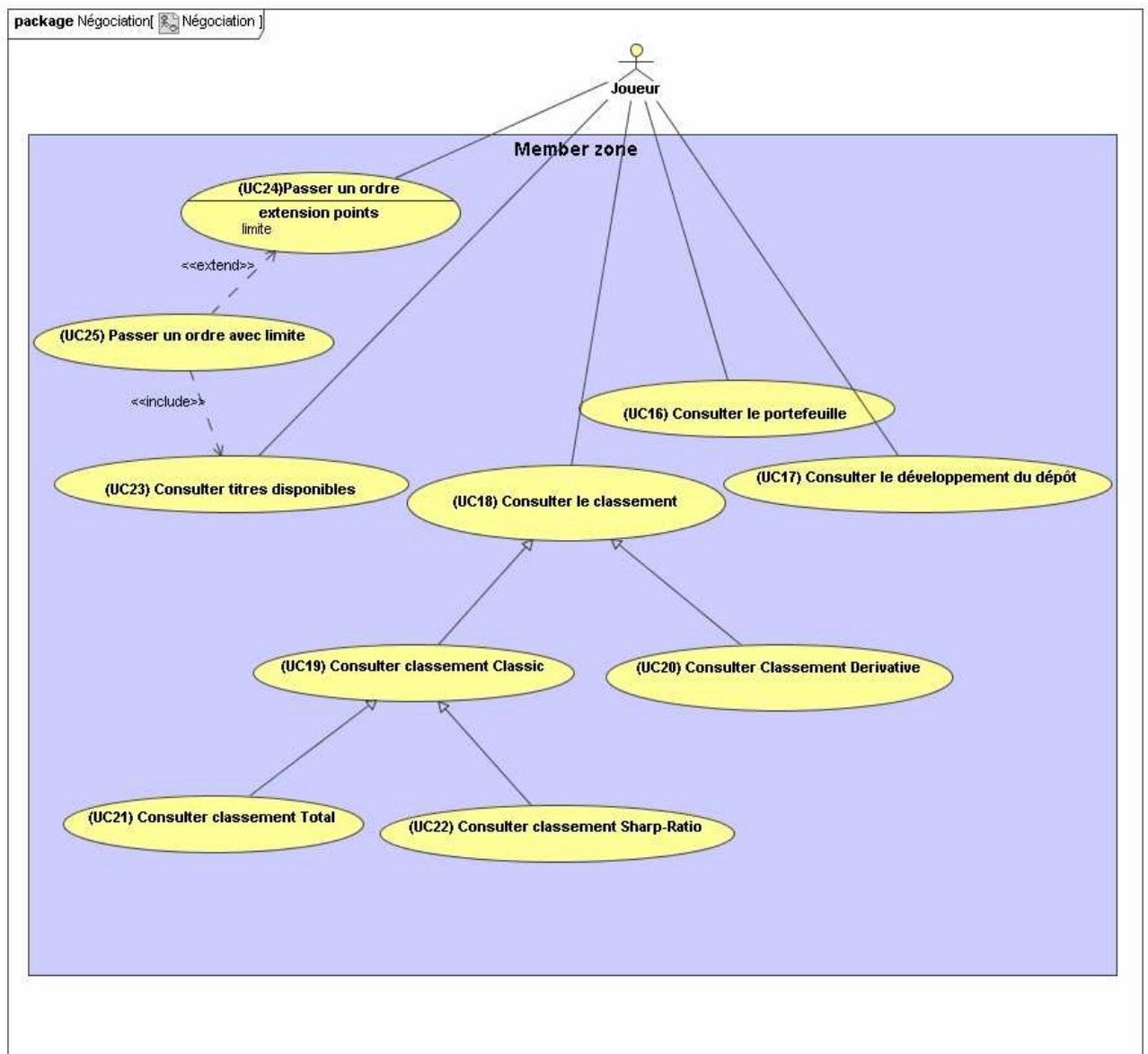


Figure 5-4: Diagramme de cas d'utilisation « Négociation » à l'intérieur du paquet **Négociation**. Passer un ordre avec limite est une nouvelle fonctionnalité qui vient sous forme d'extention de la fonctionnalité existante de passation d'ordre.

5.2 Spécification des cas d'utilisation

Les cas d'utilisation en UML sont généralement accompagnés d'une description textuelle qui suit certaines règles de structuration pour faciliter l'expression, la compréhension et la cohérence. Cette description est aussi appelée spécification du cas d'utilisation dans RUP (*Use-Case Specification*) et elle montre le flux d'évènements décrivant les interactions entre les acteurs et le système. Typiquement, elle contient d'autres informations comme préconditions et postconditions, scénarios clés et règles de gestion. Il est important de décider du degré d'élaboration d'un cas d'utilisation, notamment si on veut décrire tous les cas d'utilisation, si on veut décrire seulement les flux principaux, si on veut décrire de manière complète les préconditions et les postconditions, etc. Ici, on s'intéressera seulement à la description des cas d'utilisations liés à la négociation dans le jeu PMS.

5.2.1 Cas d'utilisation UC24 : Passer un ordre

Objectif : Ce cas d'utilisation traite des actions conduites pour passer un ordre boursier à tout prix dans le jeu PMS.

Acteur : Le joueur

Préconditions :

- Le joueur s'est connecté avec succès à la Member Zone.
- Le jeu est déjà lancé.
- La période pour la quelle le joueur veut passer l'ordre n'est pas encore terminée.

Actions :

1. Scénario principal :

Le cas d'utilisation commence quand un joueur veut passer un ordre.

- 1.1. Choisir la version du jeu désirée (Classic ou Derivative).
- 1.2. Choisir le titre désiré dans le menu déroulant.
- 1.3. Taper la quantité désirée dans la case appropriée.
- 1.4. Indiquer avec « + » ou « - » devant la quantité selon qu'on veut acheter ou vendre.

- 1.5. Cocher ou laisser vide la case « ajuster » si on veut ajuster le montant des achats ou vente d'après la somme d'argent disponible au dépôt.
 - 1.6. Point d'extension « limite », pour implémentations futures.
 - 1.7. Répéter les étapes 1.2. à 1.5. le nombre de fois désiré.
 - 1.8. Cliquer sur « OK ».
2. Scénario alternatif :
- 2.1. Choisir la version du jeu désirée.
 - 2.2. Aller à la page des titres disponibles.
 - 2.3. Choisir le titre désiré.
 - 2.4. Cliquer sur « vendre / acheter » à côté du nom du titre.
 - 2.5. Le cas d'utilisation continue avec le point 1.3. du scénario principal.

Postcondition : Les ordres d'achat et de vente introduits sont sauvgardés dans le système. Les ordres sont prêts à être exécutés au calcul de la période (cf. cas d'utilisation 7. « Calculer période »).

Règles de gestion :

- On ne peut passer des ordres que sur des titres disponibles dans le jeu PMS.
- Les joueurs consultent eux même les cours actuels dans la presse ou les sites boursier.

Point d'extension : Ce cas d'utilisation contient un point d'extension appelé « limite ».

5.2.2 Cas d'utilisation UC25: Passer un ordre avec limite

Objectif : Ce cas d'utilisation traite des actions conduites pour passer un ordre boursier à cours limité dans le jeu PMS. Ce cas est une extension du cas d'utilisation UC24, « Passer un ordre ».

Actions :

1. Scénario principal :
 - 1.6. Choisir le type d'ordre « cours limité ».
 - 1.7. Spécifier le prix limite dans le champs « limite ».

- 1.8. Cliquer sur « OK ».
2. Scénario : la valeur du champs « limite » n'est pas un nombre ou c'est un nombre négatif
 - 2.1. Un message d'erreur apparaît.
 - 2.2. Corriger la valeur dans le champ « limite ».
 - 2.3. Cliquer sur « OK ».

Règles de gestion :

- Les ordres à cours limité seront exécutés si le(s) cours de clôture du (des) titre(s) commandé(s) ne dépasse(nt) pas la limite spécifiée dans le champs « limite ». Si la condition « limite » n'est pas remplie, l'ordre reste dans la liste des ordres à exécuter au calcul de la prochaine période.

Points d'extension : Aucun point d'extension n'existe pour ce cas d'utilisation.

6 Implémentation de la nouvelle fonctionnalité

6.1 « Reengineering » dans le cas BSU

L'équipe BSU à présent, est en train de considérer une restructuration de l'architecture du système d'information et l'utilisation de nouvelles technologies, afin d'améliorer ses performances, sa facilité d'usage et sa convivialité. Ces changements visent entre autres d'améliorer le programme de gestion PMS, de faire migrer l'interface utilisateur vers la technologie ASP.Net, d'adapter la base de donnée, etc. La nouvelle fonctionnalité « Passer un ordre avec limite » pourra être implémentée aux cours de ce processus et en tenant compte des changements visés.

La Figure 6-1 montre un aperçu de l'interface utilisateurs actuelle, notamment de la page de négociation du jeu PMS, avec une proposition de changement très simple (dans le carré rouge) pour tenir compte de la nouvelle fonctionnalité. Suivant le degré de *user-friendliness* que le site sera censé d'atteindre, cette proposition devrait varier.

The screenshot displays the 'BSU Handeln PMS F' interface. At the top, it shows the browser window with the URL 'http://www-bsu.unifr.ch/f/TEST_handeln_test.asp?game=02&session_id=674320630'. The main content area is titled 'PMS Classic' and shows portfolio details for 'Portefeuille: sponsor' and 'No. participant: 60126'. A table lists various assets with columns for 'No. Valeur', 'Catégorie', 'Titre', 'Quantité', 'Cours', and 'Montant'. Below this, there is a section for 'Positions pour game over' with a table containing one row for 'EUREX Optionen' (ABB P 2007/12 16) with a quantity of 30.00. A red box highlights the 'Limit' field, which currently contains '13.50'. Below the table, there is a dropdown menu showing 'Givauden C 2007/09 1200' and a button labeled 'Envoi'.

No. Valeur	Catégorie	Titre	Quantité	Cours	Montant
152	Auslandaktien	HERCULES	1'000.00	23.05	23'050.00
189	Obligationen	FLUGH 5% 95-07	400.00	100.83	40'332.00
191	Devisen	Euro	50'000.00	1.65	82'687.50
193	Devisen	US Dollar	60'000.00	1.22	73'416.00
280	Fonds	UBS-ETF SMI	1'000.00	95.64	95'640.00
281	Fonds	UBS-ETF DJ US LG CAP	1'050.00	78.21	82'120.50
282	Fonds	ZKB GOLD ETF	10.00	26'140.00	261'400.00
283	Fonds	UBS-ETF DJ JAPAN 100	1'020.00	77.80	79'356.00
284	Fonds	iShs FT/EP EUR Prop	1'400.00	60.21	84'294.00
285	Fonds	iSHARES FTSEUROFIRST 100	1'500.00	53.96	80'940.00
286	Fonds	IMMOFONDS	300.00	355.00	106'500.00
Banque					826.49
Intérêts					35.46
Valeur du portefeuille					1'010'597.95

No. Valeur	Catégorie	Titre	Quantité(+/-)	Limit	Ajuster
304	EUREX Optionen	ABB P 2007/12 16	30.00	13.50	<input checked="" type="checkbox"/>
		Givauden C 2007/09 1200	-40		

Figure 6-1: Un aperçu imaginable de la page de négociation dans le jeu PMS avec la possibilité de spécifier le prix limite auquel on souhaite négocier un titre.

7 Conclusion

Nous vivons dans une époque de développements technologiques et d'innovation continue dans le domaine du traitement de l'information. Les systèmes d'information sont des entités en métamorphose continue et leur évolution est indispensable afin de répondre aux besoins changeants ou émergents des utilisateurs finaux. Qu'on décide de créer des solutions toutes nouvelles ou de construire à partir de solutions déjà existantes, le besoin de bien connaître le fonctionnement des systèmes présents ou de ceux à venir reste le même. Pour aider dans la réalisation de cette tâche complexe, il existe aujourd'hui des outils sophistiqués fournissant des lignes de conduite et des moyens pour mieux analyser et décrire les SI. De tels outils sont la méthode RUP et le langage UML.

A l'aide de l'approche proposée par RUP et au moyen des diagrammes UML de cas d'utilisations, on a réussi à cerner le système d'information BSU du point de vue utilisateur et d'avoir donc une meilleure idée de son mode de fonctionnement actuel, afin d'y rajouter une ou plusieurs nouvelles fonctionnalités.

Un outil précieux pour la réalisation du modèle des cas d'utilisation était le programme MagicDraw UML, qui s'est révélé assez facile d'utilisation et flexible, tout en restant efficace.

8 Références

[Edu Bourse 2007] Edu Bourse : Ordre à cours limité : le plus populaire,
<http://www.edubourse.com/guide/guide.php?fiche=ordre-cours-limite>, Décembre 2007.

[Hüsemann 2007] Hüsemann, Stefan: BSU IT Architecture. Présentation, Octobre 2007.

[IEEE Std. 830 – 1993] IEEE Std.: Guide to Software Requirements Specification, 1993.

[Jacobson 1999] Jacobson, Ivar; Booch, Grady; Rumbaugh, James: The Unified Software Development Process. Addison-Wesley, 1999.

[Kruchten 2001] Kruchten, Philippe: What is the Rational Unified Process?, 2001.

[Morley 2006] Morley, Chantal; Hugues, Jean; Leblanc, Bernard: UML 2 – Pour l’analyse d’un système d’information. Dunod, 3e éd., 2006.

[RUP 2007] IBM® Rational Unified Process®:
<http://diuf.unifr.ch/is/courses/files/information-systems-II-SS2007/RationalUnifiedProcess/index.htm> , Décembre 2007.

[Standish Group 2001] Standish Group: Extreme Chaos, 2001.

[Wikipedia 2007] Wikipedia: Unified Process,
http://fr.wikipedia.org/wiki/Processus_unifi%C3%A9 , Décembre 2007.