



Université du Québec à Montréal

Sujet

CALCUL AVEC ISO 19761 DE LA TAILLE
DE LOGICIELS DEVELOPPES SELON
RATIONAL UNIFIED PROCESS

Mémoire présenté
comme exigence partielle
de la maîtrise en génie logiciel

PAR

SAADI AZZOUZ

JUILLET 2003

Remerciements

Je tiens à remercier le Dr Alain Abran, Professeur à l'École de Technologie Supérieure, Directeur de mon mémoire. Par les conseils éclairés qu'il m'a prodigué à chaque fois que c'était nécessaire, par son suivi permanent tout au long du déroulement du projet, j'ai bien mené à terme les objectifs fixés et par là donc à contribuer à l'automatisation du calcul de la taille fonctionnelle avec la méthode COSMIC-FFP (ISO 19761).

M. Philippe Kruchten, directeur de développement de processus à Rational Software Canada, est vivement remercié pour son aide et ses commentaires constructifs.

Mes chaleureux remerciements vont droit à ma conjointe Nora qui a su me soutenir pendant les moments les plus difficiles. Bien qu'elle-même s'occupait à son propre programme de formation, elle gardait toujours un œil sur mon projet.

Aussi, tous ceux qui, de près ou de loin ou même de très loin, pour faire allusion aux familles qui sont à l'autre bout du monde, sont vivement remerciés pour leurs encouragements toujours renouvelés et toute la confiance qu'ils mettent en moi.

Je dédie ce mémoire à mon petit garçon Mohamed - Amine

Saadi Azzouz

Table des Matières

Remerciements.....	2
Liste des figures.....	4
Liste des tableaux.....	4
Liste des abréviations, sigles et acronymes.....	4
Résumé.....	5
<hr/>	
<u>Chapitre 1 : Résumé de l'expression des besoins des utilisateurs</u>	6
1.1 Introduction.....	6
1.2 Description sommaire de l'organisme d'accueil.....	6
1.3 Description des besoins à combler.....	7
1.4 Définition du mandat alloué.....	8
1.5 Entente tripartite des intervenants.....	10
<hr/>	
<u>Chapitre 2 : Résumé des concepts de base de la méthode COSMIC -FFP</u>	11
2.1 Introduction à la taille fonctionnelle	11
2.2 Principes généraux de COSMIC -FFP	11
2.3 La phase d'arrimage	12
2.5 La phase de mesure	13
2.6 Résumé de la procédure de mesure.....	14
<hr/>	
<u>Chapitre 3: Résumé du processus RUP et définition de l'activité de mesure</u>	16
3.1 Introduction	16
3.2 Définition	16
3.3 Les meilleures pratiques de développement et RUP	16
3.4 Structure du processus RUP	16
3.5 L'activité de mesure	19
3.6 Conclusion.....	21
<hr/>	
<u>Chapitre 4 : Concepts de base de l'outil de mesure automatique</u>	22
4.1 Introduction.....	22
4.2 Rapprochement des concepts COSMIC -FFP de la notation UML	22
4.3 Les différents niveaux d'abstraction de mesure	27
4.4 Vérifications.....	29
<hr/>	
<u>Chapitre 5 : Développement de la solution proposée</u>	30
5.1 Les cas d'utilisation du système	30
5.2 Description des cas d'utilisation	30
5.3 Liste des scénarios	34
5.4 Description des scénarios	35
5.5 Résumé des traitements à effectuer	35
5.6 Description de la base de données	35
5.7 Description des interfaces utilisateur.....	36
<hr/>	
<u>Chapitre 6 : Analyse des résultats obtenus</u>	37
6.1 Notions de la méthode COSMIC -FFP prises en charge dans le projet	37
6.2 Analyse des résultats de l'étude de cas " Rice Cooker "	38
6.3 Analyse des résultats de l'étude de cas " Valve control "	38
6.4 Conclusion de l'analyse	39
6.5 Limites du logiciel	39

7. Conclusion générale	40
8. Annexes	41
8.1 Principes et règles de la méthode COSMIC-FFP.....	41
8.2 Procédure d'activation de la propriété «Persistent» d'une classe dans Rational Rose	44
8.3 Utilisation du nouveau stéréotype « Trigger event »	45
8.4 Prototype des interfaces utilisateur.....	45
9. Bibliographie	50

Liste des figures

<u>Figure</u>		<u>Page</u>
2.1	Modèle général de la méthode COSMIC-FFP.....	12
2.2	Éléments de base de la méthode COSMIC-FFP.....	12
2.3	Modèle général de mesure de la méthode COSMIC-FFP	13
2.4	Représentation graphique de la proportion par mouvement de données.....	15
3.1	Schématisation du processus RUP.....	17
3.2	Exemple de définition de points de contrôle.....	19
3.3	Catégorisation des disciplines RUP.....	20

Liste des tableaux

<u>Tableau</u>		<u>Page</u>
1.1	Situation du problème.....	8
2.1	Résultats détaillés par processus fonctionnel.....	14
2.2	Résultats agrégés par processus fonctionnel et par mouvement de données	15
2.3	Résultats agrégés par proportion de chaque processus fonctionnel	15
2.4	Résultats agrégés par proportion de chaque type de mouvements de données	15

Liste des abréviations, sigles et acronymes

Attr	Attribut d'un Groupe de Données (Data Group Attribute)
CFSU	Cosmic Functional Size Unit
COSMIC-FFP	Common Software Measurement International Consortium - Full Functional Point
ÉTS	École de Technologie Supérieure
E/X/R/W	Entrée (Entry) / Exit (Sortie) / Lecture (Read) Écriture (Write)
GD	Groupe de données (Data Group)
LOC	Lignes de code (Lines Of Code)
MD	Mouvement de données (Data movement)
MGL	Maitrise en génie logiciel
PID	Process identification
PF	Processus Fonctionnel (Functional Process)
RAM	Random Access Memory
REI	Rose Extensibility Interface
ROM	Read Only Memory
RUP	Rational Unified Process
SFSU	Senario Functional Size Unit
UC	Cas d'utilisation (Use Case)
UFSU	Use case Functional Size Unit
UML	Unified Modeling Language
UQAM	Université du Québec à Montréal

Résumé

Ce projet de mémoire a pour objectif principal l'automatisation du calcul de la taille fonctionnelle avec la méthode COSMIC-FFP pour des logiciels développés selon le processus RUP. Le calcul peut se faire à trois niveaux d'abstraction du cycle de développement du logiciel :

1. Le premier calcul évalue la taille fonctionnelle à un niveau d'abstraction très élevé. Cela permet d'évaluer l'importance du logiciel en terme du nombre de cas d'utilisations qu'il renferme, du nombre d'acteurs et du nombre d'interactions mises en oeuvre.
2. Le deuxième niveau d'abstraction permet d'évaluer l'importance du logiciel en terme du nombre total de scénarios et du nombre d'objets participants à la réalisation des cas d'utilisations.
3. Le troisième niveau d'abstraction permet de mesurer la taille du logiciel en terme du nombre total de fonctions qu'il réalise, soit la taille fonctionnelle qui est caractérisée par le nombre d'échange de données entre l'enceinte du logiciel et son environnement, selon la norme ISO 19761.

La procédure de calcul automatique de la taille fonctionnelle se base sur les artefacts conceptuels du logiciel à mesurer, principalement les diagrammes de cas d'utilisation et les diagrammes de séquence. Ces artefacts sont les résultats des phases d'analyse et de conception et sont produits avec l'outil Rational Rose. Il a été donc impératif de faire un rapprochement entre les concepts de base UML (sur lesquels est basé l'outil Rational Rose) et les notions de la méthode COSMIC-FFP (ISO 19761).

Aussi, pour extraire les propriétés des artefacts du logiciel à mesurer, il a été fait usage du langage de programmation REI (Rose Extensibility Interface) qui est interne à Rational Rose.

L'outil de calcul automatique peut être utilisé tout au long du cycle de développement du logiciel selon le processus RUP. Les phases où il est intéressant de faire appel aux services de l'outil sont multiples, incluant en particulier l'étape de planification de projet.

1.1 Introduction

Ce projet intervient dans le cadre de la discipline du génie logiciel et traite particulièrement de l'aspect important qu'est la mesure de la taille du logiciel. La taille d'un logiciel peut s'appréhender de différentes façons, entre autres en terme de lignes de code ou en terme des fonctionnalités réalisées. Pour ce projet, il est fait usage de cette dernière alternative qui est utilisée pour évaluer la taille fonctionnelle d'un logiciel aux premières phases de son développement. La méthode COSMIC-FFP (ISO 19761)⁽¹⁾ est la méthode de mesure choisie.

Voici les points qui seront développés afin de cerner la problématique :

- Description sommaire de l'organisme d'accueil qu'est la compagnie Rational Software ainsi que le processus RUP sous-jacent.
- Définition des besoins à combler par un énoncé du problème et son positionnement dans l'environnement d'accueil.
- Définition du mandat et des étapes à suivre pour satisfaire les besoins à combler.
- Définition de l'entente tripartite entre les différents intervenants du projet.

Par ailleurs, afin de réduire les risques majeurs encourus par ce projet, une étude de faisabilité de l'utilisation de l'outil REI (Rose Extensibility Interface) a été réalisée en vue d'extraire les différents artefacts de l'environnement Rational Rose. En effet, l'extraction des informations concernant principalement les cas d'utilisations et les diagrammes de séquences constitue la clé de voûte quant à la réalisation de ce projet, il est donc important de s'en assurer de la faisabilité avant d'aller plus loin dans de longs développements.

Note (1) : ISO 19761: 2003 COSMIC-FFP Une mesure de taille fonctionnelle: Cette méthode de mesure a été développée initialement en 1997 par le groupe de chercheurs du Dr Abran à l'UQAM, puis adoptée par le groupe international COSMIC en 1999, pour être finalement adoptée comme norme ISO en février 2003.

1.2 Description sommaire de l'organisme d'accueil

1.2.1 Organisme d'accueil

L'organisme directement intéressé par ce projet est la compagnie Rational Software qui est représentée par M. Philippe Kruchten, directeur de développement de processus à Rational Software Canada¹. Cette dernière est très impliquée dans l'aspect important du génie logiciel qu'est le processus logiciel. Elle œuvre dans ce domaine par l'entremise de son célèbre processus RUP (Rational Unified Process) qui est, en partie, bâti sur le standard UML, norme incontournable de modélisation.

Cette compagnie est implantée presque partout dans le monde. À la fin de l'année 1999, plus de 1000 compagnies utilisent le processus RUP dans divers domaines, que ce soit pour de petits ou de grands projets.

1.2.2 Processus sous-jacent

Le processus sous-jacent est dénommé RUP, un processus unifié, résultant de maintes alliances des auteurs des meilleures façons de faire en terme de conduite de processus. L'un des principaux slogans de RUP est l'application des bonnes pratiques de développement. En effet, RUP permet de savoir à tout

¹ Rational Software a été acheté par IBM en avril 2003, et est maintenant complètement intégré à IBM.

moment **qui** fait **quoi**, **comment** et **quand**. Il guide ainsi les équipes informatiques tout au long des différentes phases du cycle de vie du logiciel en les assistant dans l'application des bonnes pratiques de développement, telles que :

- ⊙ Le développement itératif et incrémental,
- ⊙ L'importance qui est accordée à la gestion des exigences et l'appréhension des risques majeurs en aval du processus,
- ⊙ L'évaluation continue de la qualité du logiciel et le contrôle rigoureux des changements,
- ⊙ Le développement visuel et l'importance accordée à l'aspect architectural du produit qui est bâti sur le concept de composants.

1.3 Description des besoins à combler

Afin de cerner tous les aspects d'un projet en génie logiciel, le processus RUP adresse un ensemble d'activités bien ciblées qui sont en l'occurrence : la définition du modèle d'affaire, la gestion des exigences, l'analyse et la conception, l'implantation de la solution, les différents tests et finalement le déploiement sans oublier les trois disciplines de support à savoir : la gestion de projet, la gestion de configuration et de changement ainsi que l'étude de l'environnement.

Toutefois, le chef de projet n'a aucun moyen qui lui permettrait d'évaluer la taille du logiciel en cours de développement. Cette notion est très importante, en effet, l'évaluation de la taille du logiciel dans les premières phases du cycle de développement, ne serait-ce qu'approximative, permet de se préparer en conséquence en terme de la planification de projet, des moyens à mettre en œuvre, de l'effort à déployer, etc.

Ce projet se propose donc de combler ce manque dans le contexte du processus RUP. Avant d'explicitier en détail la solution proposée, voici l'énoncé du problème formulé selon le modèle préconisé dans le prototype du « Vision document » ainsi que son positionnement dans l'environnement en question.

1.3.1 Énoncé du problème

Le problème de	l'inexistence d'outils permettant de mesurer la taille des logiciels en utilisant une norme standard, et en particulier dans le contexte du processus de développement très utilisé comme le Rational Unified Process
affecte	principalement les gestionnaires de projets, les développeurs et les spécialistes de la mesure.
Cela est la cause principale	des coûts élevés et du temps consacré lors d'une mesure manuelle bien que les résultats obtenus manquent souvent de fiabilité.
La solution proposée a comme lignes directrices :	<ul style="list-style-type: none"> ⊙ La définition d'une nouvelle activité de mesure de la taille fonctionnelle qui sera intégrée avec les quatre disciplines du processus RUP. ⊙ La conception d'un outil permettant d'assister le personnel qui effectue cette nouvelle activité de mesure. L'outil leur fournira une évaluation de la taille des unités fonctionnelles et emmagasiner un historique des mesures effectuées. ⊙ Permettre aux gestionnaires d'estimer l'effort de développement dès les premiers cycles de vie du logiciel à partir des mesures de base effectuées sur la taille. ⊙ Permettre une diminution de l'intervention des experts de la mesure de la taille fonctionnelle. ⊙ Prédire automatiquement la taille du logiciel au fur et à mesure de l'avancement du projet. ⊙ Permettre ainsi, une réduction des coûts, du temps alloué à l'opération de mesure ainsi que le taux d'erreurs induit par une mesure manuelle.

1.3.2 Positionnement du produit

Pour	les gestionnaires de projets, les développeurs et les spécialistes de la mesure
qui	souhaitent se faire aider lors de la mesure de la taille de logiciels à n'importe qu'elle phase du cycle de développement d'un projet conduit selon le processus RUP,
ce produit	sera d'une grande utilité : d'une part, il prendra en charge cette tâche en fournissant les directives nécessaires afin de mener à bien cette nouvelle activité de mesure et d'autre part, en mettant à leur disposition un outil de calcul automatique de la taille des unités fonctionnelles.
Comparé l'outil proposé	au calcul manuel, sera plus rapide, plus précis et moins coûteux.

1.4 Définition du mandat alloué

1.4.1 Introduction

Le mandat alloué à ce projet se subdivise principalement en deux parties indépendantes mais complémentaires :

- ⊙ D'une part, il sera question de définir les bases conceptuelles d'une nouvelle activité de mesure au sein du processus RUP. Concrètement, cette activité consistera à définir :
 - Les phases du processus RUP où cette activité interviendra ainsi que les rôles qui assumeront ces nouvelles responsabilités,
 - Les artefacts à mettre en œuvre (anciens et / ou nouveaux) et les procédures correspondantes,
 - « Les guidelines, les templates et les Tool Mentors » éventuels à définir.
- ⊙ D'autre part, un logiciel externe au processus sera conçu. Il aura pour objectif d'assister les différents intervenants lors de l'activité de mesure et durant toutes les phases du cycle de développement. Cet outil permettra de calculer automatiquement la taille d'unités fonctionnelles.

Dans le schéma ci-après, la nouvelle **activité de mesure** est mise en évidence : elle est en étroite interaction avec les activités existantes et, par ailleurs, elle fait appel à un outil externe qui calcule la taille de certaines unités fonctionnelles. Cet outil, à son tour, extrait les artefacts de Rational Rose pour effectuer les mesures.

Phases du processus		Création	Élaboration	Construction	Transition
Activités	Autres activités	↕	↕	↕	↕
	Activité de mesure	↕	↕	↕	↕
		↕	↕	↕	↕
Outil externe		Évaluation de la taille des FURs en phase de création	Évaluation de la taille des FURs en phase d'élaboration	Évaluation de la taille des FURs en phase de construction	Évaluation de la taille des FURs en phase de transition
		↕	↕	↕	↕
Rational Rose		Artefacts (Cas d'utilisation, Diagramme de séquence, ...)			

Tableau 1.1 : Situation problème

Les activités à réaliser pour mener à bien ce projet sont subdivisées en quatre phases, chacune d'elles est constituée d'une ou de plusieurs itérations. En voici une description détaillée.

1.4.2 La phase de Création

Cette phase contient une seule itération et a pour rôle principal d'analyser en détail le problème de mesure dans le processus RUP et d'étudier la méthode *COSMIC-FFP* de mesure de la taille fonctionnelle de logiciels. Elle a comme objectifs de :

- ⊙ Délimiter la portée du système : les acteurs, les cas d'utilisation principaux, les interfaces, ...
- ⊙ Lever les ambiguïtés concernant les besoins exprimés :
 - Étude approfondie de la méthode *COSMIC-FFP*,
 - Étude détaillée du processus RUP,
 - Rapprochement entre les concepts de la méthode *COSMIC-FFP* et de la notation UML,
 - Étude détaillée des possibilités offertes par l'interface de programmation REI de Rational Rose.
- ⊙ Réduire les risques majeurs :
 - S'assurer que le rapprochement entre les concepts de la méthode *COSMIC-FFP* et la notation UML est réalisable,
 - Les éléments caractéristiques de la taille du logiciel à différents niveaux d'abstraction du cycle de développement sont identifiables.
 - L'utilisation de l'interface REI permet bien l'extraction des informations du modèle conceptuel du logiciel à mesurer (informations sur les cas d'utilisations, les diagrammes de séquences, les classes, ...)

1.4.3 La phase d'Élaboration

Cette phase est aussi sanctionnée par une seule et unique itération ayant comme objectifs :

- ⊙ Le développement des besoins et des exigences des utilisateurs par :
 - Une identification des principaux cas d'utilisation et leur description détaillée,
 - L'assignation un ordre de priorité aux cas d'utilisation suivant les risques critiques,
 - Mener les activités d'analyse, de conception, d'implémentation et de test des cas d'utilisations principaux.
- ⊙ Vérifier que les risques significatifs identifiés dans la phase de création sont maîtrisés.

1.4.4 La phase de Construction

Dans cette phase, il sera question de construire une première version fonctionnelle du produit. Les principales activités à mener sont les suivantes :

- ⊙ Finaliser la formulation des besoins :
 - Compléter les acteurs et les cas d'utilisation manquants,
 - Détailler les cas d'utilisation,
 - Prototyper les interfaces utilisateur.
- ⊙ Définir le modèle d'analyse complet :
 - Analyser les cas d'utilisation et détailler les scénarios correspondants,
 - Définir les classes, leurs attributs et méthodes et les relations entre-elles.
- ⊙ Concevoir et implanter les cas d'utilisation manquants.
- ⊙ Tester et intégrer le système.

1.4.5 La phase de Transition

Cette phase a comme objectif d'installer le produit dans l'environnement de Rational Rose et de le tester avec les deux études de cas « Rice Cooker » et « Valve Control ». Les principales tâches à mener sont :

- ⊙ Effectuer des mesures manuelles pour évaluer la taille des deux études de cas précédentes,
- ⊙ Appliquer l'outil de mesure automatique aux deux études de cas,
- ⊙ Analyser des résultats obtenus.

1.5 Entente tripartite des intervenants

Le tableau suivant dresse une liste des différents intervenants et leur rôle dans ce projet.

Nom	Description	Responsabilités
<p>M. Marc Bouisset : Superviseur du projet</p> <p>M. Alain Abran : Superviseur technique du projet</p>	<ul style="list-style-type: none"> ⊙ Supervision générale du projet en terme de consistance, de son bon déroulement, d'échéance, etc. ⊙ Supervision des activités afférentes au développement du projet. ⊙ Très bonne connaissance de COSMIC-FFP et de RUP. 	<ul style="list-style-type: none"> ⊙ S'assurer que le projet se déroule conformément aux directives en places. ⊙ S'assurer que le sujet traité est consistant et qu'il est d'un certain apport pour la discipline du génie logiciel. ⊙ S'assurer que le plan de projet est bien respecté.
<p>M. Alain Abran : Spécialiste de la méthode COSMIC-FFP</p>	<ul style="list-style-type: none"> ⊙ Maîtrise de la mise en application des concepts de COSMIC-FFP. 	<ul style="list-style-type: none"> ⊙ S'assurer que les règles et les procédures de mesure de la méthode COSMIC-FFP sont correctement appliquées.
<p>M. Philippe Kruchten : Spécialiste du processus RUP</p>	<ul style="list-style-type: none"> ⊙ Maîtrise du processus RUP et des outils de Rational tels que Rational Rose, ... 	<ul style="list-style-type: none"> ⊙ Assister le développeur dans sa compréhension du processus RUP et de ses particularités.
<p>M. Saadi Azzouz : (Étudiant) Gestionnaire de projet, analyste, développeur et testeur.</p>	<ul style="list-style-type: none"> ⊙ Capture les besoins et les exigences, ⊙ Lève les risques majeurs, ⊙ Planifie l'activité de projet, ⊙ Développe le produit logiciel, ⊙ Planifie et exécute les tests. 	<ul style="list-style-type: none"> ⊙ S'assurer que les exigences spécifiées couvrent tous les besoins du client. ⊙ S'assurer que le projet n'est pas entouré de risques. ⊙ S'assurer que les activités du projet sont bien ordonnancées et se mènent dans le respect de l'échéancier. ⊙ S'assurer que le produit en cours de développement suit bien les étapes prédéfinies. ⊙ S'assurer, tests à l'appui, que le produit rencontre bien les exigences spécifiées.



Ce chapitre présente un résumé de la méthode COSMIC-FFP qui est à la base de ce projet. Ce résumé se veut exhaustif : il traite de toutes les notions de la méthode COSMIC-FFP. Toutefois, il ne prétend pas aller dans le détail de chaque notion; pour plus d'informations, il faut consulter le manuel de mesure de la méthode COSMIC-FFP cité en référence 1.

2.1 Introduction à la taille fonctionnelle

La taille fonctionnelle est une caractéristique importante des logiciels, elle permet de leur associer une mesure indépendamment de toute considération de la technologie mise en œuvre et de tout environnement d'implantation. Les seules informations prises en compte lors de l'opération de mesure sont les requis fonctionnels exprimés en terme des besoins de l'utilisateur.

La taille fonctionnelle est une mesure dite primitive et c'est l'élément de base indispensable pour toute sorte d'estimation (effort, progrès, qualité, ...). Avec cette mesure, on peut savoir à tout moment quel est le degré de contrôle qu'on exerce sur un projet, et ce, concernant particulièrement les points suivants :

- ⦿ L'état d'avancement d'un projet par rapport aux plans projetés, comme c'est le cas du plan de projet et du plan qualité associé ;
- ⦿ L'estimation des coûts engendrés et de l'effort à consentir (ressources humaines à déployer) ;
- ⦿ Comparaison des logiciels entre-eux en terme de grandeur fonctionnelle, ce qui permet entre autre de les classer ;
- ⦿ Dans le domaine industriel, la standardisation de ta taille fonctionnelle (comme c'est le cas de la méthode COSMIC-FFP) permet de faciliter la communication des concepteurs de logiciels entre-eux et avec leurs clients.

Pour mesurer la taille fonctionnelle, plusieurs méthodes se sont succédées. Entre autres, la méthode FPA (Functional Point Analysis) qui a été conçue pour le domaine des applications d'affaire (« MIS »), où le besoin de déplacement de données est accentué. Sa limitation à ce type d'application en fait son principal défaut.

La méthode **COSMIC -FFP** (Full Functional Point) vient combler ce manque en généralisant le calcul de la taille fonctionnelle des applications d'affaire (gestion), temps réel et hybrides de ces deux dernières. Aussi, aucune méthode n'est encore adaptée pour les systèmes spécialisés conçus sur la base d'algorithmes ou de règles mathématiques complexes. Dans de tels cas, COSMIC-FFP permet, toutefois d'intégrer les résultats des calculs réalisés selon les conventions internes spécifiques à l'organisme en question.

2.2 Principes généraux de COSMIC-FFP

Pour répondre aux besoins de calculer la taille fonctionnelle indépendamment de la manière dont les logiciels sont implantés, la méthode COSMIC-FFP se base sur l'expression des exigences fonctionnelles de l'utilisateur (**FURs : Functional User Requirements**) pour réaliser la mesure. Pour se faire, la méthode COSMIC-FFP définit un ensemble de règles et de procédures exhaustives à appliquer afin d'atteindre cet objectif.

Concrètement, cela se réalise en deux phases (voir figure 2.1 suivante). En premier lieu, on procède à un arrimage (« mapping ») des requis fonctionnels (FURs) du logiciel à mesurer dans un modèle générique spécifique à la méthode COSMIC-FFP et qui a la caractéristique importante de se prêter très bien à l'opération de mesure.

Dans une deuxième phase, on procède à la mesure de la taille fonctionnelle proprement dite, et ce, à partir du modèle générique défini dans la phase précédente et par application de règles et de procédures bien précises.

Généralement, un logiciel n'est pas constitué d'une composante unique; dans ce cas, il va falloir délimiter individuellement chaque partie et l'associer à une couche, laquelle sera mesurée individuellement. Par ailleurs, une couche est considérée selon COSMIC-FFP comme utilisatrice d'autres couches, ce qui doit être pris en compte comme opération d'entrée ou de sortie et rentrera dans le compte total du calcul de la taille.

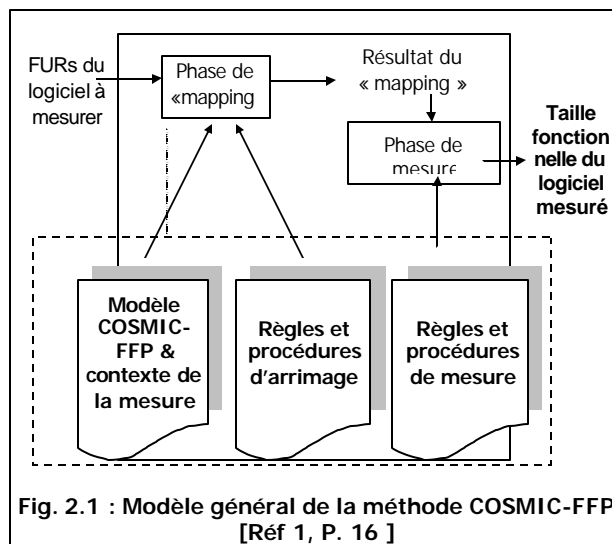


Fig. 2.1 : Modèle général de la méthode COSMIC-FFP [Réf 1, P. 16]

2.3 La phase d'arrimage

La phase d'arrimage (ou « mapping ») consiste à générer un modèle générique spécifique à la méthode COSMIC-FFP. Pour se faire, on applique aux requis fonctionnels (FUR) du logiciel à mesurer un ensemble de règles et de procédures prédéfinies pour identifier les éléments de base indispensables pour l'opération de mesure. Ci-après un résumé exhaustif schématisant les éléments à identifier lors de cette phase d'arrimage.

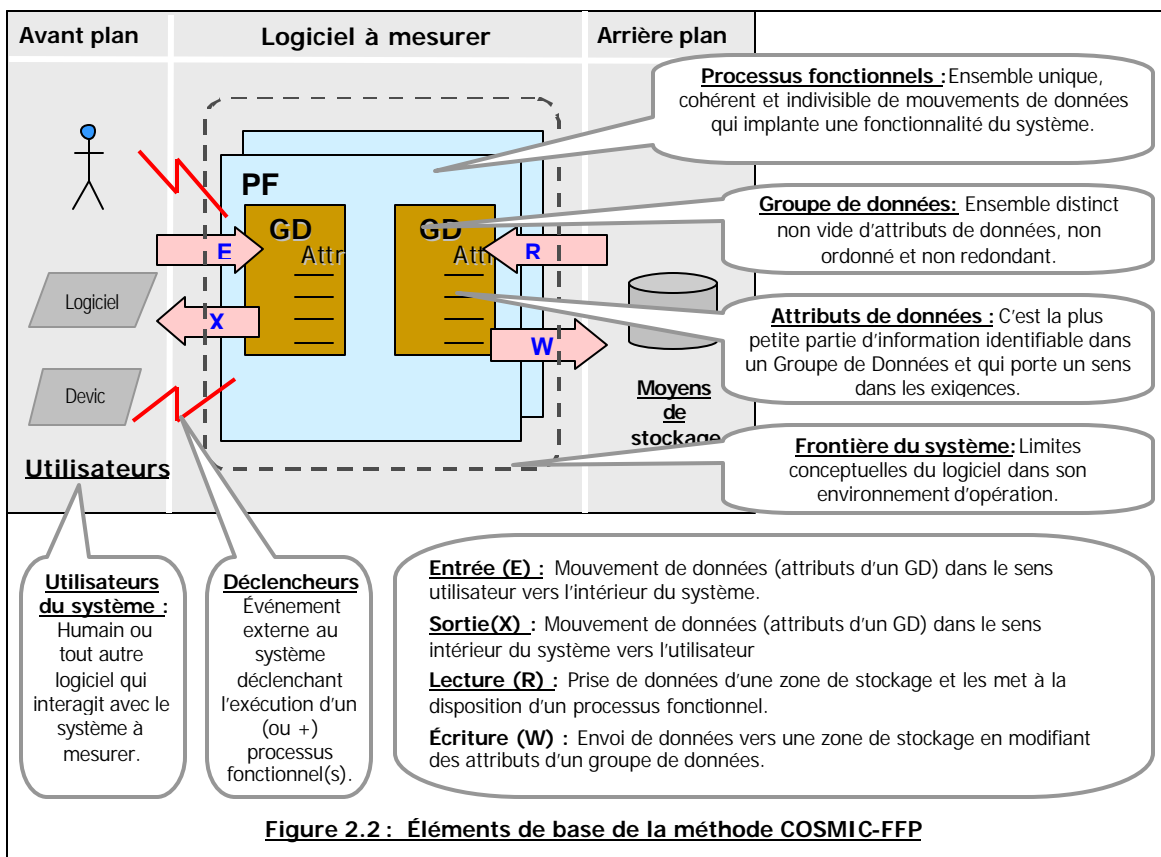


Figure 2.2 : Éléments de base de la méthode COSMIC-FFP

2.3.1 Identification des couches et délimitation de la frontière du logiciel

2.3.1.1 Identification des couches

Un logiciel peut avoir une seule ou plusieurs couches. Dans ce dernier cas, il va falloir les distinguer les unes des autres. Généralement, les requis fonctionnels (FUR) peuvent explicitement fournir des éléments pour repérer les différentes couches. (Voir en annexe 8.1.1, les éléments qui peuvent aider le mesureur pour identifier des couches).

2.3.1.2 Délimitation de la frontière du logiciel

Un élément crucial à considérer de près lors du calcul de la taille fonctionnelle est la délimitation de la frontière du logiciel à mesurer. Tout logiciel a comme frontière des équipements matériels en avant plan (« front end ») tels que le clavier, la souris, l'imprimante, etc. ; et en arrière plan (« back end »), on retrouve les moyens de stockage de données tel que le disque dur, la RAM, la ROM, etc. (Voir en annexe 8.1.2, les règles qui peuvent aider le mesureur pour déterminer la frontière du logiciel).

2.3.1.3 Utilisateur du système

Définition : Un utilisateur (User) du système peut être un être humain, tout autre logiciel ou un dispositif quelconque qui interagit avec le logiciel mesuré.

2.4 Identification des éléments de base candidats

À partir des requis fonctionnels, la deuxième étape du processus de mesure consiste à identifier les éléments de base de l'opération de mesure, à savoir les processus fonctionnels, les déclencheurs et les groupes de données. Ces éléments sont considérés comme candidats tant qu'ils ne sont pas validés selon les règles correspondantes de la méthode COSMIC-FFP. (Voir en annexe 8.1.3, les principes et règles qui peuvent aider le mesureur pour identifier les éléments de base candidats).

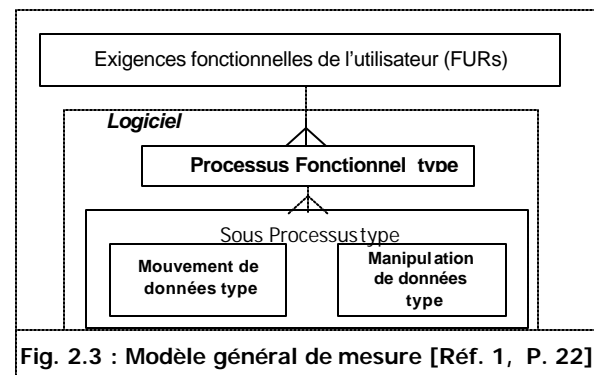


Fig. 2.3 : Modèle général de mesure [Réf. 1, P. 22]

2.5 La phase de mesure

2.5.1 Définition de la mesure

La phase de mesure consiste à appliquer pour le modèle générique COSMIC-FFP défini dans la phase précédente, un ensemble de règles et de procédures bien précises. Ce qui produit une valeur numérique caractérisant la taille fonctionnelle et ayant comme unité le Cfsu (Cosmic Functional Size Unit). Le principe de base qui régit ce calcul est directement lié au nombre de mouvements de données du logiciel.

2.5.2 Méthode et règles à appliquer

La méthode consiste à passer en revue tous les processus fonctionnels validés dans les phases précédentes et d'identifier tous les mouvements de données (Entry, Exit, Read et Write). Un mouvement de données COSMIC-FFP (Data movement) est un déplacement élémentaire de données pendant l'exécution d'un processus fonctionnel. Il y a quatre types de mouvements de données, Voir en annexe 8.1.4, les principes et règles qui peuvent aider le mesureur pour identifier les différents types de mouvements de données.

2.6 Résumé de la procédure de mesure

Étape 1 : Identification de la frontière du logiciel à mesurer

À partir des exigences exprimées par les utilisateurs et des spécifications du système logiciel en terme d'interactions avec son environnement, le mesureur doit identifier la **frontière du logiciel** en question. Il s'ensuit aussi une identification des **utilisateurs** du système.

Étape 2 : Identification des éléments de base candidats

Toujours à partir des requis fonctionnels, la deuxième étape du processus de mesure consiste à identifier les éléments de base de l'opération de mesure, à savoir les processus fonctionnels, les événements déclencheurs et les groupes de données. Ces éléments sont considérés comme candidats tant qu'ils ne sont pas encore validés selon les règles correspondantes de la méthode COSMIC-FFP.

Étape 3 : Validation des processus fonctionnels candidats

Cette étape consiste à valider les processus fonctionnels candidats définis dans l'étape précédente. Pour se faire, on procède à l'arrimage (mapping) des éléments candidats (les processus fonctionnels, les déclencheurs et les groupes de données) selon les règles de la méthode COSMIC-FFP.

Étape 4 : Représentation des résultats détaillés de la mesure

Les résultats obtenus sont répertoriés sous forme d'un tableau (Voir Table 2.1 suivante). Pour chaque processus fonctionnel, il est défini :

- Son rang dans la liste des processus fonctionnels,
- Son identificateur, composé du numéro de couche qui le contient et le rang dans la couche,
- La description littérale du processus,
- Le déclencheur qui est à l'origine du processus,
- La liste de tous les mouvements de données qui le composent, en fournissant :
 - leur description,
 - le groupe de données mis en cause,
 - le type du mouvement de données (Entry, Exit, Read ou Write),
 - la taille associée,
 - la taille totale du processus fonctionnel.
- Dans la section total, on a la taille totale du logiciel.

No	ID du Processus	Description du processus	Déclencheur	Description du mouvement de données	Groupe de données	Type du mouvement de données	FFP	Total
No 1	PID 1	Description 1	Trigger 1	MD 1 MD 2 ...	Data group Data group ...	Type MD1 Type MD1 ...	FFP FFP ...	total
...
No n	PID n	Description n	Trigger n	MD 1 MD 2 MD 3 ...	Data group Data group Data group ...	Type MD1 Type MD2 Type MD3 ...	FFP FFP FFP ...	total

Total d'unités COSMIC-FFP : Total

Table 2.1 : Résultats détaillés par processus fonctionnel

Étape 5 : Représentation des résultats agrégés de la mesure

Dans cette dernière étape, comme le montre les quatre tables suivantes, il est question de résumer les résultats de l'opération de mesure en procédant à des agrégations de la taille fonctionnelle selon :

- Le nombre de mouvements de données de chaque type par processus fonctionnel (Table 2.2),
- La proportion de la taille de chaque processus fonctionnel dans l'application (Table 2.3),
- La proportion de la taille de chaque type de mouvement de données dans l'application (Entry, Exit, Read, Write) (Table 2.4),
- Et, finalement la représentation graphique de la proportion de la taille par type de mouvement de données dans l'application (Figure 2.4).

No	ID du Processus	Description du processus	Mouvements de données				Unités FFP (CFSU)
			E	X	R	W	
No 1	PID 1	Description 1	Total E	Total X	Total R	Total W	Total par Processus
...
No n	PID n	Description n	Total E	Total X	Total R	Total W	Total par Processus

Résumé : X Processus TE TX TR TW Total

Table 2.2 : Résultats agrégés par processus fonctionnel et par mouvement de données

No	ID du Processus	Description du processus	Nombre d'unités FFP (CFSUs)	Proportion dans l'application (%)
No 1	PID 1	Description 1	Nombre 1	Total par Processus
...
No n	PID n	Description n	Nombre n	Total par Processus

Total : Total 100%

Table 2.3 : Résultats agrégés par proportion de chaque processus fonctionnel dans l'application

Mouvements de données COSMIC-FFP	Unités COSMIC-FFP (CFSU)	Proportion (%)
Entry (E)	X	X
Exit (X)	X	X
Read (R)	X	X
Write (W)	x	x

Total : Total CFSUs 100%

Table 2.4 : Résultats agrégés par proportion des mouvements de données dans l'application

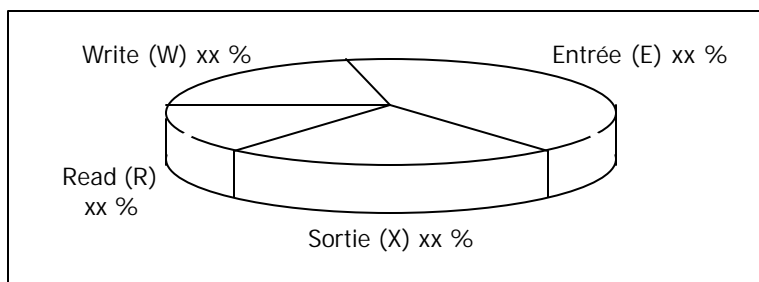


Figure 2.4 : Représentation graphique de la proportion par mouvement de données

3.1 Introduction

Ce chapitre a deux objectifs : d'une part, il présente un résumé des principaux concepts qui régissent le processus RUP; d'autre part, il souligne les disciplines qui peuvent faire appel à l'outil externe de calcul de la taille fonctionnelle et de bénéficiaire ainsi de ce service.

Pour le processus RUP, on se limite à définir les éléments de base indispensables à la compréhension des principes de fonctionnement du processus. Aussi, l'accent sera mis sur les concepts nécessaires et suffisants permettant l'intégration de la nouvelle activité de mesure de la taille fonctionnelle au sein du processus RUP.

3.2 Définition

Rational Unified Process (RUP) est un processus du génie logiciel développé par la société Rational Software. C'est un processus qui est basé sur une approche disciplinée afin de bien maîtriser l'assignation des tâches et la responsabilisation des différents acteurs participant tout au long du cycle de développement du logiciel. RUP a comme objectif principal de faire appliquer les bonnes pratiques de développement aux entreprises, ce qui confère au produit final une meilleure qualité.

RUP présente trois particularités spécifiques dont voici une brève description :

- ⊙ RUP est un produit, c'est à dire qu'il peut évoluer au même titre qu'un logiciel, ce qui lui confère alors la capacité d'évoluer pour rencontrer de nouvelles exigences des utilisateurs du processus.
- ⊙ C'est un processus de type « Framework », c'est à dire qu'on peut le configurer à convenance pour répondre aux besoins spécifiques des organisations en y intégrant les procédures internes par exemple.
- ⊙ Il est distribué sous forme d'un produit web, ce qui permet aux entreprises d'avoir accès à une nouvelle version du produit dès son déploiement sur Internet.

3.3 Les meilleures pratiques de développement et RUP

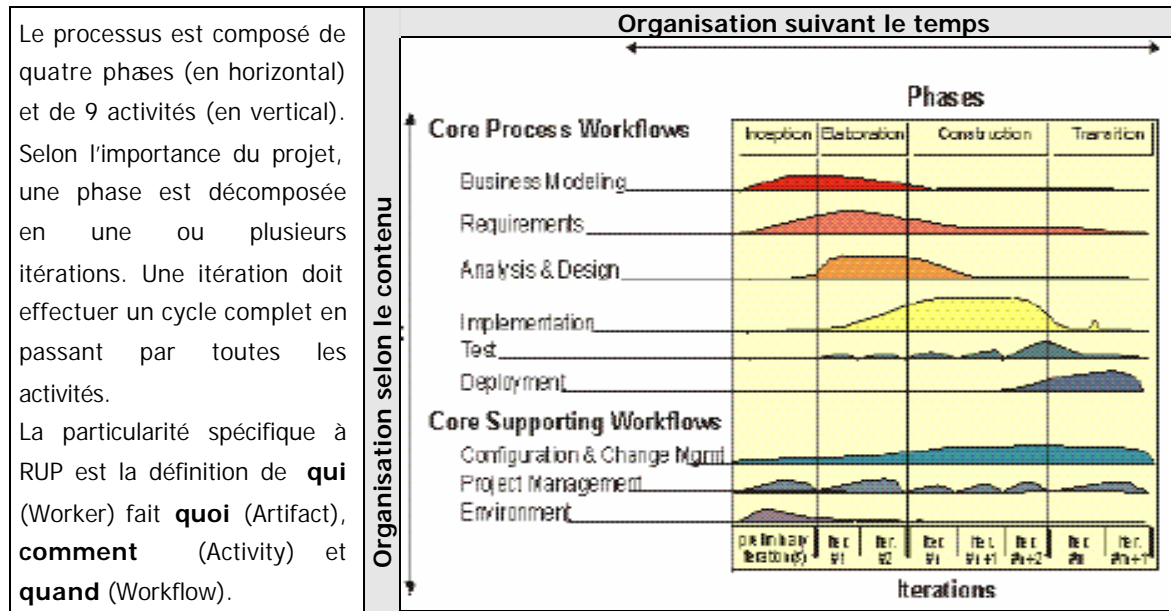
RUP adhère aux six meilleures pratiques de développement énoncées par Grady Booch ainsi qu'à d'autres pratiques qui sont de son propre cru. En voici un résumé succinct non limitatif :

- ⊙ La gestion des exigences est conduite selon un processus systématique en passant par l'explicitation des besoins, leur organisation et la gestion des futurs changements.
- ⊙ La pratique du développement itératif et incrémental, ce qui consiste à découper le projet en petites parcelles logiquement congrues et de progresser ainsi à petits pas comme s'il s'agissait d'une suite de mini projets.
- ⊙ On y accorde une grande importance à l'architecture logicielle et à la réutilisation de composants.
- ⊙ Le processus RUP est basé sur l'utilisation du standard UML pour modéliser le produit sous ses différentes vues, Etc.

3.4 Structure du processus RUP


Le processus RUP est organisé selon deux structures. La structure statique qui définit les éléments figés, tels que les différents travailleurs, les activités à réaliser, les artefacts à utiliser, ... La structure dynamique quant à elle, permet de mettre le processus en 'mouvement' dans le temps et ce, par l'entremise de la notion de phase et d'itération qui seront décrites plus loin dans ce chapitre.


3.4.1 La structure statique

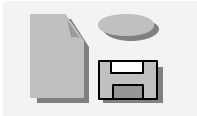


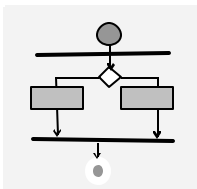
Ainsi, lors de la planification d'une itération, il est impératif de définir clairement ces 4 éléments (Worker, Artifact, Activity et Workflow) afin d'en assurer une exécution parfaite. En voici une description des quatre éléments fondamentaux caractérisant la structure statique du processus :

- ⊙ **« Worker »** : Un travailleur (worker) est une personne qui œuvre pour mener à bien les tâches qui lui sont assignées. Celui-ci peut assurer plusieurs rôles. Un architecte, un analyste système, un concepteur sont des exemples de travailleurs.


- ⊙ **« Activity »** : Toutes les tâches réalisées par un travailleur pour le compte d'un projet sont des activités. Chaque activité est réalisée selon une procédure bien définie qui se divise en trois phases : Une phase de réflexion, une phase de prise d'action et finalement la vérification du résultat obtenu.


- ⊙ **« Artifact »** : Tout produit tangible (document, logiciel, ...) utilisé lors de l'exécution d'une activité du processus est dénommé « artefact ». Celui-ci peut être utilisé en entrée d'une activité ou résulter en sortie de celle-ci.


- ⊙ **« Workflow »** : Un workflow est une description procédurale des opérations subséquentes afin de réaliser une activité en mettant en évidence l'interaction entre les utilisateurs du système. Chacune des activités du processus, de la modélisation d'affaire à l'étude de l'environnement, a son propre workflow, ce qui permet alors d'assister le travailleur dans la réalisation de ses activités.



Les « workflows » RUP permettent de décrire clairement la séquence d'activités à réaliser selon le type de discipline traitée, de préciser les travailleurs participants et les rôles qu'ils occupent ainsi que les artefacts mis en œuvre. Ils sont décrits au moyen de diagrammes d'activités au sens UML. L'exécution d'un workflow commence toujours par un point de départ et se termine par un ou plusieurs points de terminaison. Les activités correspondantes peuvent s'exécuter de façon séquentielle ou concurrente.

En plus des quatre éléments principaux déjà cités, d'autres notions complémentaires permettent de donner plus de précision au processus, en voici quelques-unes :

- ⊙ Les « Templates » : ce sont des modèles ou gabarits d'artefacts pré-établis permettant de faciliter la tâche aux travailleurs.
- ⊙ Les « ToolMentors » : Ce sont des directives complètes permettant de décrire comment utiliser un outil spécifique qui est externe au processus tel que Rational Rose ou requisite Pro.
- ⊙ Les « Guidelines » : ils permettent de fournir des informations complémentaires concernant toutes les notions traitées dans le processus, en particulier pour les différentes disciplines.

3.4.2 La structure dynamique

⊙ **Notion d'itération :**

La notion d'itération constitue la clé de voûte qui est principalement responsable du dynamisme du processus RUP. En effet, un projet est logiquement scindé en plusieurs parties dites « itérations » qui sont planifiées et exécutées individuellement comme s'il s'agissait de mini projets. L'exécution de chaque itération nécessite de passer en revue toutes les phases du cycle de développement, à savoir la modélisation d'affaire, la gestion des exigences, l'analyse et le design, le codage, les tests unitaires et les tests finaux d'intégration.

⊙ **Notion de phase :**

Le processus RUP est décomposé en quatre phases qui sont l'inception (ou la création), l'élaboration, la construction et la transition. Une phase n'est rien d'autre qu'une macro itération qui, elle-même est décomposée en un certain nombre d'itérations. Une phase est réalisée en exécutant une ou plusieurs itérations. Le nombre d'itérations d'une phase est défini en fonction du type de phase considérée et de l'importance du projet. En voici une description sommaire de chaque phase :

- **La phase d'inception ou de création**

Cette phase est la première parmi les quatre, elle consiste à expliciter et à bien comprendre les besoins de l'utilisateur du produit final. Pour se faire, on définit les éléments suivants : les exigences principales formulées en terme de cas d'utilisations, un plan de projet, une architecture candidate, les risques majeurs, etc.... Le résultat est consigné dans un document nommé « Vision document ».

- **La phase d'élaboration**

Elle consiste à analyser le problème, à produire un modèle de cas d'utilisation complet à 80%, à lever les risques majeurs ainsi que la validation d'une architecture de référence. Cette phase a comme finalité la stabilisation des éléments de base définis dans le « Vision document » ébauché dans la phase précédente.

- **La phase de construction**

C'est une expérimentation des résultats élaborés en phase précédente par une mise sur pieds d'une première solution opérationnelle conforme aux exigences des utilisateurs. C'est la phase la plus lente vu qu'elle consiste principalement à gérer les ressources et surtout à écrire les programmes qui implémentent la solution choisie.

- **La phase de transition**

Dans cette dernière phase, il s'agit de s'assurer que le produit répond bien au niveau de qualité exprimé par les utilisateurs du produit et que toutes les conditions sont réunies pour faire son déploiement.

Pour s'assurer de l'avancement d'un projet dans de bonnes conditions, une notion de point de contrôle (milestone) est mise à contribution (Voir figure 3.2 suivante). Un point de contrôle se localise soit à chaque fin d'itération, auquel cas il est dit majeur, soit à chaque fin de phase, auquel cas il est dit mineur. Un point de contrôle permet de vérifier que les résultats de la phase ou de l'itération précédente

sont conformes aux prévisions et que ces résultats rencontrent bien les exigences des entrées pour la phase ou de l'itération suivante, comme cela est schématisé sur un exemple fictif dans la figure qui suit.

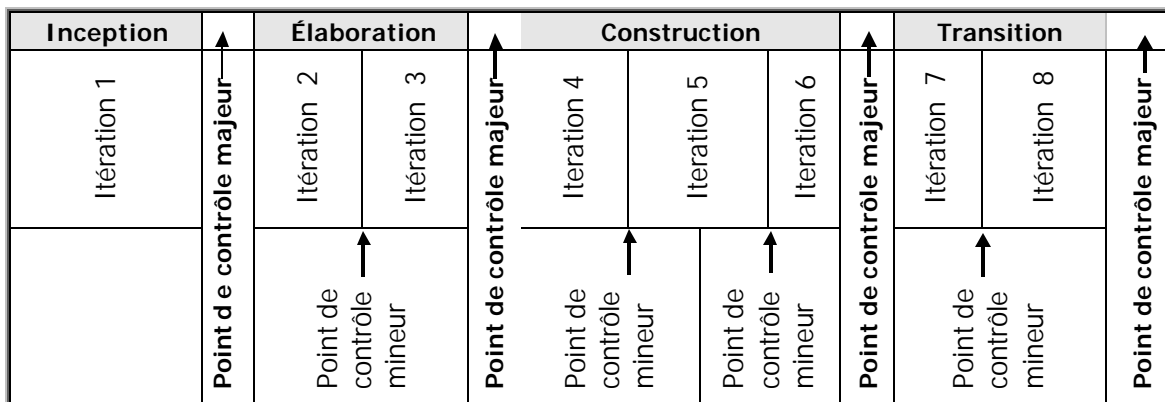


Figure 3.2 : Exemple de définition de points de contrôle

3.5 L'activité de mesure

3.5.1 Introduction

Le processus RUP définit neuf disciplines, dont six de type ingénierie, soient la modélisation d'affaire, la spécification des exigences, l'analyse et le design, l'implémentation, les tests et le déploiement ; et les trois autres sont de type support, à savoir la gestion de projet, la gestion de configuration et de changement et finalement l'étude de l'environnement.

À chaque discipline est associé un diagramme (workflow) décrivant les différentes activités à réaliser ainsi que leur séquençement. Ci-après, il sera question de définir la nouvelle activité de mesure, l'accent sera mis sur toutes les différentes alternatives où il sera intéressant de faire appel à l'outil de mesure externe de la taille fonctionnelle.

3.5.2 Catégorisation des disciplines RUP en rapport avec l'activité de mesure

La connaissance de la taille fonctionnelle est très utile particulièrement lors de la réalisation des disciplines de gestion de projet, de la gestion de configuration et de changement. En effet, ces deux disciplines font énormément usage de la notion de mesure, à cet égard, la taille fonctionnelle se trouve être une mesure primitive indispensable en tant qu'élément de base pour toute sorte d'estimation (effort, progrès, qualité, ...) Ci-après, quelques cas où la taille fonctionnelle s'avère nécessaire.

Après une étude approfondie des workflows des 9 disciplines du processus RUP, il en résulte deux catégories de disciplines (voir Figure 3.3 suivante) :

- Celles qui génèrent des informations en entrée de l'outil externe de calcul de la taille fonctionnelle. Cette catégorie inclut les disciplines suivantes : la modélisation d'affaire, la spécification des exigences et l'analyse et le design.
- Celles qui utilisent les résultats fournis par l'outil externe de calcul de la taille fonctionnelle. Cette catégorie inclut les disciplines suivantes : la gestion de projet et la gestion de configuration et de changement et, avec une moindre mesure, l'implantation, les tests et le déploiement.

En résumé, l'outil exploitera les 3 premières phases du cycle de développement (localisées à droite de la figure 3.3) pour faire l'arrimage du modèle générique COSMIC-FFP, calculer la taille fonctionnelle du projet et mettre à jour la base de données contenant l'historique des mesures effectuées sur divers projets.

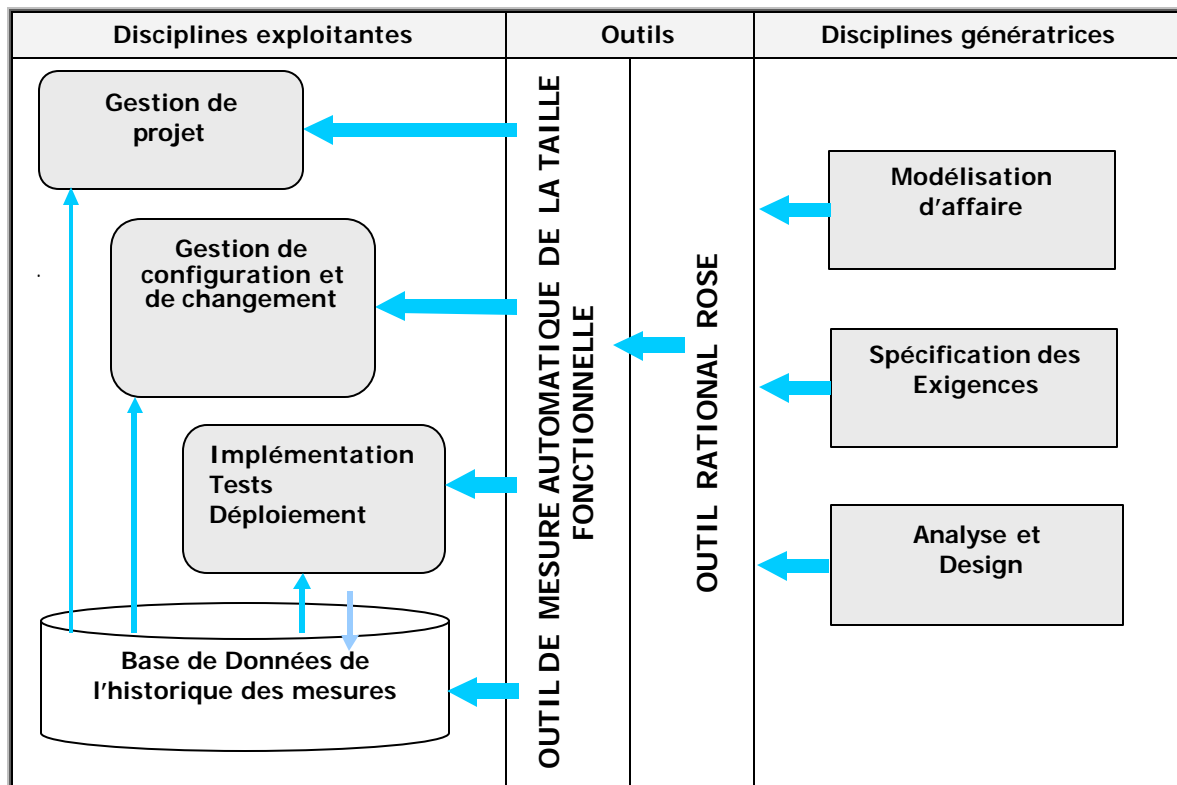


Figure 3.3 : Catégorisation des disciplines RUP

Par ailleurs, pour assister les travailleurs responsables des disciplines dites exploitantes (localisées à gauche de la figure 3.3), il est offert deux alternatives : soit se limiter aux résultats des calculs réalisés sur le projet en cours, soit consulter la base de données de l'historique des mesures pour avoir une vue globale de la tendance que suivent les projets de même nature.

3.5.3 Les disciplines exploitantes de la mesure

⊙ La gestion de projet

Le gestionnaire de projet peut faire appel à l'outil de calcul automatique de la taille fonctionnelle lors de la réalisation des activités suivantes :

- L'évaluation des risques :

Comme la taille du logiciel est un des facteurs de risque, la connaissance de cette taille permet d'avoir une meilleure idée du risque encouru.

- Le développement du plan de projet :

Comme on doit s'y attendre, une opération de planification nécessite d'avoir une bonne idée sur la taille du logiciel en question. Ce qui est particulièrement le cas lors de la définition du plan des mesures, de la définition des effectifs (effort) et surtout lors de la planification des phases et des itérations du projet.

- Le monitoring et le contrôle du projet :

L'opération de monitoring nécessite de mesurer l'effort et le progrès réalisés; par ailleurs, le contrôle du projet nécessite de faire des opérations de cédule des tâches ce qui dépend étroitement de la taille du logiciel.

◎ **La gestion de configuration et de changement**

C'est la deuxième discipline du processus RUP où il est possible de tirer profit de la connaissance de la taille fonctionnelle du logiciel. Cela est particulièrement intéressant lors de la planification des activités subséquentes aux demandes de changement. En effet, il est nécessaire d'avoir une grandeur approximative de la taille de la demande de changement afin de pouvoir estimer l'effort requis pour l'accomplir dans les délais impartis.

◎ **L'implantation**

Lors de la phase d'implantation, la connaissance de la taille fonctionnelle peut servir pour structurer le modèle correspondant en terme de la consistance des sous-systèmes à implémenter. Par ailleurs, la taille en lignes de code (LOC) résultant de cette phase servira pour mettre à jour la base de données de l'historique des mesures, ce qui en fera un modèle complet des résultats en Ufsu, Sfsu, Cfsu et LOC.

◎ **Tests / Déploiement**

Lors des phases de tests et de déploiement, la taille fonctionnelle permettra d'estimer l'importance de la fonctionnalité à tester ou à déployer, ce qui permet de prévoir l'effort à consentir.

3.6 Conclusion

Le processus RUP fait beaucoup usage de la notion de mesure sous toutes ses formes et ce dans presque la totalité de ses disciplines. À cet effet, la taille fonctionnelle se trouve être une mesure de base indispensable à la réalisation d'un programme de mesure dans ce processus. Concrètement, un outil externe de calcul de la taille fonctionnelle permettra d'assister le mesureur pour accomplir l'activité de mesure qui, par ailleurs peut être prise en charge par le gestionnaire de projet.

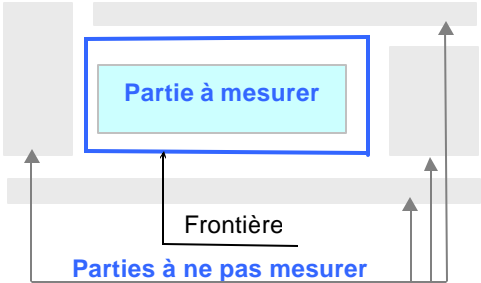
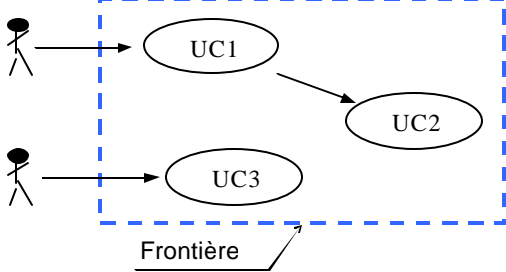
4.1 Introduction


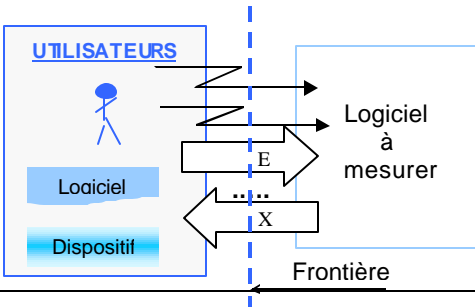
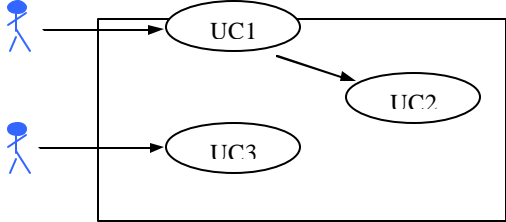
Ce document rentre dans le cadre du projet de calcul de la taille fonctionnelle de logiciels développés selon le processus RUP. Il permet d'éclaircir deux aspects importants sur lesquels sera bâti l'outil de mesure automatique de la taille fonctionnelle. Le premier aspect a trait au rapprochement des concepts du modèle générique COSMIC-FFP et de leur équivalent éventuel UML. Le deuxième aspect, quant lui, consiste à déterminer les différents types de mesures que l'on peut réaliser tout au long du cycle de développement d'un logiciel.

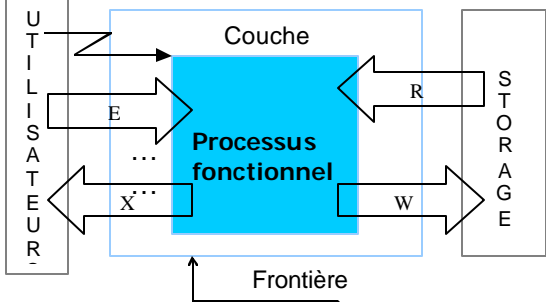
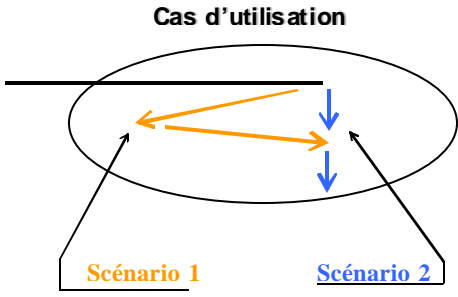
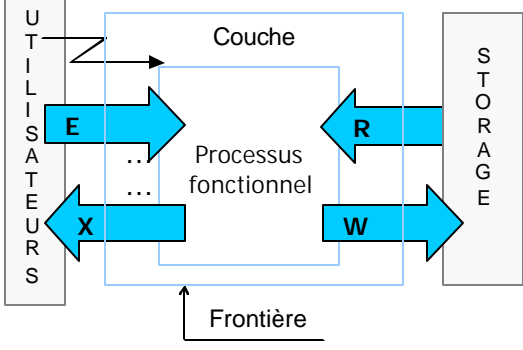
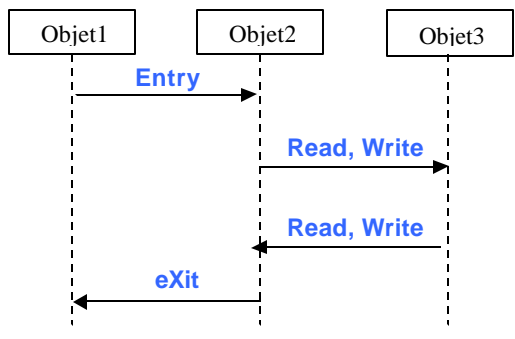
4.2 Rapprochement des concepts COSMIC-FFP de la notation UML

L'intérêt principal de ce rapprochement est de définir les bases conceptuelles sur lesquelles sera bâti l'outil de mesure automatique de la taille fonctionnelle, c'est dire l'importance particulière que revêt cette phase.

Ce rapprochement s'inspire de l'étude citée en référence 6, il sera réalisé entre la version 2.2 de la méthode COSMIC-FFP ISO 19761 de Janvier 2003 et la notation UML version 2.0. Il sera passé en revue tous les concepts COSMIC-FFP; pour chacun, son équivalent éventuel UML sera explicité. Il s'ensuivra une discussion et finalement on conclura sur l'utilisabilité du concept dans le cadre de ce projet.

Concept COSMICFFP Frontière du logiciel	Équivalent UML Diagramme de cas d'utilisation
<p>La frontière d'un logiciel (Software Boundary) est la ligne conceptuelle séparant ce logiciel de l'environnement dans lequel il opère, tel que perçu par ses utilisateurs d'un point de vue externe. La frontière permet à la personne qui mesure de distinguer, sans ambiguïté, ce qui est inclus dans le logiciel de ce qui fait partie de l'environnement dans lequel il fonctionne.</p> 	<p>Un diagramme de cas d'utilisation est un ensemble de cas d'utilisation complémentaires qui concourent pour réaliser les fonctionnalités du système en question. Pour ce faire, ces cas d'utilisations interagissent avec les acteurs du système, aussi, ils forment des associations entre-eux telles que l'inclusion, l'extension et des généralisations / spécialisation.</p> 
<p>Discussion La connaissance de la liste de tous les cas d'utilisations (soit le diagramme des cas d'utilisations) du logiciel à mesurer constitue la limite conceptuelle du système, soit donc sa frontière.</p> <p>Conclusion</p> <ul style="list-style-type: none"> ⊙ Il apparaît donc de ce qui précède que le diagramme de cas d'utilisations peut servir pour définir de manière non ambiguë et complète, la frontière du système en question. ⊙ Dans le cadre de ce projet, la recherche des cas d'utilisations appartenant à un diagramme de cas d'utilisation ne se fera pas automatiquement par programme: c'est à la personne qui s'occupe de l'activité de mesure au sein du processus RUP de délimiter les cas d'utilisations correspondants, ce qui revient en réalité à circonscrire la frontière du logiciel à mesurer. 	

<p align="center"><u>Concept COSMIC-FFP</u></p> <p align="center">Couches du logiciel</p>	<p align="center"><u>Équivalent UML</u></p> <p align="center">Aucun</p>
<p>Une couche (Layer) est le résultat du partitionnement fonctionnel de l'environnement du logiciel où tous les processus fonctionnels s'exécutent au même niveau d'abstraction.</p> 	<p>Il n'existe pas d'équivalent UML direct pour la notion de couches COSMIC-FFP.</p>
<p>Discussion</p> <p>La notion de couches COSMIC-FFP peut se résumer à un ensemble de systèmes conceptuellement indépendants et concourant pour réaliser un objectif commun et global. Dans le cas du processus RUP, il peut être question d'en faire un rapprochement avec le 'Layer Pattern' sous-jacent.</p>	
<p>Conclusion</p> <ul style="list-style-type: none"> ⊙ Dans le cas où le système est constitué de plusieurs couches, chacune d'elle sera considérée comme un système à part entière. Ainsi, la mesure de la taille du système global sera constituée du cumul des tailles de tous les systèmes pris individuellement. ⊙ Dans le cadre de ce projet, de la même façon que pour la frontière du logiciel, le dénombrement des couches du logiciel à mesurer ne se fera pas automatiquement par programme; c'est à la charge de la personne qui s'occupe de l'activité de mesure au sein du processus RUP de définir les couches et leurs frontières. 	
<p align="center"><u>Concept COSMIC-FFP</u></p> <p align="center">Utilisateur COSMIC-FFP</p>	<p align="center"><u>Équivalent UML</u></p> <p align="center">Acteur UML</p>
<p>Un utilisateur (User) du système peut être un être humain, tout autre logiciel ou un dispositif quelconque qui interagit avec le logiciel mesuré.</p> 	<p>Il s'agit d'un stéréotype prédéfini décrivant une entité hors du système, qui interagit avec ce dernier par l'intermédiaire des cas d'utilisation.</p> 
<p>Discussion</p> <p>On voit bien que ces deux notions d'utilisateur COSMIC-FFP et d'acteur UML sont les mêmes. Elles se réfèrent toutes les deux aux utilisateurs humains, logiciel ou tout autre dispositif qui est en interaction avec le système.</p>	
<p>Conclusion</p> <ul style="list-style-type: none"> ⊙ On peut conclure donc que la notion d'utilisateur FFP est assimilable à la notion d'acteur UML. ⊙ Dans le cadre de ce projet, le concept d'utilisateur COSMIC-FFP sera considéré exactement pareil que la d'acteur de la notation UML. 	

<p align="center"><u>Concept COSMIC-FFP</u></p> <p align="center">Processus fonctionnel</p>	<p align="center"><u>Équivalent UML</u></p> <p align="center">Cas d'utilisation</p>
<p>Un processus fonctionnel (Functional Process) est un ensemble unique et ordonné de mouvements de données (entrée, sortie, lecture et écriture) implantant un ensemble cohésif de requis fonctionnels. Il est déclenché directement ou indirectement via un « acteur » par un déclencheur. Il est complet lorsqu'il a réalisé tout ce qui doit être exécuté en réponse à un événement déclencheur.</p>  <p>The diagram shows a central blue box labeled 'Processus fonctionnel' within a larger box labeled 'Couche'. To the left is a vertical box 'UTILISATEUR' with arrows 'E' and 'X' pointing towards the process. To the right is a vertical box 'STORAGE' with arrows 'R' and 'W' pointing towards the process. A 'Frontière' line is at the bottom.</p>	<p>Il s'agit d'une séquence d'actions effectuées par un système, qui produit pour un acteur donné un résultat de valeur observable. (généralement, les scénarios illustrent les cas d'utilisation)</p>  <p>The diagram shows an oval representing a 'Cas d'utilisation'. Two arrows labeled 'Scénario 1' and 'Scénario 2' point to the use case. Inside the use case, there are orange arrows forming a loop and blue arrows pointing downwards.</p>
<p>Discussion</p> <p>De ce qui précède, il apparaît qu'un processus fonctionnel COSMIC-FFP se rapproche d'un cas d'utilisation UML. En effet, qu'il s'agisse d'un cas d'utilisation UML ou d'un processus fonctionnel COSMIC-FFP, tous les deux représentent une suite d'actions (considérées comme mouvements de données dans COSMIC-FFP) effectuées par le système, et dont la finalité est la satisfaction d'un des requis fonctionnels des utilisateurs (considérés comme acteurs dans UML) du système.</p>	
<p>Conclusion</p> <ul style="list-style-type: none"> ⊙ Ces deux notions ont toutes les deux comme finalité manifeste de satisfaire un requis fonctionnel exprimé par l'utilisateur. ⊙ Dans le cadre de ce projet, on peut donc conclure que le rapprochement entre un processus fonctionnel COSMIC-FFP et d'un cas d'utilisation UML est à envisager sans risque d'erreur. 	
<p align="center"><u>Concept COSMIC-FFP</u></p> <p align="center">Mouvement de données</p>	<p align="center"><u>Équivalent UML</u></p> <p align="center">Scénario</p>
 <p>The diagram is similar to the FFP diagram but with thicker blue arrows. 'E' and 'X' are on the left, 'R' and 'W' are on the right, and 'Frontière' is at the bottom.</p> <p>Un mouvement de données COSMIC-FFP (Data movement) est un déplacement de données élémentaires pendant l'exécution d'un processus fonctionnel. Il y a quatre classes de mouvements de données :</p>	 <p>The diagram shows three vertical dashed lines for 'Objet1', 'Objet2', and 'Objet3'. Messages are: 'Entry' from Obj1 to Obj2; 'Read, Write' from Obj2 to Obj3; 'Read, Write' from Obj3 to Obj2; and 'eXit' from Obj2 to Obj1.</p> <p>Il s'agit d'une séquence spécifique d'actions illustrant des comportements (effets observables d'une opération ou d'un événement). Un scénario peut être utilisé pour illustrer une interaction (spécification comportementale comprenant un ensemble de messages échangés entre des objets, dans un contexte particulier pour atteindre un but spécifique).</p>

Entrée (Entry) : Une entrée déplace les valeurs d'attributs d'un groupe de données de la partie usager de la frontière du logiciel vers l'intérieur de la frontière du logiciel. *(une entrée ne met pas à jour la donnée qu'elle déplace).*

Sortie (Exit) : Une sortie déplace les valeurs d'attributs d'un groupe de données de l'intérieur de la frontière du logiciel vers la partie usager de la frontière du logiciel. *(une sortie ne lit pas la donnée qu'elle déplace.)*

Lecture (Read) : Une lecture fait référence aux attributs de données d'un groupe de données sans en modifier les valeurs. Sur le plan fonctionnel, une lecture prend des données se trouvant à l'extérieur de la frontière du logiciel, dans la partie stockage, et les met à la disposition du processus fonctionnel auquel elle appartient.

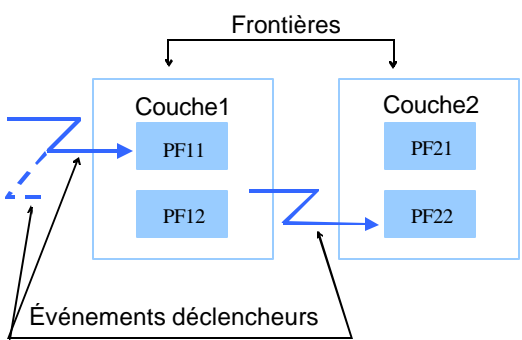
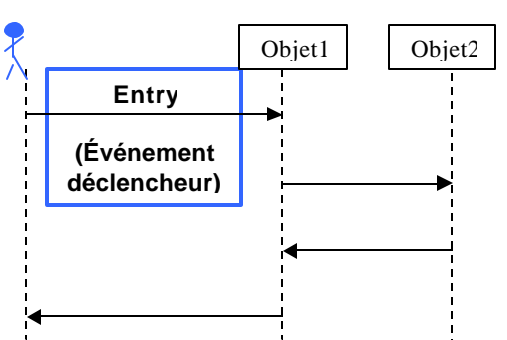
Écriture (Write) : Une écriture fait référence aux attributs de données d'un groupe de données et en modifie les valeurs. Sur le plan fonctionnel, une écriture envoie des données manipulées par le processus fonctionnel auquel elle appartient vers l'extérieur de la frontière du logiciel, dans la partie stockage.

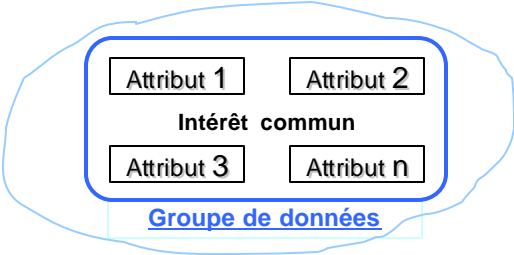
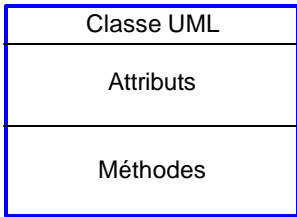
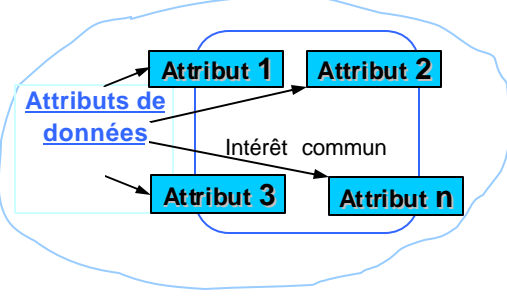
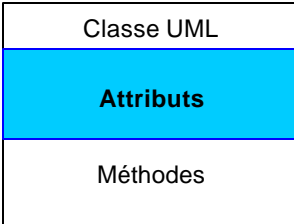
Discussion

On voit bien que les quatre types de mouvements de données (entrées, sorties, lectures ou écritures COSMIC-FFP) ne sont rien d'autre que les opérations réalisées au niveau d'un diagramme de séquences (ou scénario UML). Il est à remarquer que les mouvements de données de type Entry et eXit sont identifiables puisqu'ils opèrent sur un objet stéréotypé comme acteur du système, ce qui n'est cependant pas le cas des deux autres types de mouvements de données (Read et Write).

Conclusion

- De ce qui précède, il apparaît qu'un scénario UML se rapproche d'une séquence de mouvements de données COSMIC-FFP. En effet, les actions mentionnées dans un scénario UML peuvent être exprimées en terme de mouvements élémentaires de données (entrées, sorties, lectures ou écritures FFP) et donc de mouvements de données COSMIC-FFP.
- Dans le cadre de ce projet, les mouvements de données de type Entry et eXit seront bien identifiés en type et en nombre, par contre, ceux de type Read et Write seront identifiés seulement en nombre.

<p align="center"><u>Concept COSMIC-FFP</u> Événement déclencheur</p>	<p align="center"><u>Équivalent UML</u> Aucun</p>
<p>Un événement déclencheur (Triggering event) se produit à l'extérieur de la frontière du logiciel à mesurer et qui initie un ou plusieurs processus fonctionnel(s). Lorsque chaque couche identifiée est entourée par une frontière, des événements déclencheurs peuvent se produire dans une couche et initié un processus fonctionnel dans une autre couche.</p>  <p>Le diagramme illustre deux couches fonctionnelles, Couche1 et Couche2, chacune entourée d'une frontière. Couche1 contient les processus fonctionnels PF11 et PF12, tandis que Couche2 contient PF21 et PF22. Des flèches bleues indiquent des événements déclencheurs entrants et sortants à travers les frontières. Des flèches noires à l'intérieur des couches montrent des interactions entre les processus fonctionnels.</p>	<p>Il n'existe pas d'équivalent UML pour la notion d'événement déclencheur de COSMIC-FFP.</p>  <p>Le diagramme de séquence UML montre un acteur (stick figure) qui déclenche un message 'Entry (Événement déclencheur)' vers un objet 'Objet1'. Ensuite, il y a des messages de communication entre 'Objet1' et 'Objet2'.</p>

<p>Discussion</p> <p>Nous ferons le rapprochement entre un déclencheur FFP et une interaction acteur/système UML pour laquelle il n'y a pas d'échange de données (celles qui apparaissent sur le diagramme de séquence) entre l'acteur et le système (dans le sens acteur vers le système).</p>	
<p>Conclusion</p> <p>En plus de constituer une unité de taille fonctionnelle à part entière, un mouvement de données de type Entry sera considéré comme déclencheur qui va initier le (ou les) processus fonctionnel(s) mis en cause.</p>	
<p><u>Concept COSMIC-FFP</u></p> <p>Groupe de données</p>	<p><u>Équivalent UML</u></p> <p>Classe UML</p>
<p>Un groupe de données (Data group) est un ensemble non ordonné, non vide et non redondant d'attributs de données, où chaque attribut décrit un aspect complémentaire du même objet d'intérêt.</p> 	<p>Une classe est une description d'un ensemble d'objets de même nature partageant les mêmes attributs, méthodes, relations et sémantiques.</p> 
<p>Discussion</p> <p>Si nous nous limitons uniquement aux attributs (sans considérer les méthodes, ni les relations), il apparaît qu'une classe UML devient tout simplement un ensemble (non ordonné) non vide et non redondant d'attributs de données, et donc un groupe de données FFP.</p>	
<p>Conclusion</p> <ul style="list-style-type: none"> ⊙ La notion conceptuelle COSMIC-FFP d'attributs d'un groupe de données est exactement la même que les attributs d'une classe UML. ⊙ Dans le cadre de ce projet, un attribut d'un groupe de données COSMIC-FFP sera considéré comme un attribut d'une classe UML. 	
<p><u>Concept COSMIC-FFP</u></p> <p>Attribut de données</p>	<p><u>Équivalent UML</u></p> <p>Attribut d'un objet</p>
<p>Un attribut de données (Data attribute) est la plus petite parcelle d'information codée et significative du point de vue des réquis fonctionnels des utilisateurs.</p> 	<p>Il s'agit d'une propriété nommée d'un objet.</p> 

Discussion

Il apparaît qu'un attribut de données COSMIC-FFP se rapproche bien d'un attribut d'objet UML. En effet, les attributs d'objets en UML ne sont que de l'information codée et significative du point de vue des requis fonctionnels des utilisateurs.

Conclusion

- ⊙ La notion conceptuelle COSMIC-FFP d'attribut d'un groupe de données est exactement la même que les attributs d'une classe UML.
- ⊙ Dans le cadre de ce projet, un attribut d'un groupe de données COSMIC-FFP sera considéré comme un attribut d'une classe UML.

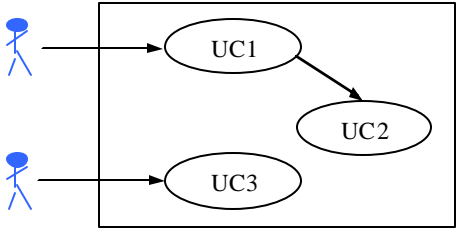
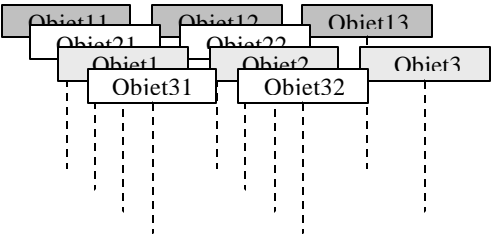
4.3 Les différents niveaux d'abstraction de mesure

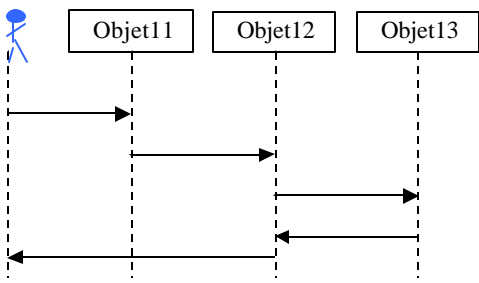
Cette partie consiste à définir les différentes mesures que l'on peut réaliser au cours du cycle de développement du logiciel. En effet, d'une phase à une autre, les artefacts permettant d'évaluer (ou de mesurer) la taille fonctionnelle sont différents, ce qui nécessite alors de les identifier et de définir les éléments conceptuels à mettre en œuvre pour cerner le problème.

Phases du cycle de développement	Artefact utilisé	Unités et proportions
<ul style="list-style-type: none"> • Modélisation d'affaire ou • Spécification des exigences 	<ul style="list-style-type: none"> • Diagramme de cas d'utilisation 	
<ul style="list-style-type: none"> • Analyse / Design 	<ul style="list-style-type: none"> • Scénarios 	
<ul style="list-style-type: none"> • Analyse / Design 	<ul style="list-style-type: none"> • Détail des scénarios 	

À cet effet, il a été identifié trois phases du cycle de développement où la mesure pourrait être réalisée :

- ⊙ Au niveau des phases de modélisation d'affaire et de spécification des exigences, où il sera fait usage des diagrammes de cas d'utilisation comme artefacts à partir desquels sera évaluée la taille fonctionnelle. L'unité de la taille à ce niveau est identifiée comme **Ufsu** – *Use Case functional size unit* (Voir détail au tableau suivant)
- ⊙ Au niveau de la phase d'analyse, où il sera fait usage des scénarios comme artefacts à partir desquels la taille fonctionnelle sera évaluée. L'unité de taille à ce niveau est identifiée comme **Sfsu** – *Scenario functional size unit* (Voir détail au tableau suivant)
- ⊙ Au niveau de la phase d'analyse (et des fois de Design), où il sera fait usage du détail des scénarios comme artefacts à partir desquels sera évaluée la taille fonctionnelle. L'unité de la taille à ce niveau est identifiée comme **Cfsu** – selon la norme ISO 19761 *COSMIC functional size unit* (Voir détail au tableau suivant)

Diagramme des cas d'utilisation	Indicateurs précurseurs de la taille
	<ul style="list-style-type: none"> ⦿ Le nombre de cas d'utilisation, ⦿ Le nombre d'acteurs, ⦿ Le nombre d'interactions dans le diagrammes des cas d'utilisation, <p>Dans l'exemple ci-contre, le nombre de cas d'utilisation est de 3, le nombre d'acteurs est de 2 et le nombre d'interactions est de 3, ce qui donne un vecteur (indicateur) précurseur de la taille fonctionnelle:</p> <p style="text-align: center;">3 Ufsu(2,3) ou encore (3 Ufsu, 2,3)</p>
<p>Explication</p> <p>Cela peut s'appliquer à deux niveaux du cycle de développement : la modélisation d'affaire et la spécification des exigences (requirements). La quantification de la taille fonctionnelle à ces niveaux portera sur l'ampleur du diagramme de cas d'utilisation. Les éléments déterminant pour quantifier la taille d'un diagramme de cas d'utilisation sont :</p> <ul style="list-style-type: none"> ⦿ Le nombre de cas d'utilisation, qui mesure l'étendue du système en question. ⦿ Le nombre d'acteurs, ce qui permet de déterminer la force d'interaction du système avec son environnement. ⦿ Le nombre d'interactions entre les cas d'utilisation, qui permet de quantifier les fonctionnalités offertes par le système. 	
<p>Unité utilisée</p> <p>La quantification d'un diagramme de cas d'utilisation aura comme unité le Ufsu (pour Use case Functional Size Unit) et sera exprimée comme suit :</p> <p style="text-align: center;">x : nombre de «use cases» Ufsu (y : nombre d'acteurs, z : nombre d'interactions)</p>	
<p>Intérêt</p> <p>Cette quantité (ou vecteur) est un indicateur précurseur de la taille fonctionnelle. Son utilité se révèle très importante quand on est en présence de plusieurs projets dont les tailles sont consignées dans une base de données. Dans ce cas, ces informations s'avèrent d'un grand intérêt pour définir les ratios caractéristiques de l'évolution de la taille fonctionnelle durant le cycle de développement selon le processus RUP.</p>	
Diagramme des scénarios	Indicateurs précurseurs de la taille
	<ul style="list-style-type: none"> ⦿ Le nombre de scénarios, ⦿ Le nombre d'objets participants, <p>Dans l'exemple ci-contre, le nombre de scénarios est de 4 et le nombre d'objets participants est de 10, ce qui donne comme évaluation de la taille fonctionnelle :</p> <p style="text-align: center;">(4,10) Sfsu</p>
<p>Explication</p> <p>Cela peut s'appliquer aux phases d'analyse et de design du cycle de développement. L'évaluation de la taille fonctionnelle à ce niveau se basera sur les scénarios des cas d'utilisation. Les éléments déterminants pour évaluer la taille des scénarios sont :</p> <ul style="list-style-type: none"> ⦿ Le nombre de scénarios, qui mesure l'étendue des cas d'utilisation du système en question. ⦿ Le nombre d'objets participants, ce qui permet de quantifier la portée des scénarios. 	
<p>Unité utilisée</p> <p>La quantification des scénarios des cas d'utilisation aura comme unité le Sfsu (pour des Scénarios Functional Size Unit) et sera exprimée sous forme d'un vecteur comme suit :</p> <p style="text-align: center;">(x : nombre de scénarios, y : nombre d'objets participants) Sfsu</p>	

<p>Intérêt</p> <p>De même que pour le diagramme des cas d'utilisation, cette quantité est un vecteur (indicateur) précurseur. Son utilité se révèle très importante quand on est en présence de plusieurs projets dont les tailles sont consignées dans une base de données. Dans ce cas, ces données s'avèrent d'un grand intérêt pour définir les ratios caractéristiques de l'évolution de la taille fonctionnelle durant le cycle de développement.</p>	
<p>Détail des scénarios</p> 	<p>Éléments de la mesure de la taille</p> <ul style="list-style-type: none"> ● Nombre de mouvements de données entre les objets. <p>Dans l'exemple ci-contre, le nombre de mouvements de données entre les objets est de 5, dont 1 entrée, 1 sortie et 3 lectures ou écritures dépendamment de la persistance de l'objet cible. Ce qui donne comme mesure de la taille fonctionnelle selon la norme ISO 19761 :</p> <p style="text-align: center;">5 Cfsu</p>
<p>Explication</p> <p>Cela peut s'appliquer aux phases d'analyse et design du cycle de développement. L'évaluation de la taille fonctionnelle à ce niveau se basera sur le détail des scénarios. L'élément déterminant pour mesurer la taille d'un scénario est :</p> <ul style="list-style-type: none"> ● Le nombre de mouvements de données entre les objets, qui mesure l'étendue du scénario. ● Le type (acteur, objet quelconque ou objet persistant) des objets participant dans une opération permet de définir le type du point de fonction (Entrée, Sortie, Lecture ou écriture). 	
<p>Unité utilisée</p> <p>La mesure du détail d'un scénario a comme unité le Cfsu (pour Cosmic Functional Size Unit, selon la norme ISO 19761) et sera exprimée comme suit :</p> <p style="text-align: center;">x : nombre de points de fonction Cfsu</p>	
<p>Intérêt</p> <p>Combiné avec les deux indicateurs précurseurs précédents de la taille fonctionnelle (Ufsu et Sfsu) et à partir d'une série de mesures consignées dans une base de données, la taille en Cfsu permettra d'identifier un ou des « ratios » qui quantifie(nt) selon l'unité à un niveau, puis estime les unités à un autre niveau, en passant d'une phase à une autre du cycle de développement selon le processus RUP.</p>	

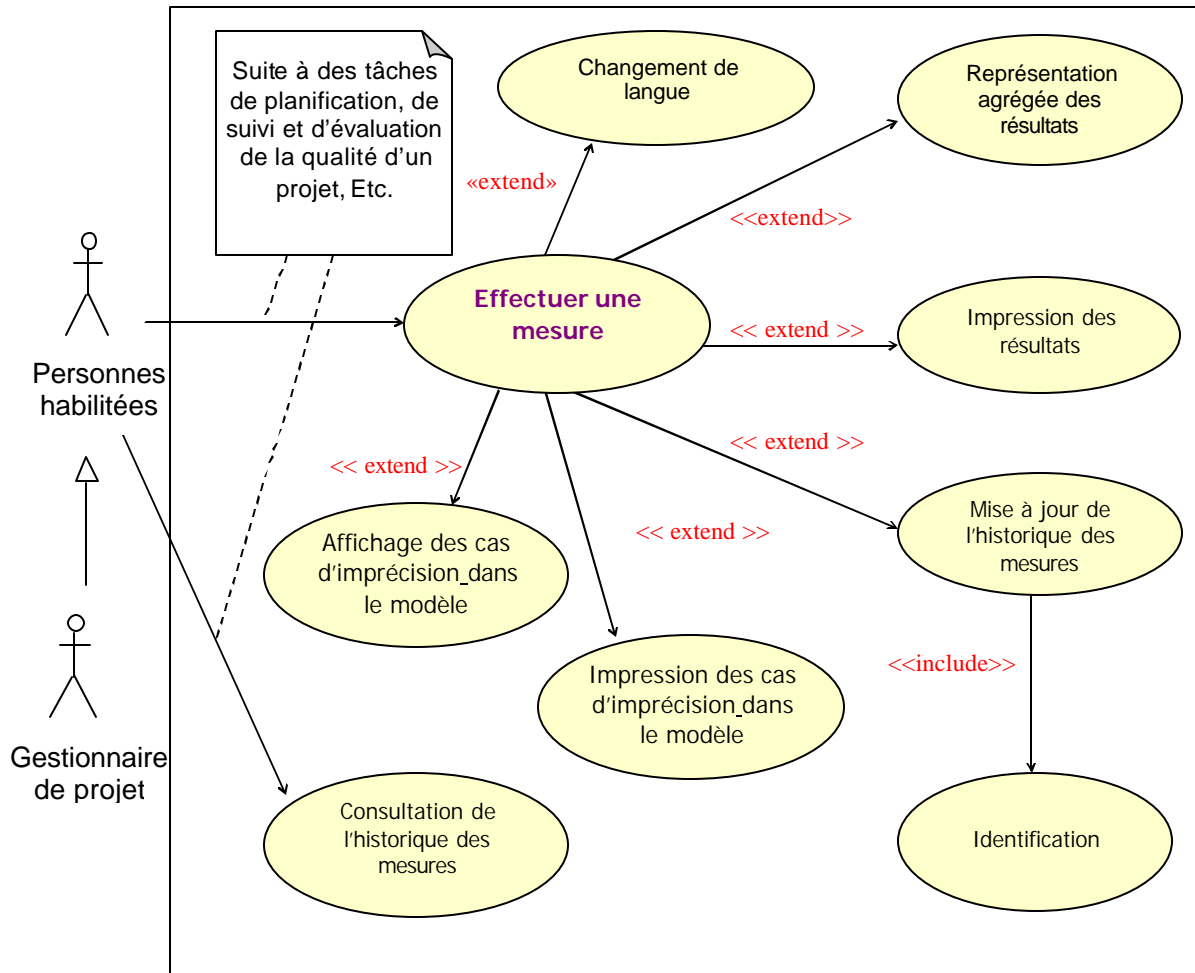
4.4 Vérifications

Les conclusions de cette étude sont prometteuses : l'objectif visé est de répondre à la question suivante : « Quels sont les facteurs d'échelle entre les différents types d'unités de mesure », soit idéalement une formule mathématique de la forme : $x \text{ Ufsu} = y \text{ Sfsu} = z \text{ Cfsu}$

Il est toutefois, trop tôt d'évaluer la précision et la portée des résultats tant que des études expérimentales sur des cas réels ne sont pas probantes. À cette fin, l'outil à développer fera l'objet de tests avec deux études de cas (Rice Cooker et Valve Control-14143-4 RUR) dont les tailles ont été préalablement calculées manuellement avec la méthode COSMIC-FFP par des experts du domaine. Une analyse comparative des résultats manuels et automatisés (avec l'outil) sera effectuée. Parmi les éléments à analyser on cite, entre autres :

- le degré de fiabilité de l'outil,
- les causes probables d'une divergence éventuelle des résultats,
- les améliorations possibles.

5.1 Les cas d'utilisation du système



5.2 Description des cas d'utilisation

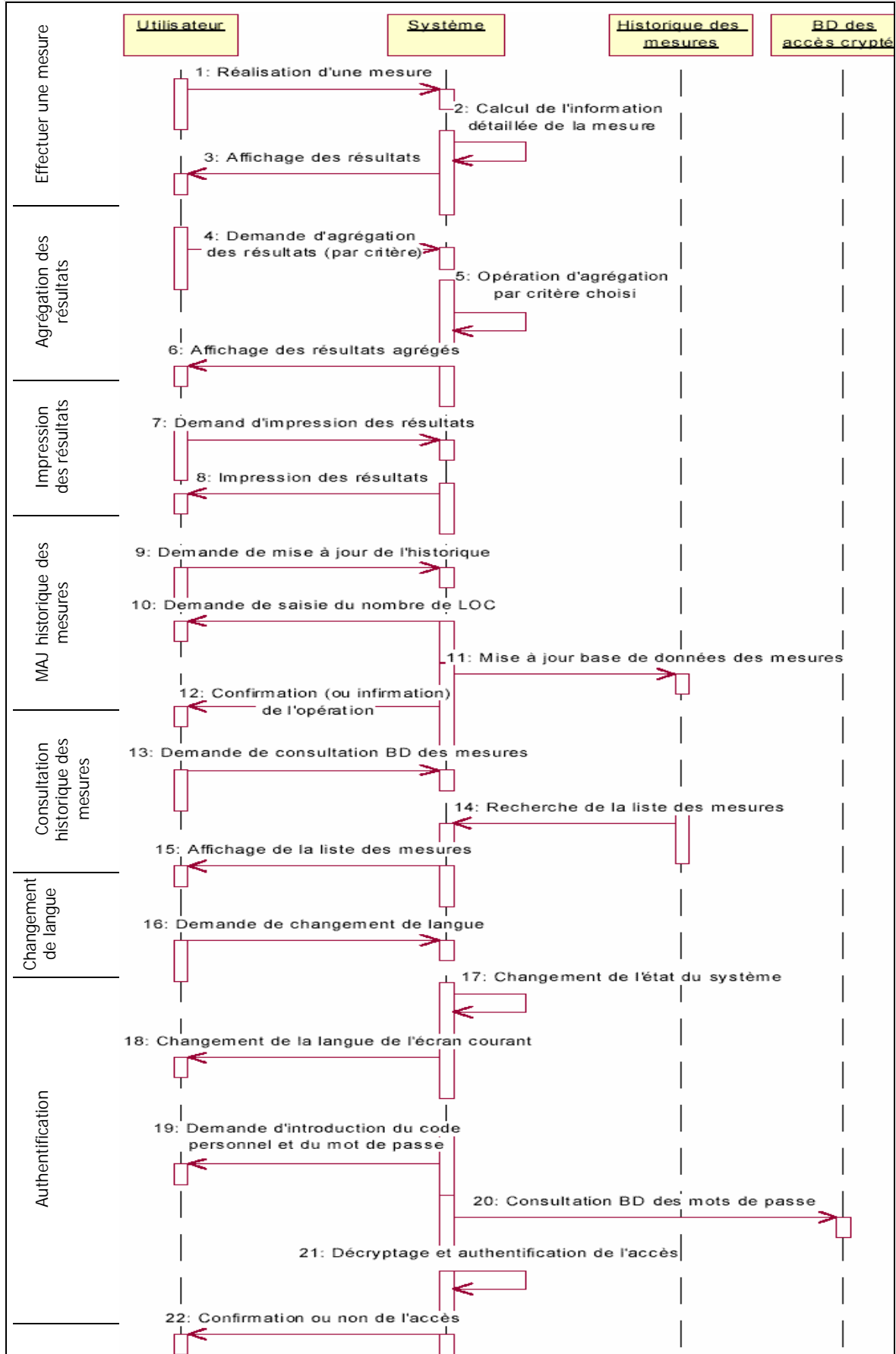
Titre du cas d'utilisation : Effectuer une mesure	Numéro : 1
Description sommaire : Ce cas d'utilisation permet de réaliser une mesure de la taille fonctionnelle d'un projet donné. Il doit fournir les résultats complets.	
Acteur primaire : Le gestionnaire de projet	
Acteur secondaire : Aucun	
Règles d'initiation : <ol style="list-style-type: none"> 1. L'outil Rational Rose doit être ouvert, 2. Le projet correspondant à mesurer doit être chargé dans Rational Rose, 3. Choix de l'option correspondante au calcul de la taille fonctionnelle. 	
Description du processus : Au besoin, et c'est le cas lors des opérations de planification, de suivi des progrès du projet ou de l'évaluation de la qualité ... , on procède au calcul de la taille fonctionnelle du logiciel. Cette dernière est la mesure de base indispensable à l'accomplissement de tels besoins. Pour se faire, le gestionnaire de projet doit préalablement s'identifier au système puis faire le choix du calcul de la taille fonctionnelle. Le résultat doit être délivré sous trois formes :	

<ol style="list-style-type: none"> 1. La mesure en Ufsu, qui évalue la taille fonctionnelle à un niveau d'abstraction très élevé (niveau des cas d'utilisation), 2. La mesure en Sfsu, qui évalue la taille fonctionnelle à un niveau d'abstraction élevé (niveau des scénarios), 3. La mesure en Cfsu, qui évalue la taille fonctionnelle à un niveau d'abstraction selon la norme ISO (niveau mouvement de données),
<p>Règles de terminaison :</p> <ol style="list-style-type: none"> 1. Choix de fermeture du logiciel par l'utilisateur, 2. Dans le cas d'un nouveau projet, le système doit signaler si l'historique des mesures n'a pas été mise à jour.
<p>Extensions :</p> <ol style="list-style-type: none"> 1. Représentation agrégée des résultats, 2. Impression des résultats 3. Mise à jour de l'historique des mesures.
<p>Inclusion : Aucun</p>
<p>Titre du cas d'utilisation : Représentation agrégée des résultats Numéro : 2</p>
<p>Description sommaire : Ce cas d'utilisation permet de représenter les résultats du calcul de la taille fonctionnelle d'une façon agrégée conformément au manuel de la méthode COSMIC-FFP.</p>
<p>Acteur primaire : Aucun</p>
<p>Acteur secondaire : Aucun</p>
<p>Règles d'initiation : Exécution du cas d'utilisation « Effectuer une mesure ».</p>
<p>Description du processus: Les résultats agrégés dont traite ce cas d'utilisation, sont définis conformément au manuel de mesure de la méthode COSMIC-FFP. Ils sont au nombre de cinq, que voici :</p> <ol style="list-style-type: none"> 1. Résultats agrégés par processus fonctionnel et par mouvement de données, 2. Résultats agrégés par proportion de chaque processus fonctionnel dans l'application, 3. Résultats agrégés par proportion des mouvements de données dans l'application, 4. Représentation graphique de la proportion par mouvement de données
<p>Règles de terminaison : Choix de fermeture de l'option par l'utilisateur.</p>
<p>Extensions : Aucune</p>
<p>Inclusion : Aucune</p>
<p>Titre du cas d'utilisation : Impression des résultats Numéro : 3</p>
<p>Description sommaire : Ce cas d'utilisation permet de d'imprimer les résultats détaillés du calcul de la taille fonctionnelle.</p>
<p>Acteur primaire : Aucun</p>
<p>Acteur secondaire : Aucun</p>
<p>Règles d'initiation : Exécution du cas d'utilisation « Effectuer une mesure ».</p>
<p>Description du processus: Une impression des résultats détaillés est réalisée au besoin. Elle contient tous les mouvements de données de chaque processus fonctionnel. Aussi, un cumul général et par processus fonctionnel de la taille sera fourni.</p>
<p>Règles de terminaison : Choix de fermeture de l'option d'impression par l'utilisateur,</p>
<p>Extensions : Aucune</p>
<p>Inclusion : Aucune</p>

Titre du cas d'utilisation : Mise à jour de l'historique des mesures	Numéro : 4
Description sommaire : Ce cas d'utilisation permet de mettre à jour l'historique des mesures.	
Acteur primaire : Le gestionnaire de projet	
Acteur secondaire : Aucun	
Règles d'initiation : <ol style="list-style-type: none"> 1. L'utilisateur doit s'identifier par son code personnel et son mot de passe, 2. L'utilisateur doit être une personne habilitée à réaliser l'opération de mise à jour. 	
Description du processus : Tout projet a une seule taille à un moment donnée. La mise à jour efface donc la taille précédente. La taille peut être exprimée sous quatre formes : <ol style="list-style-type: none"> 1. La mesure en Ufsu, 2. La mesure en Sfsu, 3. La mesure en Cfsu, 4. La taille en lignes de code (LOC). <p>Cette dernière n'est pas calculée par le logiciel, elle est mise à jour manuellement. Sa présence complète le modèle, et elle est indispensable pour de futures études de ratios (relations entre ces différents types de mesures).</p>	
Règles de terminaison : <ol style="list-style-type: none"> 1. Mise à jour de la base des mesures, 2. Choix de fermeture de l'option de mise à jour par l'utilisateur. 	
Extensions : Aucune	
Inclusion : Cas d'utilisation « Authentification »	
Titre du cas d'utilisation : Consultation de l'historique des mesures	Numéro : 5
Description sommaire : Ce cas d'utilisation permet de consulter l'historique des mesures.	
Acteur primaire : Le gestionnaire de projet	
Acteur secondaire : Aucun	
Règles d'initiation : <ol style="list-style-type: none"> 1. L'utilisateur doit s'identifier par son code personnel et son mot de passe, 2. L'utilisateur doit être une personne habilitée à réaliser l'opération de mise à jour. 	
Description du processus : Ce cas d'utilisation permet de consulter l'historique des mesures effectuées sur les anciens projets. Cette consultation n'existe qu'à titre informatif, elle n'a pas la prétention de fournir des déductions quelconques. Ce sera le but d'un autre projet qui utilisera ces données des mesures afin de faire une étude de ratios.	
Règles de terminaison : Choix de fermeture de l'option de consultation par l'utilisateur.	
Extensions : Aucune	
Inclusion : Aucun	
Titre du cas d'utilisation : Identification	Numéro : 6
Description sommaire : Ce cas d'utilisation permet d'identifier le personnel autorisé à exécuter la mise à jour de l'historique des mesures.	
Acteur primaire : Le gestionnaire de projet ou toute autre personne habilitée.	
Acteur secondaire : Aucun	
Règles d'initiation : <ol style="list-style-type: none"> 1. L'outil Rational Rose doit être ouvert, 2. Le projet correspondant à mesurer doit être chargé dans Rational Rose, 3. exécution de l'option de mise à jour de l'historique des mesures, 4. Introduction du code personnel de l'utilisateur et de son mot de passe. 	

Description du processus: À partir du code personnel de l'utilisateur et de son mot de passe, on procède à l'authenticité de l'accès au logiciel. Un message sera affiché si l'accès n'a pas abouti.	
Règles de terminaison : Afficher un message si l'accès est infructueux.	
Extensions : Aucune	
Inclusion : Aucune	
Titre du cas d'utilisation : Affichage des cas d'imprécision dans le modèle	Numéro : 7
Description sommaire : Ce cas d'utilisation permet d'afficher les cas d'imprécision dans le modèle conceptuel du logiciel à mesurer.	
Acteur primaire : Aucun	
Acteur secondaire : Aucun	
Règles d'initiation : Exécution du cas d'utilisation « Effectuer une mesure ».	
Description du processus: Lors de la réalisation d'une mesure, le modèle générique COSMIC-FFP doit déterminer le type du mouvement de données : Entrée (E), Sortie (X), Lecture(R) ou Écriture(W). Si l'utilisateur ne précise pas cette information pour le cas de Lecture(R) et Écriture(W), un message doit l'informer de cette imprécision et indiquer le cas d'utilisation et les deux objets mis en cause. Toutefois, le compte de la taille fonctionnelle n'est pas affecté.	
Règles de terminaison : Choix de fermeture de l'option d'affichage par l'utilisateur,	
Extensions : Aucune	
Inclusion : Aucune	
Titre du cas d'utilisation : Impression des cas d'imprécision dans le modèle	Numéro : 8
Description sommaire : Ce cas d'utilisation permet d'imprimer les cas d'imprécision dans le modèle conceptuel du logiciel à mesurer.	
Acteur primaire : Aucun	
Acteur secondaire : Aucun	
Règles d'initiation : Exécution du cas d'utilisation « Effectuer une mesure ».	
Description du processus: Un message est imprimé pour tout mouvement de données (Lecture et Écriture) dont le type est imprécis. Le message doit indiquer le cas d'utilisation et les deux objets mis en cause.	
Règles de terminaison : Choix de fermeture de l'option d'impression par l'utilisateur,	
Extensions : Aucune	
Inclusion : Aucune	
Titre du cas d'utilisation : Changement de langue	Numéro : 9
Description sommaire : Ce cas d'utilisation permet de procéder au changement de langue du logiciel pendant la session de travail. Par défaut, la langue est le français.	
Acteur primaire : Aucun	
Acteur secondaire : Aucun	
Règles d'initiation : Exécution du cas d'utilisation « Effectuer une mesure ».	
Description du processus: Suite à la demande de l'utilisateur, le basculement dans la langue choisie est effectué. Cela reste valable tant qu'un autre choix n'est pas effectué.	
Règles de terminaison : Aucune	
Extensions : Aucune	
Inclusion : Aucune	

5.3 Liste des scénarios



5.4 Description des scénarios

- **Effectuer une mesure :** Suite à des tâches de planification ou de suivi et de vérification de la qualité d'un projet, ... l'utilisateur de cet outil de calcul de la taille fonctionnelle initie le cas d'utilisation « Effectuer une mesure ». Concrètement, cela consiste à ouvrir le projet à mesurer dans Rational Rose et à exécuter le logiciel de calcul. Ce dernier effectue une opération de mesure.
- **Représentation agrégée des résultats :**
 Sur demande de l'utilisateur, les résultats représentés d'un point de vue agrégé selon :
 - Le nombre de mouvements de données de chaque type par processus fonctionnel,
 - La proportion de la taille de chaque processus fonctionnel dans l'application,
 - La proportion de la taille de chaque type de mouvement de données dans l'application (Entry, Exit, Read, Write),
 - La représentation graphique de la proportion de la taille de chaque type de mouvement de données dans l'application.
- **Impression des résultats :** Sur demande de l'utilisateur, une impression des résultats détaillés par processus fonctionnel et par mouvement de données est effectuée.
- **Mise à jour de l'historique des mesures :** Sur demande de l'utilisateur, une mise à jour de l'historique des mesures est effectuée. Le système demande la saisie du type de projet et de la taille en lignes de code pour compléter le modèle de l'historique des mesures.
- **Consultation de l'historique des mesures :** Sur demande de l'utilisateur, un affichage de la liste de l'historique des mesures est effectué.
- **Identification :** Le système demande l'introduction du code de l'utilisateur et de son mot de passe pour authentifier l'accès.
- **Affichage / Impression des cas d'imprécision dans le modèle :** Sur demande de l'utilisateur, un affichage / impression des cas d'imprécision dans le modèle conceptuel du logiciel mesuré est effectué.
- **Changement de langue :** Sur demande de l'utilisateur, le contenu de l'écran en cours est traduit dans la langue demandée. L'état interne définissant la langue en cours est modifié en conséquence.

5.5 Résumé des traitements à effectuer

En résumé, voici la liste des traitements à programmer :

- Réalisation détaillée d'une mesure,
- Agréger les résultats de la mesure selon l'un des 4 critères précédents,
- Impression des résultats détaillés,
- Mise à jour de l'historique des mesures,
- Consultation de l'historique des mesures,
- Une authentification pour accéder au système,
- Un affichage des cas d'imprécision dans le modèle,
- Une impression des cas d'imprécision dans le modèle,
- Changement de langue.

5.6 Description de la base de données

La base de données est très simple, elle est constituée de deux tables indépendantes : la première pour stocker l'historique des mesures et la deuxième pour contenir les codes d'accès cryptés.

5.6.1 Table de l'historique des mesures

Niveau de détail	Identificateur	Type	Taille	Description
	nomProjet	Texte	30	Identification du projet ouvert dans Rational Rose en vue de le mesurer. Le nom n'inclut pas le chemin du projet.
	typeProjet	Texte	1	Identification du type de projet par un code (1: Application d'affaire, 2: Application temps réel, 3 : Application embarquée, 4 : Autre)
Ufsu (Niveau Use cas)	nbreUC	Numérique	4	Nombre de cas d'utilisation.
	nbreActeur	Numérique	4	Nombre d'acteurs intervenant dans le diagramme de cas d'utilisation.
	nbreInter	Numérique	4	Nombre d'interactions dans le diagramme de cas d'utilisation.
Sfsu (Niveau Scénario)	nbreScen	Numérique	4	Nombre total des scénarios dans tous les cas d'utilisation.
	nbreObj	Numérique	4	Nombre total d'objets utilisés dans tous les scénarios de tous les cas d'utilisation.
Cfsu (Niveau COSMIC-FFP)	nbreE	Numérique	4	Nombre de mouvements de données de type Entrée (E).
	nbreX	Numérique	4	Nombre de mouvements de données de type Sortie (X).
	nbreR	Numérique	4	Nombre de mouvements de données de type Lecture (R).
	NbreW	Numérique	4	Nombre de mouvements de données de type Écriture (W).
	nbreU	Numérique	4	Nombre de mouvements de données de type Imprécis (U).
LOC (Niveau ligne de code)	nbreLoc	Numérique	7	Taille du projet en lignes de code. Cette information n'est pas calculée par outil, elle est intégrée manuellement dans la base de données à des fins de calculs des ratios (Dans de futurs projets).

5.6.2 Table des codes d'accès

Identificateur	Type	Taille	Description
Code personnel	Texte	30	Code personnel identifiant le propriétaire.
Mot de passe	Crypté	6	Mot de passe crypté.


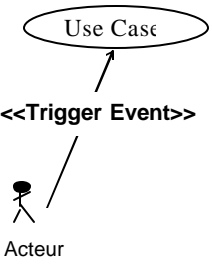
5.7 Description des interfaces utilisateur

Ce logiciel de calcul automatique de la taille fonctionnelle est construit sous forme d'un complément dans l'environnement de Rational Rose. Il est exécutable à partir d'une option nommée « COSMIC-FFP Functional Size » du menu « Tools » de Rational Rose. Son exécution procède au calcul de la taille fonctionnelle du projet en cours de conception. L'annexe 8.4 fournit, pour chaque interface, une copie de l'écran correspondant et une description détaillée du contenu. Il est à remarquer que seule la version française des différents interfaces utilisateur a été reproduite, cependant, l'exécution du logiciel peut se faire aussi dans la langue anglaise.

Ce dernier chapitre va conclure cette étude, il consiste d'une part, à faire un bilan de la prise en charge des notions de la méthode COSMIC-FFP dans le projet et, d'autre part, il sera effectué une analyse des résultats obtenus pour les deux études de cas.

6.1 Notions de la méthode COSMIC -FFP prises en charge dans le projet

Notion COSMIC-FFP	Prise en charge	Commentaires
Frontière du logiciel	Oui	La notion de frontière est automatiquement délimitée par la prise en charge de tous les processus fonctionnels du logiciel à mesurer.
Couche du logiciel	Partielle	La notion de couche COSMIC-FFP n'a pas d'équivalent direct en UML. Je pense que la tentative d'automatisation de l'opération de discernement des couches pourrait aboutir à des résultats imprévisibles. Aussi, à mon sens, il serait intéressant de laisser l'initiative au mesureur pour décider quels processus fonctionnels appartenant à quelle couche par l'entremise d'une procédure informatisée. Dans cette étude, les logiciels à mesurer sont considérés à couche unique.
Utilisateur COSMIC -FFP	Oui	La notion d'utilisateur COSMIC-FFP est en correspondance directe avec les acteurs UML.
Processus fonctionnel	Oui	La notion de processus fonctionnel est en correspondance directe avec les cas d'utilisation UML.
Mouvement de données	Oui	<p>Pour distinguer les quatre types de mouvements de données, il a été fait usage de la notion d'acteur et de la propriété « Persistent » des classes définies dans Rational Rose (Voir annexe 8.2, la procédure permettant d'activer la propriété « Persistent » d'une classe). Pour faire cette distinction, on applique les règles suivantes :</p> <ul style="list-style-type: none"> • Un mouvement de données de type Entrée (E) est caractérisé par un événement ou une opération UML qui s'opère dans le sens acteur (portant donc le stéréotype standard « Actor ») vers un objet UML. • Un mouvement de données de type Sortie (X) est caractérisé par un événement ou une opération UML qui s'opère dans le sens objet UML vers un acteur (portant donc le stéréotype standard « Actor »). • Un mouvement de données de type Lecture (R) est caractérisé par une opération qui s'opère entre deux objets UML et dont la classe de l'objet d'origine a la propriété « Persistent ». • Un mouvement de données de type Écriture (W) est caractérisé par une opération qui s'opère entre deux objets UML et dont la classe de l'objet de destination a la propriété « Persistent ». • Un mouvement de données qui n'est pas de l'un des quatre standards précédents est dit « Imprécis » ou « Undetermined ». Il est noté (E,X,R,W ?). Ce type survient lorsque le modèle conceptuel manque de précision, notamment dans les cas suivants : <ul style="list-style-type: none"> – Les objets émetteur et récepteur intervenant dans la réalisation d'une opération (envoi de message) n'ont pas de classes de base, – L'envoi de message (opération) entre deux objets dont les classes de base sont toutes les deux persistantes, – Aucune des classes des deux objets ne porte comme stéréotype 'Actor' et n'a pas la propriété 'Persistent'.

Événement déclencheur	Oui	Il n'existe pas d'équivalent direct en UML de la notion de déclencheur de la méthode COSMIC-FFP. Toutefois, il sera possible que le mesureur en désigne un déclencheur pour chaque processus fonctionnel. A cet effet, pour l'aider dans sa tâche, un nouveau stéréotype que j'ai conçu et dénommé « Trigger Event » et auquel est attribuée une icône caractérisée par une flèche brisée symbolisant un événement (voir schéma ci-contre) est mis au point. Ainsi, le choix de l'icône permettra de créer un message portant le stéréotype « Trigger Event » entre un acteur et un cas d'utilisation. (Voir annexe 8.3, la procédure permettant d'affecter le stéréotype « Trigger Event » à une interaction dans un cas d'utilisation).	icône du stéréotype	
			Stéréotype dans le diagramme des cas d'utilisation	
Groupe de données	Oui	La notion de groupes de données est en correspondance directe avec les objets UML.		
Attribut de données	Partielle	La notion d'attribut de données est en correspondance directe avec les attributs des objets UML. Toutefois, dans le cadre de ce projet, il n'a pas été nécessaire de mettre à contribution la notion d'attribut de données pour distinguer le type des mouvements de données, les notions de groupe de données, d'acteur et de la persistance des classes ont été suffisantes.		

6.2 Analyse des résultats de l'étude de cas « Rice Cooker »

Les spécifications de cette étude de cas peuvent être consultées en suivant le lien Internet cité en référence 3. Les résultats obtenus avec cette première étude de cas sont satisfaisants : le compte de la taille fonctionnelle obtenu avec l'outil (11 Cfsu) se rapproche bien du calcul manuel (12 Cfsu) à quelques remarques élémentaires près :

- Selon le calcul automatique, le résultat de la taille fonctionnelle est de 11 Cfsu, soit une unité de moins que le résultat du calcul manuel. Cet écart s'explique par la prise en compte lors du calcul manuel de l'événement déclencheur dans le processus fonctionnel « Control heater ». Or dans le cas des trois autres processus fonctionnels, l'événement déclencheur n'est pas inclus dans le compte de la taille fonctionnelle ! En conclusion, je dirai qu'il n'existe pas d'éléments techniques permettant à l'outil de décider de la prise en compte ou non d'un événement déclencheur comme mouvement de données à part entière.
- L'autre remarque concerne le processus fonctionnel « Select cooking mode », précisément le groupe de données utilisé par le mouvement de données « Receive cooking mode ». En effet, le groupe de données résultant du calcul automatique est « Operator », ce qui est bien le cas.

Le calcul manuel, quant à lui, donne comme résultat le groupe de données « Cooking mode » ; or ce groupe de données n'intervient pas dans la réalisation du mouvement de données « Receive cooking mode », il intervient plutôt, dans le processus « Write cooking mode ».

6.3 Analyse des résultats de l'étude de cas « Valve control »

Les spécifications de cette étude de cas peuvent être consultées en suivant le lien Internet cité en référence 4. Les résultats obtenus avec cette deuxième étude de cas sont aussi satisfaisants que la première : le compte de la taille fonctionnelle obtenu avec l'outil (21 Cfsu) se rapproche bien du calcul manuel (22 Cfsu), ci-après, trois remarques relevées :

- La taille fonctionnelle calculée est supérieure d'une unité au résultat manuel. Cela s'explique par la lecture séparée des deux « flag » A et B, ce qui ramène le compte des lectures à 13 et le compte total de la taille fonctionnelle à 21 au lieu de 20.
- La procédure de calcul automatique ne distingue pas individuellement tous les groupes de données qui sont localisés dans les mémoires RAM ou ROM. En effet, dans le diagramme de séquences, seuls les objets RAM et ROM sont représentés. Ils référencent respectivement les groupes de données flag

A, flag B, INIT, Cst_X, LT, UsIp, INCR pour ROM et T, ET, Erev, PSRev, Slp et Vs pour RAM. Ceci est dû au fait que la phase de conception d'un logiciel présente un niveau d'abstraction assez élevé qui ne peut pas distinguer les 13 objets individuellement : l'étendu de la vision se limite aux deux objets RAM et ROM.

- ⊙ À titre indicatif, si les classes des deux objets RAM et ROM ne sont pas définies comme étant persistante (avec la propriété « Persistent »), le nombre de mouvements de données dont le type est indéterminé serait de 14 sur 21. Toutefois, le compte total de la taille fonctionnelle n'est pas pour autant altéré.

6.4 Conclusion de l'analyse

Dans l'environnement de développement RUP, les résultats du calcul de la taille fonctionnelle selon la méthode COSMIC-FFP reflètent fidèlement ceux produits par une mesure manuelle. Toutefois, si quelques précautions ne sont pas prises, les résultats peuvent être altérés. Ci-après quelques sources possibles d'erreurs décelées suite à l'exécution des deux études de cas précédentes.

- ⊙ Le mesureur doit, au préalable, identifier de manière exhaustive les différentes couches du logiciel et définir quels processus fonctionnels appartenant à quelle couche. Ceci en respectant scrupuleusement les règles et procédures de la méthode COSMIC-FFP.
- ⊙ Aussi, si le concepteur transforme fidèlement les requis fonctionnels (FURs) en terme d'objets (Groupe de données) et d'opérations (Mouvements de données) qui réalisent le modèle en question, le résultat de la taille fonctionnelle ne devrait pas s'éloigner de la taille calculée manuellement. Toutefois, les types des mouvements de données peuvent ne pas correspondre à ceux escomptés et ce, si l'un des principes suivants n'est pas respecté :
 - Toute classe implémentant un utilisateur du système doit porter le stéréotype standard « Actor »
 - Toute classe implémentant une zone de stockage persistante doit avoir la propriété « Persistent »
 - Tout objet référencé dans un diagramme de séquence doit dériver d'une classe, autrement, il ne peut pas être considéré comme un objet persistant.
 - Un mouvement de données ne devrait pas se réaliser entre deux objets dont les classes sont toutes les deux persistantes.
 - Un mouvement de données ne devrait pas se réaliser entre deux objets dont les classes sont toutes les deux non persistantes et ne portant pas le stéréotype « Actor ».
 - Bien que ce projet n'a pas pour objectif d'évaluer l'aspect qualitatif des logiciels, il a été remarqué lors de l'opération de calcul, que tout manquement à la qualité du modèle de conception remonte en surface, je cite par exemple :
 - Les opérations (envoi de message) entre acteurs,
 - Les objets isolés qui ne participent à aucune opération,
 - Des processus fonctionnels dont la taille est inférieure à 2,
 - Les cas d'utilisation sans diagrammes de séquences,
 - L'ordre illogique des mouvements de données, Etc.

6.5 Limites du logiciel

Ce logiciel ne prétend pas automatiser le calcul de la taille fonctionnelle dans toutes ses dimensions, il présente les limites suivantes :

- ⊙ Ce logiciel traite le cas d'une seule couche. Comme il a été expliqué précédemment, à mon sens, il n'est pas intéressant de tenter une délimitation automatique par logiciel de la notion de couches. L'alternative que je propose est de laisser le mesureur définir quels processus fonctionnels appartenant à quelle couche.
- ⊙ Le logiciel mesuré doit être conçu avec Rational Rose et par une définition détaillée des diagrammes de séquences. À défaut, ceux-ci doivent être déduits des diagrammes de collaborations avant de procéder à l'opération de mesure ; ce qui peut se faire par une simple opération de transformation offerte par Rational Rose.

7. Conclusion générale



Ce projet de mémoire m'a été profitable à plus d'un titre. De prime abord, cela m'a permis d'approfondir mes connaissances concernant le standard de mesure qu'est la méthode COSMIC-FFF de mesure de la taille fonctionnelle de logiciels (ISO 19761 version 2.2)

En second lieu, le développement de l'outil de mesure automatique de la taille fonctionnelle selon les concepts COSMIC-FFP m'a permis d'une part, de bien pratiquer les concepts de base de la notation UML ainsi que de l'outil Rational Rose ; d'autre part, cela m'a permis d'apprendre le langage de développement natif à Rational Rose qu'est REI (Rose Extensibility Interface).

En effet, l'outil développé est basé sur les artefacts de Rational Rose, à savoir les diagrammes de cas d'utilisations, les diagrammes de séquences et les classes. Aussi, pour explorer les propriétés des artefacts d'un tel modèle conçu avec Rational Rose, l'utilisation de l'interface REI est très appropriée. J'ai alors profiter de l'occasion pour apprendre ce langage afin d'implémenter l'outil de mesure.

Par ailleurs, pour pouvoir tirer profit de cet outil de mesure automatique, il a été nécessaire de faire une investigation du processus RUP (Rational Unified Process) afin de déterminer à quels niveaux du cycle de développement il serait plus approprié de faire appel aux services de l'outil. Ce qui m'a permis alors de bien comprendre les principes clé du processus RUP et de prendre connaissance des principes du génie logiciel qui y sont mise en œuvre.

Quant aux résultats obtenus, ils sont très satisfaisants pour une première tentative d'automatisation du calcul de la taille fonctionnelle. En plus de l'expérimentation sur les deux études de cas dont les résultats ont été commentés dans le chapitre précédent, d'autres jeux d'essais ont confirmé la pertinence des résultats obtenus.

Pour résumer, l'outil permet d'évaluer la taille de logiciels développés selon le processus RUP et ce, à trois niveaux d'abstraction du cycle de développement, à savoir la définition des cas d'utilisations, le dénombrement des scénarios et finalement, le détail des messages échangés. Ces trois types de résultats sont consignés dans une base de données et seront d'une grande utilité pour déduire les « lois » qui régissent les ratios de passage d'un niveau d'abstraction à un autre en terme de la taille. Ainsi, par exemple, la connaissance de l'ampleur des cas d'utilisation permet d'évaluer la taille fonctionnelle et même la taille en lignes de code.

Ce qui est en définitive très bénéfique pour la planification de projet selon toutes ses perspectives. Toutefois, pour doter une entreprise d'un tel programme de calcul automatique de la taille fonctionnelle de logiciels, il est impératif d'instaurer une certaine discipline pour conférer au résultat une meilleure précision.



Annexe 8.1: Principes et règles de la méthode COSMIC-FFP

8.1.1 Règles permettant d'identifier les couches

Définition : Une couche (Layer) est le résultat du partitionnement fonctionnel de l'environnement du logiciel où tous les processus fonctionnels s'exécutent au même niveau d'abstraction. Pour identifier les couche d'un logiciel, on applique les principes et les règles qui suivent :

Principes :

- Toute couche délivre des fonctionnalités pour ses utilisateurs qui peuvent être des être humains, des éléments matériels ou logiciels.
- Une couche subordonnée offre des services à une couche cliente.
- Un logiciel d'une couche subordonnée doit fonctionner sans l'assistance de logiciels de ses couches clientes.
- Un logiciel d'une couche cliente ne doit pas fonctionner correctement si un logiciel d'une couche subordonnée dont il dépend ne l'est pas.
- Un logiciel d'une couche cliente n'utilise pas nécessairement toutes les fonctionnalités offertes par un logiciel d'une couche subordonnée.
- Un logiciel d'une couche subordonnée peut être à son tour client d'une autre couche subordonnée.
- Les logiciels des couches cliente et subordonnée peuvent partager physiquement les mêmes données, toutefois, les données doivent être interprétées différemment.
- Les logiciels qui partagent les mêmes données et qui les interprètent de la même façon ne peuvent pas être situés dans des couches différentes.

Règles :

- Les packages de fonctionnalités de service, tels que les systèmes de gestion des bases de données, les interfaces graphiques, les systèmes d'exploitations ou les « devices driver » sont généralement considérés comme des couches à part entière.
- Si un logiciel est conçu selon une architecture dont le paradigme est connu, on peut utiliser ce paradigme pour identifier les couches.
- Généralement, le niveau application est considéré comme le plus haut dans l'architecture des couches.
- Si le doute persiste, il est conseillé d'utiliser la notion de couplage pour distinguer les interactions entre les couches.

8.1.2 Règles permettant de délimiter la frontière du logiciel

Définition : La frontière d'un logiciel (Software Boundary) est la ligne conceptuelle séparant ce logiciel de l'environnement dans lequel il opère, tel que perçu par ses utilisateurs d'un point de vue externe. La frontière permet à la personne qui mesure de distinguer, sans ambiguïté, ce qui est inclus dans le logiciel de ce qui fait partie de l'environnement dans lequel il fonctionne. Pour identifier la frontière d'un logiciel, on applique les principes et les règles qui suivent :

Règles :

- Commencer par identifier les déclencheurs et les processus fonctionnels impliqués. La limite définie entre ces déclencheurs et ces processus fonctionnels constitue la frontière du logiciel à mesurer.
- Identifier les « composantes » d'entrée / sortie utilisées pour interagir avec les différents utilisateurs du système. Relier ces « composantes » aux fonctions réalisées. La limite définie entre ces « composantes » et ces fonctions constitue la frontière du logiciel à mesurer.
- Utiliser le concept de couche pour les systèmes temps réel ou les logiciels spécifiques.

8.1.3 Règles permettant d'identifier des éléments de base candidats

8.1.3.1 Processus fonctionnel

Définition : Un processus fonctionnel (Functional Process) est un ensemble unique et ordonné de mouvements de données (entrée, sortie, lecture et écriture) implantant un ensemble cohésif de requis fonctionnels. Il est déclenché directement ou indirectement via un acteur par un déclencheur. Il est complet lorsqu'il a réalisé tout ce qui doit être exécuté en réponse à un événement déclencheur. Pour identifier un processus fonctionnel, on applique les principes et les règles qui suivent :

Principes :

- Un processus fonctionnel doit être dérivé d'au moins un requis fonctionnel (FUR),
- Un processus fonctionnel est déclenché par un et un seul événement (trigger),
- Un processus fonctionnel contient au moins deux mouvements de données (une entrée et une sortie ou une écriture),
- Un processus fonctionnel ne doit contenir qu'un seul état d'attente (qui se produit après sa terminaison),
- Un processus fonctionnel doit appartenir à une et une seule couche.

Règles :

- Les déclencheurs qui se produisent suite au déclencheur principal ne peuvent pas à leur tour donner naissance à un processus fonctionnel,
- Dans le cas d'un système temps réel, un processus fonctionnel est aussi déclenché par un événement, il se termine quand un point de synchronisation est atteint.

8.1.3.2 Déclencheur

Définition : Un événement déclencheur (Triggering event) se produit à l'extérieur de la frontière du logiciel à mesurer et qui initie un ou plusieurs processus fonctionnels. Lorsque chaque couche identifiée est entourée par une frontière, des événements déclencheurs peuvent se produire dans une couche et initié un processus fonctionnel dans une autre couche.

8.1.3.3 Groupe de données

Définition : Un groupe de données (Data group) est un ensemble non ordonné, non vide et non redondant d'attributs de données, où chaque attribut décrit un aspect complémentaire du même objet d'intérêt. Pour identifier groupe de données, on applique les principes et les règles qui suivent :

Principes :

- Un groupe de données doit être matérialisé dans le système informatique qui supporte le logiciel à mesurer.
- Tout groupe de données doit être unique, cette unicité est matérialisée par ses attributs qui lui sont propres.
- Les groupes de données sont directement reliés aux objets définis dans les requis fonctionnels (FUR).

Règles :

- Les pratiques de mesure ont établi que pour les applications d'affaire (gestion), un groupe de données est directement lié à un objet de type entité du modèle de données du logiciel à mesurer.
- Pour les systèmes temps réel, un groupe de données suit l'une des formes suivantes :
 - Les mouvements de données de type entrée provenant de « devices » externes contiennent généralement des données d'état qui peuvent être éligibles comme groupes de données.
 - Les structures de données représentant des objets mentionnés dans les FURs et localisés dans la mémoire volatile et accessible aux processus fonctionnels.
 - Les structures de données représentant des graphes ou des tableaux de données mentionnés dans les FURs et localisés dans la mémoire permanente et accessible aux processus fonctionnels.

8.1.3.4 Attribut de données

Définition : Un attribut de données (Data attribute) est la plus petite parcelle d'information codée et significative du point de vue des requis fonctionnels des utilisateurs. Pour identifier un attribut de données, on applique les principes et les règles qui suivent :

Principes :

- Un attribut de données doit représenter un type de donnée valide.
- Un attribut de données doit représenter la plus petite quantité d'information référencée dans les requis fonctionnels (FUR).

Règles :

- Les types d'attributs de données valides sont :
 - Les données qui décrivent ou qui représentent le monde de l'utilisateur. En effet, celles-ci sont indépendantes de tout contexte d'implantation.
 - Les données contextuelles qui indiquent quoi, quand et comment d'autres données sont traitées.
- Les types d'attributs de données invalides sont généralement ceux qui se rapportent à la manière d'implanter le logiciel.

8.1.4 Règles permettant d'identifier les mouvement de données et leur type

8.1.4.1 Mouvement de données de type Entrée (Entry - E)

Définition : Une entrée déplace les valeurs d'attributs d'un groupe de données de la partie usager de la frontière du logiciel vers l'intérieur de la frontière du logiciel. (*une entrée ne met pas à jour la donnée qu'elle déplace*). Pour identifier un mouvement de données de type « Entrée », on applique les principes et les règles qui suivent :

Principes :

- Le mouvement de données reçoit des attributs de données de l'extérieur de la frontière du logiciel et du côté de l'utilisateur.
- Les données reçues proviennent d'un seul groupe de données. Si des données proviennent de plusieurs groupes de données, il faut identifier une « Entrée » pour chaque groupe de données.
- Le mouvement de données ne doit pas se confondre avec une Sortie, une Lecture ou une Écriture.
- Le mouvement de données identifié doit être unique dans un même processus fonctionnel.

Règles :

- Si un processus fonctionnel génère périodiquement des événements, ceux-ci ne sont pas considérés comme des Entrées(E), vu que les événements sont internes au processus fonctionnel.
- À moins qu'un processus fonctionnel spécifique l'exige, l'obtention du temps de l'horloge système n'est pas considérée comme une Entrée(E).

8.1.4.2 Mouvement de données de type Sortie (Exit - X)

Définition : Une sortie déplace les valeurs d'attributs d'un groupe de données de l'intérieur de la frontière du logiciel vers la partie usager de la frontière du logiciel. Pour identifier un mouvement de données de type « Entrée », on applique les principes et les règles qui suivent :

Principes :

- Le mouvement de données envoie des attributs de données à l'extérieur de la frontière du logiciel et du côté de l'utilisateur.
- Les données envoyées appartiennent à un seul groupe de données. Si des données appartiennent à plusieurs groupes de données, il faut identifier une « Sortie » pour chaque groupe de données.
- Le mouvement de données ne doit pas se confondre avec une Entrée, une Lecture ou une Écriture.
- Le mouvement de données identifié doit être unique dans un même processus fonctionnel.

Règle : Un seul mouvement de données de type Sortie(X) est pris en compte pour tous les messages en sortie d'un processus fonctionnel (message d'erreur, de confirmation, ...) à condition que le message ne contienne pas d'attributs de données.

8.1.4.3 Mouvement de données de type Lecture (Read - R)

Définition : Une lecture fait référence aux attributs de données d'un groupe de données sans en modifier les valeurs. Sur le plan fonctionnel, une lecture prend des données se trouvant à l'extérieur de la frontière du logiciel, dans la partie stockage, et les met à la disposition du processus fonctionnel auquel elle appartient. Pour identifier un mouvement de données de type « Entrée », on applique les principes et les règles qui suivent :

Principes :

- Le mouvement de données retrouve des attributs de données à partir d'un groupe de données localisé dans une zone de stockage.
- Les données lues doivent appartenir à un seul groupe de données. Si des données appartiennent à plusieurs groupes de données, il faut identifier une « Lecture » pour chaque groupe de données lues.
- Le mouvement de données ne doit pas se confondre avec une Entrée, une Sortie ou une Écriture.
- Le mouvement de données identifié doit être unique dans le processus fonctionnel.

8.1.4.4 Mouvement de données de type Écriture (Write - W)

Définition : Une écriture fait référence aux attributs de données d'un groupe de données et en modifie les valeurs. Sur le plan fonctionnel, une écriture envoie des données manipulées par le processus fonctionnel auquel elle appartient vers l'extérieur de la frontière du logiciel, dans la partie stockage. Pour identifier un mouvement de données de type « Entrée », on applique les principes et règles qui suivent :

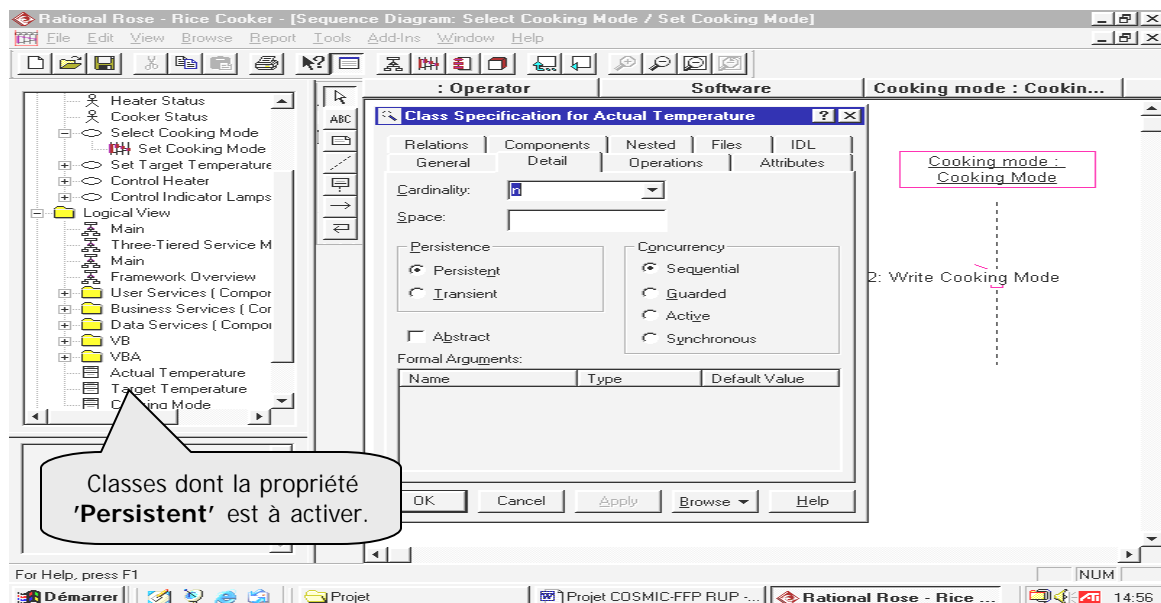
Principes :

- Le mouvement de données emmagasine des attributs de données dans un groupe de données localisé dans une zone de stockage
- Les données écrites doivent appartenir à un seul groupe de données. Si des données appartiennent à plusieurs groupes de données, il faut identifier une « Écriture » pour chaque groupe de données lues.
- Le mouvement de données ne doit pas se confondre avec une Entrée, une Sortie ou une Lecture.
- Le mouvement de données identifié doit être unique dans un même processus fonctionnel.

Annexe 8.2 : Procédure d'activation de la propriété « Persistent » d'une classe

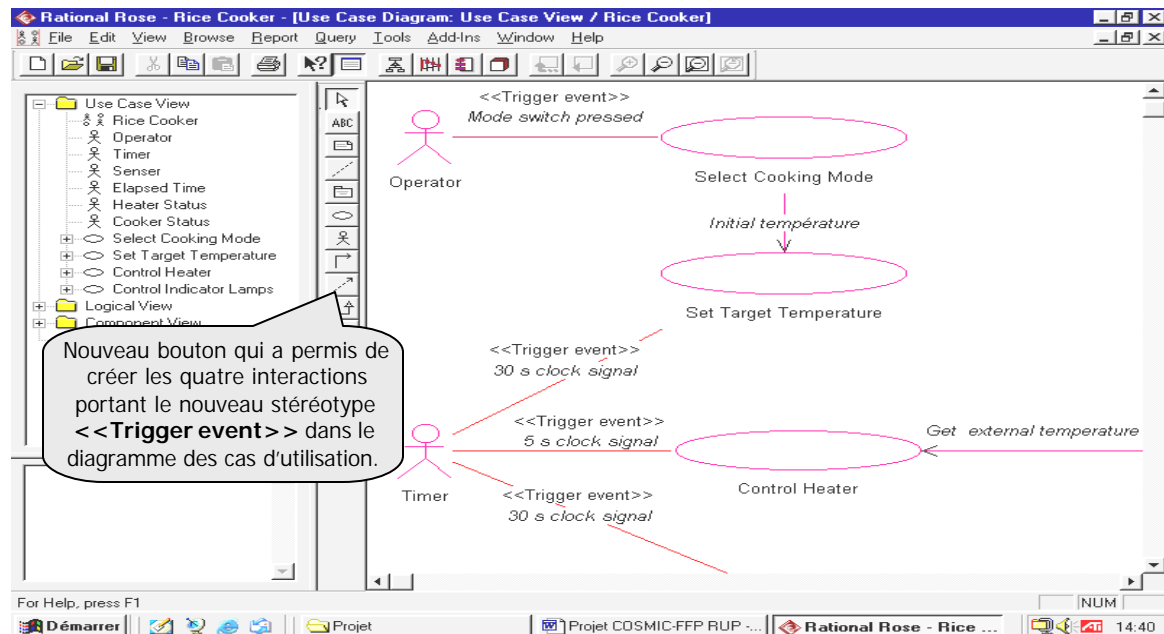
La procédure à suivre est la suivante :

1. Accéder à la vue logique du modèle (Logical view),
2. Repérer la classe en question et cliquer dessus avec le bouton droit de la souris,
3. Cliquer sur l'option « Open specification », une fenêtre apparaît,
4. Choisir l'onglet « Detail »,
5. Sélectionner l'option « Persistent » dans le pavé « Persistence »,
6. Appuyer sur « OK » pour valider l'opération.



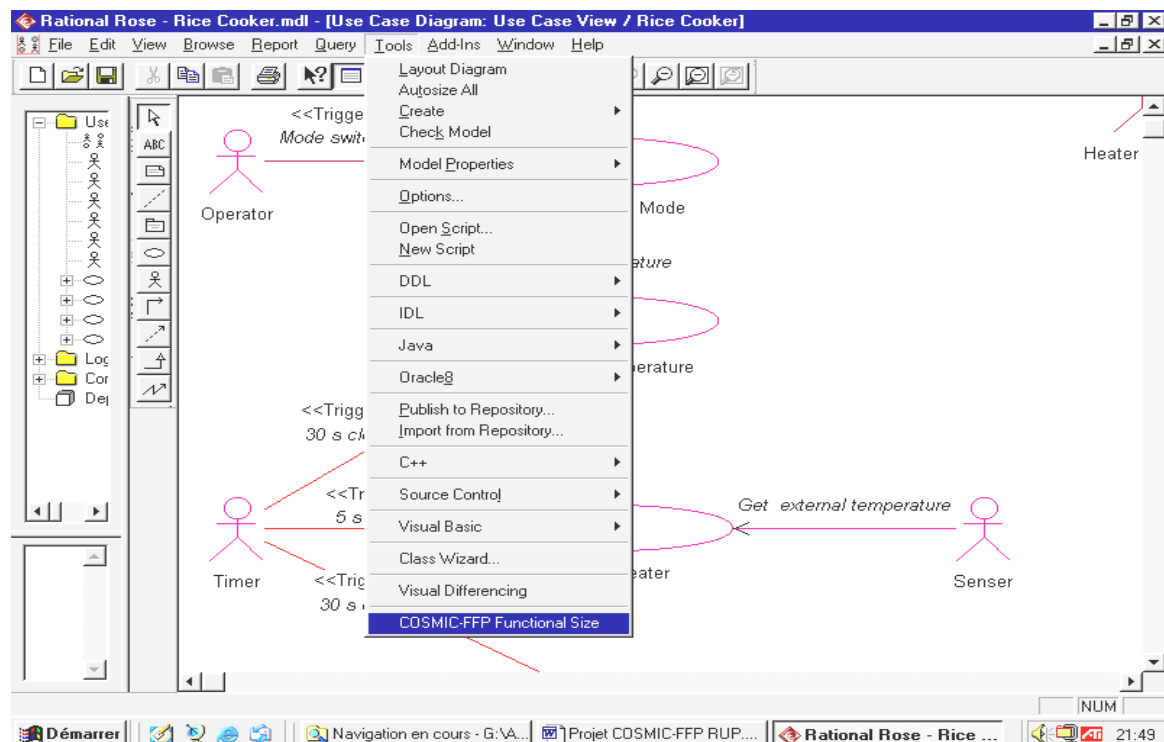
Annexe 8.3 : Procédure d'affectation du nouveau stéréotype « Trigger Event » à une interaction dans un cas d'utilisation

Dans la figure ci-après, il a été créé quatre interactions portant le nouveau stéréotype <<Trigger event>>. Ce qui permettra alors de localiser les événements déclencheur pour chaque processus fonctionnel.



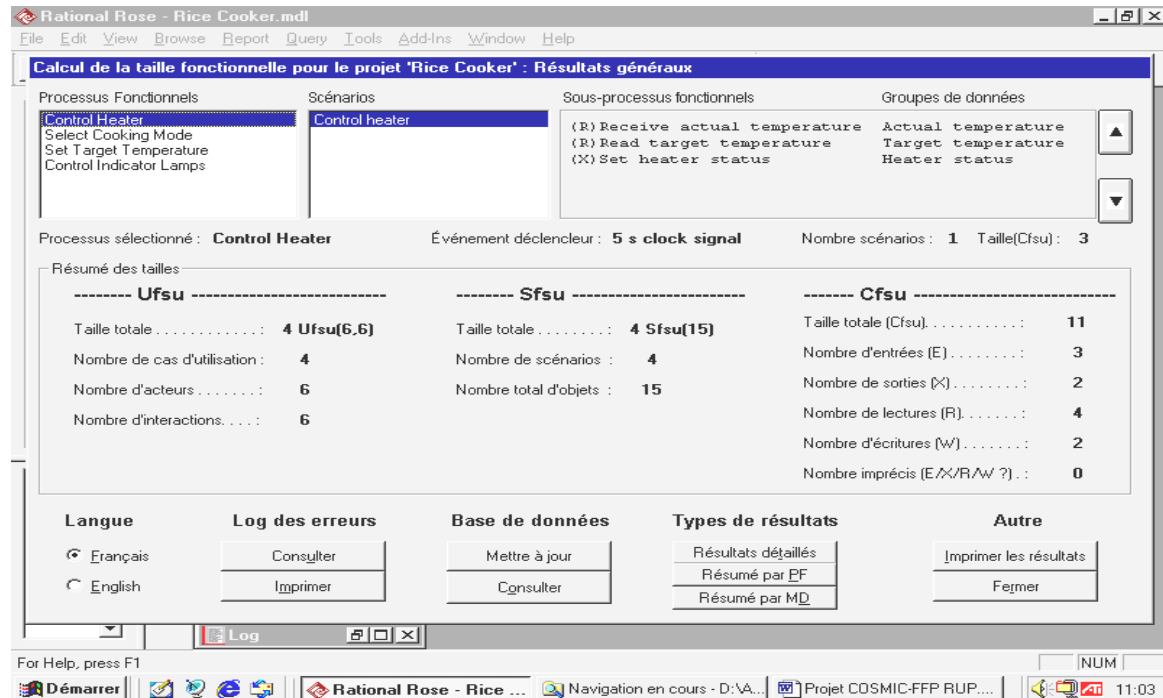
Annexe 8.4 : Prototypes des interfaces du logiciel

8.4.1 La procédure d'exécution du logiciel à partir de Rational Rose



L'interface précédent montre comment l'outil de calcul de la taille fonctionnelle est intégré dans le menu « Tools » de Rational Rose et comment procéder pour l'exécuter.

8.4.2 L'interface principal



Ce premier écran précise en titre le nom du projet mesuré et le type de résultat (dans ce cas, on est en présence de résultats généraux). Ces résultats sont présentés de deux façons :

- Quatre « liste-box » contenant respectivement :
 - Tous les processus fonctionnels du logiciel mesuré.
 - Tous les scénarios du processus fonctionnel sélectionné.
 - Tous les mouvements de données du processus fonctionnel sélectionné, chacun d'eux est précédé de son type : Entrée (E), Sortie (X), Lecture (R), Écriture(W) ou Imprécis (?). Le type imprécis est prévu pour représenter les cas où le modèle conceptuel qui a servi comme base de l'opération de mesure ne présente pas assez de précisions pour nous permettre de classer le mouvement de données dans l'un des quatre types standards (E,X,R et W).
 - Le groupe de données utilisé par chacun des mouvements de données.
 - Une ligne d'information indique le nom du processus fonctionnel sélectionné, l'événement déclencheur, le nombre de scénarios qui le constituent et sa taille en CFSUs.
- Un résumé des tailles à trois niveaux de détail est ensuite affiché :
 - La taille en Ufsu (niveau cas d'utilisation) exprimée en nombre de cas d'utilisations, nombre d'acteurs et le nombre d'interactions dans le diagramme des cas d'utilisations.
 - La taille en Sfsu (niveau scénario) exprimée en nombre total de scénarios et le nombre total d'objets impliqués (chaque objet est compté autant de fois qu'il apparaît dans tous les scénarios).
 - La taille en Cfsu (niveau mouvement de données) exprimée en nombre total de mouvements de données ainsi que le détail de chaque type : Entrée(E), Sortie (X), Lecture (R), Écriture(W) ou Imprécis (?).
- Une dizaine de fonctionnalités est listée en bas de d'écran. Chacune d'elles est décrite en détail dans les pages suivantes.

8.4.3 Le changement de langue

Le logiciel permet de travailler en français ou en anglais. Le français est la langue par défaut. Le basculement d'une langue à l'autre peut se faire à tout moment et une fois qu'on est positionné sur l'interface précédent des résultats généraux. Un changement de langue provoque une mise à jour automatique des informations sur l'écran en cours. La langue choisie est maintenue pour la session en cours aussi longtemps qu'un nouveau changement n'est opéré.

8.4.4 L'interface des résultats détaillés

Calcul de la taille fonctionnelle pour le projet 'Rice Cooker' : Résultats détaillés

No	PID	Description du processus	Déclencheur	Description de Sous-processus	Groupe de données	S-P Type	FFP Total
1	1.1	Control Heater	5 s clock signal	Receive actual temperature	Actual Temperature	R	1
				Read target temperature	Target Temperature	R	1
				Set heater status	Heater Status	X	1
3							
2	1.2	Select Cooking Mode	Mode switch pressed	Receive Cooking Mode	Operator	E	1
				Write Cooking Mode	Cooking Mode	W	1
2							
3	1.3	Set Target Temperature	30 s clock signal	Receive elapsed time	Elapsed Time	E	1
				Read cooking time	Cooking Mode	R	1
				Write target temperature	Target Temperature	W	1
3							
Total d'unités COSMIC-FFP :							11

OK

Cet interface affiche sous forme d'une liste déroulante, les résultats détaillés des processus fonctionnels. Pour chacun, on affiche la liste des mouvements de données qui le composent, soit :

- Son rang dans la liste des processus fonctionnels,
- Son identificateur (« Process ID »), composé du numéro de la couche à laquelle il appartient et le rang dans la couche (le numéro de couche est fourni à titre indicatif, la notion de couche n'est pas traitée dans ce projet),
- La description littérale du processus,
- Le déclencheur qui est à l'origine du processus,
- La liste de tous les mouvements de données qui le composent, à savoir :
 - la description,
 - le groupe de données mis en cause,
 - le type du mouvement de données: Entrée(E), Sortie (X), Lecture (R), Écriture(W) ou Imprécis (?),
 - la taille associée au mouvement de données,
 - la taille totale du processus fonctionnel.
- En résumé, est affichée la taille totale du logiciel.

8.4.5 L'interface des résultats agrégés par processus fonctionnel

Calcul de la taille fonctionnelle pour le projet 'Rice Cooker' : Résumé par processus fonctionnel

No	Processus ID	Description du processus	Sous-Processus FFP					Points FFP (Cfsu)	Proportion dans le projet (%)
			E	X	R	W	U		
1	1.1	Control Heater	0	1	2	0	0	3	27,27 %
2	1.2	Select Cooking Mode	1	0	0	1	0	2	18,18 %
3	1.3	Set Target Temperature	1	0	1	1	0	3	27,27 %
4	1.4	Control Indicator Lamps	1	1	1	0	0	3	27,27 %
Résumé :		4 Processus	3	2	4	2	0	11	100 %

OK

Cet interface affiche sous forme d'une liste déroulante, les résultats agrégés par processus fonctionnel à savoir :

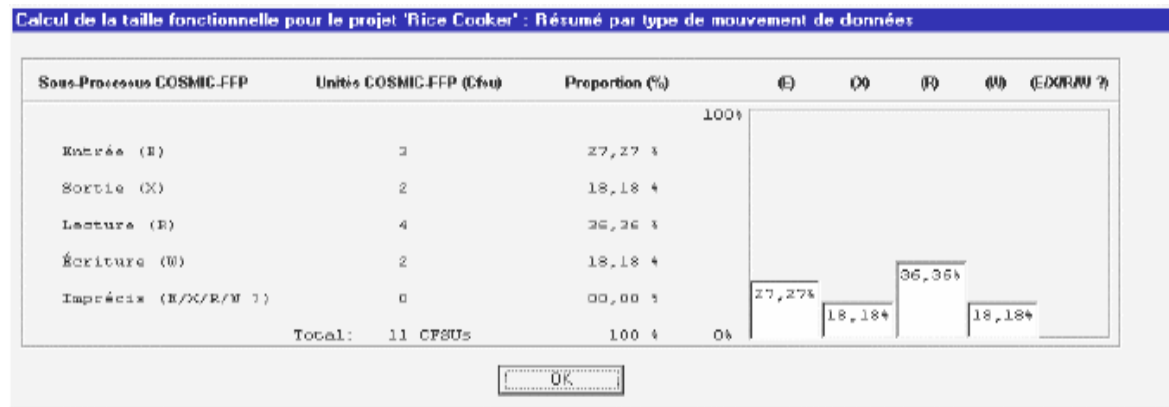
- Son rang dans la liste des processus fonctionnels,
- Son identificateur (« Process ID »), composé du numéro de la couche à laquelle il appartient et le rang dans la couche (le numéro de couche est fourni à titre indicatif, la notion de couche n'est pas traitée dans ce projet),
- La description littérale du processus,
- Le nombre de mouvements de données de type Entrée (E),
- Le nombre de mouvements de données de type Sortie (X),
- Le nombre de mouvements de données de type Lecture (R),

- Le nombre de mouvement de données de type Écriture (W),
- Le nombre de mouvements de données de type Imprécis (U),
- La taille totale du processus fonctionnel en Cfsu,
- La proportion en taille du processus fonctionnel dans le logiciel mesuré.

En résumé, il est affiché :

- Le nombre total de processus fonctionnel,
- Le nombre total de mouvements de données de type Entrée (E),
- Le nombre total de mouvements de données de type Sortie (X),
- Le nombre total de mouvements de données de type Lecture (R),
- Le nombre total de mouvements de données de type Écriture (W),
- Le nombre total de mouvements de données de type Imprécis (U),
- Et finalement, la taille totale du logiciel.

8.4.6 L'interface des résultats agrégés par mouvement de données



Cet interface affiche les résultats agrégés par mouvement de données, à savoir :

- Le type du mouvement de données (Entrée(E), Sortie (X), Lecture (R), Écriture(W) ou Imprécis (E,X,R,W ?)),
- La taille totale de chaque type de mouvement de données en Cfsu,
- La proportion en taille de chaque mouvement de données dans le logiciel mesuré,
- Finalement, il est affiché un graphique représentant ces proportions pour chaque type de mouvement de données

8.4.7 Impression des résultats détaillés

Calcul de la taille fonctionnelle pour le projet 'Rice Cooker' : Détail de la mesure - Classeur1

No	PID	Description du processus	Déclencheur	Description de Sous-proc	Groupes de donn	Type de sou	Unités COSM	Tot
1	1,1	Control Heater	5 s clock signal	Receive actual temperature	Actual Temperat	Lecture (R)	1	
2				Read target temperature	Target Temperat	Lecture (R)	1	
3				Set heater status	Heater Status	Sortie (X)	1	
								Total Partiel 3
4	1,2	Select Cooking Mode	Mode switch pressed	Receive Cooking Mode	Operator	Entrée (E)	1	
5				Write Cooking Mode	Cooking Mode	Écriture (W)	1	
								Total Partiel 2
6	1,3	Set Target Temperature	30 s clock signal	Receive elapsed time	Elapsed Time	Entrée (E)	1	
7				Read cooking time	Cooking Mode	Lecture (R)	1	
8				Write target temperature	Target Temperat	Écriture (W)	1	
								Total Partiel 3
9	1,4	Control Indicator Lamps	30 s clock signal	Receive elapsed time	Elapsed Time	Entrée (E)	1	
10				read cooking mode	Cooking Mode	Lecture (R)	1	
11				Send Status to Lamps	Cooker Status	Sortie (X)	1	
								Total Partiel 3
								Total général 11

Une copie des résultats détaillés par processus fonctionnel et par mouvements de données qui le composent est envoyée dans un document Microsoft Excel. Cette copie Excel peut être sauvegardée ou imprimée, en outre, elle se prête bien aux fins d'analyses statistiques ou autre.

8.4.8 Consultation de l'historique des mesures

Calcul de la taille fonctionnelle pour le projet 'Rice Cooker' : Consultation de l'historique des mesures

Identification du projet	Type	Ufsu			Sfsu		Cfsu					LOCs
		Cas util.	Acteurs	Interac.	Scénarios	Objets	Entrées(E)	Séries(S)	Lectures(R)	Écritures(W)	(E/S/R/W?)	
Cosmic	2	5	2	9	4	0	7	7	0	0	4	0
Gestion	1	5	2	9	4	0	7	7	0	0	4	3
Gestions	4	5	2	9	4	0	7	7	0	0	4	0
Gestionngg	1	5	2	9	4	0	7	7	0	0	4	0
Gestion Glossaire1234	3	5	2	9	4	0	7	7	0	0	4	0
Rice Cooker	3	4	6	6	4	15	3	2	3	1	2	33
Valve Contrôle	4	1	5	1	1	9	4	1	10	3	0	6
lvev	1	0	0	0	0	0	0	0	0	0	0	0
Evvev	2	0	0	0	0	0	0	0	0	0	0	0
3vfv	3	0	0	0	0	0	0	0	0	0	0	0
4vev	1	0	0	0	0	0	0	0	0	0	0	0
5vfv	1	0	0	0	0	0	0	0	0	0	0	0
6vevva	1	0	0	0	0	0	0	0	0	0	0	0

Codes des types de projets :
 1-Gestion 2-Temps réel
 3-Emballé 4-Autre

Ok

L'historique des mesures contient pour chaque projet mesuré les informations suivantes:

- L'identificateur du projet et son type,
- La mesure en Ufsu contenant :
 - Le nombre de cas d'utilisation,
 - Le nombre d'acteurs,
 - Le nombre d'interactions dans le diagramme de cas d'utilisation.
- La mesure en Sfsu contenant :
 - Le nombre de scénarios,
 - Le nombre d'objets participants,
- La mesure en Cfsu contenant le nombre de mouvements de données de type E, X, R, W et U.

8.4.9 Impression de la liste des erreurs

Calcul de la taille fonctionnelle pour le projet 'Rice Cooker' : Liste des erreurs - Classeur1

	A	B	C	D
1	Numéro	1		
2	Cas d'utilisation	Control Heater		
3	Scénario	Control heater		
4	Message	Receive actual temperature		
5	Description de l'erreur	Type de message imprécis entre les objets 'Actual temperature' et 'System'.		
6	Solution proposée	Définir la classe de l'un des objets avec la propriété 'Persistent'		
7				
8	Numéro	2		
9	Cas d'utilisation	Control Heater		
10	Scénario	Control heater		
11	Message	Read target temperature		
12	Description de l'erreur	Type de message imprécis entre les objets 'Target temperature' et 'System'.		
13	Solution proposée	Définir la classe de l'un des objets avec la propriété 'Persistent'		
14				
15	Numéro	3		
16	Cas d'utilisation	Set Target Temperature		
17	Scénario	Set Target Temperature		
18	Message	Write target temperature		
19	Description de l'erreur	Type de message imprécis entre les objets 'System' et 'Target temperature'.		
20	Solution proposée	Définir la classe de l'un des objets avec la propriété 'Persistent'		
21				
22				
23				
24				
25				
26				

Liste des erreurs / Feuil1 / Feuil2 / Feuil3 /

Prêt

Démarrer | Rational Rose - Ri... | Navigation en cou... | Projet COSMIC-FF... | Calcul de la ta... | 11:29

Pour chaque erreur, il est précisé le cas d'utilisation, le scénario et le message éventuel où elle s'est produite ainsi qu'une description de l'erreur et une solution proposée.

9. Bibliographie



- [1] **Measurement Manual (The COSMIC Implementation Guide For ISO/IEC 19761:2003)** version 2.2, de janvier 2003 :
<http://132.208.130.41/FFPguide/FFPguideRegister.html>
- [2] The **ISO/IEC TR 14143 -4:2002** Information technology -- Software measurement -- Functional size measurement -- Part 4: Reference model
- [3] **Rice Cooker - Cosmic Group Case Study**, version du 26 janvier 2003
<http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/>
- [4] **Valve Control System - Cosmic Group Case Study**, version du 25 janvier 2003
<http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/>
- [5] **Booch, Grady; Rumbaugh, James et Jacobson, Ivar.** *"The unified modeling language user guide"*, Addison-Wesley , c1999
- [6] **Valéry Béro, Ghislain Lévesque et Alain Abran**, « *Application de la méthode FFP à partir d'une spécification selon la notation UML: Comte-rendu des premiers essais d'application et questions.* »
- [7] **Philippe Kruchten**, *"The rational unified process, an introduction"*
2000, addison wesley
- [8] **Philippe Kruchten**, *"The RUP platform"*
Présentation de Philippe Kruchten au SPIN de Montréal 14/01/03
- [9] **Chantal Bouthat**, « *Guide de présentation des mémoires et thèses* » 1993
- [10] Brochure – Directives pour le projet en génie logiciel
Remise par M. Bouisset.