

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

APPLICATION DE LA MESURE FONCTIONNELLE COSMIC-FFP :
UNE APPROCHE COGNITIVE

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
JEAN-MARC DESHARNAIS

NOVEMBRE 2003

AVANT-PROPOS

Je remercie spécialement mes directeurs Alain Abran et André Mayers qui m'ont soutenu et encouragé tout au long de ma recherche doctorale.

Je remercie les étudiants stagiaires suivants : Solange Ndagijimana qui a fait une étude de la qualité des logiciels dans le cadre de son mémoire de maîtrise, Tim Küessing qui m'a beaucoup aidé dans la réalisation du prototype 1 dans le cadre de son mémoire de maîtrise, Julien Vilz et François Gruselin qui m'ont apporté leur support lors de la vérification de la base de connaissances du prototype 2 dans le cadre de leurs mémoires de maîtrise.

Je remercie Pierre Bourque et Éric Lefebvre qui m'ont permis d'expérimenter avec leurs étudiants de maîtrise le premier prototype pour l'application de la méthode mesure COSMIC-FFP. Je remercie les étudiants qui ont bien voulu prendre une journée entière à apprendre et à appliquer la méthode de mesure COSMIC-FFP.

Je remercie aussi les personnes suivantes : Denis St-Pierre, Francis Dion, Vinh Tuong Ho et Hassan Diab, pour avoir révisé et commenté, à titre d'experts, la base de connaissances du premier prototype.

Je remercie Michèle Hébert, Marthe Lacroix, Suzanne Paquette et Annie Saulnier pour la révision technique de ce document.

Je remercie aussi mes proches dont Saniye pour leur compréhension et leur patience.

AVANT-PROPOS.....	II
LISTE DES FIGURES.....	VII
LISTE DES TABLEAUX.....	IX
ABRÉVIATIONS, SIGLES ET ACRONYMES	X
GLOSSAIRE.....	XI
INTRODUCTION	1
1. CONTEXTE.....	3
1.1 Brève présentation de COSMIC-FFP	4
1.2 Les concepts de COSMIC-FFP.....	7
1.3 Origine de nos recherches	9
2. REVUE DE LITTÉRATURE ET PROBLÉMATIQUE.....	12
2.1 Domaine de la mesure en génie logiciel	13
2.1.1 Définition de l'unité de mesure.....	14
2.1.2 Les classes de mesure du logiciel.....	16
2.1.3 Schème de classification des mesures.....	16
2.1.4 La mesure fonctionnelle	17
2.1.5 Processus de développement et mesure fonctionnelle	20
2.1.6 Le processus de mesure et la taille du logiciel.....	23
2.1.7 Catégories d'utilisateurs de la mesure fonctionnelle	26
2.1.8 Reconnaissance de la méthode de mesure fonctionnelle.....	27
2.1.9 Validation d'une méthode de mesure	29
2.1.10 Qualité de la documentation	32
2.1.11 Constatations et problématique de la mesure fonctionnelle	35
2.2 Domaine cognitif	36
2.2.1 Définition d'un système à base de connaissances.....	37
2.2.2 Composants d'un système à base de connaissances.....	40
2.2.3 Nature et objectifs des systèmes à base de connaissances.....	53
2.2.4 Justification d'un système à base de connaissances et son contenu.....	55
2.2.5 Modélisation de la connaissance.....	65
2.2.6 Vérification des modèles de connaissances.....	70
2.2.7 Originalité de l'approche cognitive choisie et conclusion	71
2.3 Problématiques et objectifs de la recherche	72
3. MÉTHODOLOGIE DE RECHERCHE.....	75

3.1	Cheminement du mesureur.....	76
3.2	Autres contributions théoriques	78
3.3	Cheminement détaillé	79
3.4	Traitement des connaissances.....	80
3.5	Design d'exemples de cas problèmes	82
3.6	Design d'un premier prototype	82
3.7	Vérification auprès des experts.....	83
3.8	Expérimentation avec les débutants.....	84
3.9	Construction du deuxième prototype	87
3.10	Vérification.....	87
3.11	Validation.....	88
4.	FORMALISATION DE LA DÉMARCHE DU MESUREUR.....	90
4.1	Cheminement du mesureur et problématique de la mesure.....	90
4.1.1	Comprendre	91
4.1.2	Interpréter.....	92
4.1.3	Utiliser.....	93
4.1.4	Solutionner.....	94
4.1.5	Exemples	94
4.2	Cheminement du mesureur et autres modèles de mesures.....	96
4.3	Le mesureur et la méthode de type diagnostic	98
4.3.1	Méthodes et modèles de construction du logiciel.....	99
4.3.2	Connaissance de la méthode de mesure COSMIC-FFP.....	99
4.3.3	Modélisation de la connaissance du mesureur.....	102
4.4	Méthode de type diagnostic	109
4.4.1	Les étapes et les sous-étapes de la méthode	109
4.4.2	Description des étapes de la méthode de type diagnostic	111
4.4.3	Création des cas problèmes	115
4.5	Exemple d'application des règles.....	115
5.	PRÉSENTATION ET EXPÉRIMENTATIONS DU PROTOTYPE 1.....	120
5.1	Présentation des exigences du prototype 1.....	120
5.2	Vérification et expérimentation du prototype 1.....	124
5.2.1	Vérification par des experts du prototype 1.....	125
5.2.2	Expérimentation auprès de mesureurs débutants avec le prototype 1	128
5.2.3	Observations sur la vérification avec les experts et les expérimentations avec les mesureurs débutants.....	143
6.	PRÉSENTATION ET VÉRIFICATION DU PROTOTYPE 2.....	146
6.1	Interface du mesureur.....	147
6.2	Approche de réingénierie	149
6.3	Choix du langage formel.....	150
6.4	Définition des types de fichiers et leurs relations	150
6.4.1	RTF et HTML.....	151

6.4.2	XML	152
6.4.3	XML schéma.....	152
6.4.4	XSL.....	152
6.4.5	XHTML.....	153
6.5	Description des types de données de la base de connaissances	153
6.5.1	InsideP.....	157
6.5.2	Mot-clé	160
6.5.3	Concept topologique	161
6.5.4	Cas problème	162
6.5.5	Thème	163
6.5.6	Recommandation.....	164
6.6	Procédure de transformation.....	165
6.6.1	1° Cycle	165
6.6.2	2° Cycle	166
6.6.3	3° Cycle.....	167
6.7	Détecter les redondances.....	168
6.8	Définir une nouvelle organisation de l'information	170
6.8.1	Unités de stockage.....	171
6.8.2	Définition des nouveaux types de documents.....	172
6.8.3	Concept topologique	174
6.8.4	Cas problème	176
6.8.5	Étude de cas.....	177
6.8.6	Expert.....	178
6.9	Définir les vues.....	179
6.9.1	Glossaire.....	179
6.9.2	Concept topologique	180
6.9.3	Cas problème	180
6.9.4	Étude de cas.....	181
6.9.5	Expert.....	181
6.10	Traduire les documents	181
6.11	Vérifications avec le prototype 2	182
6.11.1	Déroulement de la vérification.....	182
6.11.2	Résultats pour la vérification	185
6.11.3	Déroulement de la vérification des recommandations	187
6.11.4	Résultats de la vérification avec le prototype 2	189
7.	SOMMAIRE DES RÉSULTATS.....	191
7.1	Problématique, objectifs, méthodologie et résultats.....	191
7.2	Réalisations	192
7.3	Recherches à venir	195
	BIBLIOGRAPHIE.....	196

PRÉSENTATION DES ANNEXES.....	202
ANNEXE A : Description de Help!CPR.....	205
ANNEXE B : Description du prototype 1.....	212
ANNEXE C : Description de l'environnement de développement du prototype 2.....	284
ANNEXE D : Procédure de vérification de la base de connaissances du prototype 1.....	293
ANNEXE E : Curriculum des experts pour le prototype 1.....	296
ANNEXE F : Liste des associations internationales : mesure fonctionnelle	302
ANNEXE G : Exemples de « règles locales »	304
ANNEXE H : Études de cas	309
ANNEXE I : Formules du prototype 1.....	320
ANNEXE J : Analyses statistiques.....	325
ANNEXE K : Résultats des bonnes réponses.....	333
ANNEXE L : Exemples de mauvaises réponses.....	337
ANNEXE M : Spécifications présentées aux mesureurs débutants.....	347

LISTE DES FIGURES

Figure 1-1 : Méthode générale du processus de mise en correspondance COSMIC-FFP.....	5
Figure 1-2 : Procédure générale de la phase de mesure COSMIC-FFP.....	6
Figure 1-3 : Concepts de COSMIC-FFP [10].....	7
Figure 1-4 : Carte conceptuelle – méthode IFPUG [13].....	10
Figure 2-1 : Démarche de la revue de littérature.....	12
Figure 2-2 : Démarche de la revue de littérature pour la mesure.....	13
Figure 2-3 : Les trois dimensions du logiciel [22].....	19
Figure 2-4 : Modèle en cascade [19].....	22
Figure 2-5 : Phases du processus de mesure [29].....	23
Figure 2-6 : FUR : documentation (de haut en bas) [2, section 2.2].....	25
Figure 2-7 : FUR : documentation (de bas en haut) [2, section 2.2].....	25
Figure 2-8 : Qualité de la documentation.....	34
Figure 2-9 : Démarche de revue de littérature pour le domaine cognitif.....	37
Figure 2-10 : Cycle du CBR (adaptation à partir de Watson).....	58
Figure 3-1 : Étapes de la méthodologie de recherche.....	76
Figure 3-2 : Cheminement détaillé.....	79
Figure 3-3 : Traitement des connaissances.....	81
Figure 3-4 : Expérimentation avec les débutants.....	85
Figure 4-1 : Modèle du cheminement du mesureur.....	91
Figure 4-2 : La compréhension du mesureur à trois niveaux d'abstraction.....	92
Figure 4-3 : Logiciels et COSMIC-FFP (interpréter).....	93
Figure 4-4 : Cheminement du mesureur, activités de la phase 2 et COSMIC-FFP.....	97
Figure 4-5 : Mouvements de données.....	101
Figure 4-6 : Étapes et tâches de la méthode de mesure COSMIC-FFP.....	101
Figure 4-7 : Types de connaissances – inspiré de [60].....	103
Figure 4-8 : Procédure d'application.....	104
Figure 4-9 : Types d'inférences.....	106
Figure 4-10 : Ontologie COSMIC-FFP [10].....	107
Figure 4-11 : Étapes et sous-étapes de la méthode de type diagnostic.....	110
Figure 5-1 : Diagramme de cas.....	122
Figure 6-1 : Interface du mesureur (concept topologique).....	147
Figure 6-2 : Interface du mesureur (cas problèmes et thèmes).....	148
Figure 6-3 : Interface du mesureur (recommandation).....	148
Figure 6-4 : Présentation des types de fichiers.....	151
Figure 6-5 : Modèle du prototype 2.....	154
Figure 6-6 : Type TextHTML.....	155
Figure 6-7 : Choix.....	156
Figure 6-8 : Type de données InsideP.....	157
Figure 6-9 : Liste ordonnée.....	158
Figure 6-10 : Liste non ordonnée.....	158
Figure 6-11 : Séquence.....	158

Figure 6-12 : Type de données Mot-clé	160
Figure 6-13 : Type de données Concept topologique	161
Figure 6-14 : Type de données Cas problème	162
Figure 6-15 : Type de données Thème	163
Figure 6-16 : Type de données Recommandation	164
Figure 6-17 : Éléments redondants des types de données	169
Figure 6-18 : Liens contenus dans la base de données.....	170
Figure 6-19 : Liens entre les types de fichiers	172
Figure 6-20 : Type de document Glossaire.....	173
Figure 6-21 : Type de document Concept topologique	174
Figure 6-22 : Type de document Cas problème	176
Figure 6-23 : Type de document Étude de cas	177
Figure 6-24 : Type de document Expert.....	178
Figure 6-25 : Liens entre les différents types de documents utilisateurs.....	184

LISTE DES TABLEAUX

Tableau 2-1 : Analyses de fiabilité de la mesure fonctionnelle par Abran [21].....	31
Tableau 2-2 : Renforcement des croyances (règles).....	51
Tableau 4-1 : Exemple d'application des règles.....	116
Tableau 4-2 : Thème 1 selon les faits.....	117
Tableau 4-3 : Thème 2 selon les faits.....	117
Tableau 4-4 : Selon deux faits (positifs).....	118
Tableau 4-5 : Selon deux faits (positif et négatif)	118
Tableau 4-6 : Selon deux faits (négatif)	119
Tableau 5-1 : Scénarios du mesureur.....	123
Tableau 5-2 : Scénarios de l'expert et de l'administrateur.....	124
Tableau 5-3 : Résultats de la vérification par les 4 experts – prototype 1.....	126
Tableau 5-4 : Nombre de bonnes et de mauvaises réponses par participant et par groupe avec ou sans prototype	135
Tableau 6-1 : Éléments de textHTML.....	156
Tableau 6-2 : <i>Persistence of Data Group</i>	159
Tableau 6-3 : Documentation du type de document Glossaire	173
Tableau 6-4 : Documentation du type de document Concept topologique.....	174
Tableau 6-5 : Documentation du type de document Cas problème	176
Tableau 6-6 : Documentation du type de document Étude de cas.....	178
Tableau 6-7 : Documentation du type de document Expert.....	179

ABRÉVIATIONS, SIGLES ET ACRONYMES

ASMA :	Australian Software Metrics Association
CBR :	Case Based Reasoning
CFUT :	COSMIC-FFP unité de taille
CIM :	Centre d'intérêt sur les métriques
COSMIC :	COmmon Software Measurement International Consortium
COSMIC-FFP :	COmmon Software Measurement International Consortium – Full Function Point
DESMA :	Deutsche Anwendergruppe für Software Metrik und Aufwandschätzung (German Software Metrics Association)
EI :	External Input
EIF :	External Interface File
EO :	External Output
EQ :	External Query
ERM :	Empirical Research Model
FUR :	Functional User Requirements (Fonctionnalités Utilisateur Requisites)
GUFPI :	Gruppo Utenti Function Point Italia (Italian Software Metrics Association)
HTML :	Hypertext Markup Language
IDMS :	Integrated Database Management System
IFPUG :	International Function Point Users Group
ILF :	Internal Logical File
JFPUG :	Japanese Function Point Users Group
KADS :	Knowledge-Base System Analysis and Design Support
KBS :	Knowledge-Base System
LOC :	Line of Code
LRGL :	Laboratoire de Recherche en Gestion du Logiciel
NESMA :	Netherlands Software Metrics Users Association
RTF :	Rich Text Format
SWEBOK :	SoftWare Engineering Body of Knowledge
UML :	Unified Modeling Language
W3C :	World Wide WEB Consortium
XML :	eXtensible Markup Language (norme W3C)
XHTML :	un HTML défini pour rencontrer la norme XML
XSL :	Extensible Style Language

GLOSSAIRE

Voici la liste des termes que nous avons utilisés dans cette thèse. Plusieurs de ces termes proviennent du glossaire de COSMIC-FFP.

Administrateur : C'est la personne qui gère les accès au prototype en inscrivant les noms des utilisateurs selon leur type (mesureur et expert) ainsi que les mots de passe et la durée des accès.

Attributs (type) : C'est la plus petite parcelle d'information codée, dans un groupe de données, portant une signification dans la perspective fonctionnelle des besoins de l'utilisateur.

Cfut : COSMIC-FFP unité de taille. Étalon de mesure de COSMIC-FFP (en anglais Cfsu).

Base des faits : Les réponses fournies aux thèmes.

Cas : Pour Janet Kolodner, c'est un "morceau" de connaissance "contextualisé" et représentant une expérience [1, p. 193].

Cas problème : C'est un cas pour lequel le mesureur doit utiliser un processus de diagnostic pour identifier correctement un concept de l'ontologie COSMIC-FFP (ex. : identifier un groupe de données ou encore identifier un processus).

Concept topologique : Dans le contexte de notre recherche, ce sont des concepts qui servent à établir des liens entre les concepts de l'ingénierie du logiciel et ceux de COSMIC-FFP.

Connaissance : Un mode d'emploi permettant de transformer les données en information. Une machine intellectuelle pour réaliser un but et potentiellement générer de nouvelles informations.

COSMICXpert : Nom générique donné à l'outil logiciel de notre système à base de connaissances. Aussi appelé prototype 1 et 2.

Couche : C'est le résultat du partitionnement fonctionnel de l'environnement du logiciel où tous les processus fonctionnels inclus montrent un haut degré de cohésion et s'exécutent au même niveau d'abstraction.

Dans un environnement logiciel à plusieurs couches, celles-ci interagissent les unes avec les autres via leurs processus fonctionnels respectifs. Ces interactions sont hiérarchiques par nature ; lorsque considérée par paire, une couche est "cliente" de l'autre. Une couche "cliente" utilise les services fonctionnels fournis par des couches subordonnées. Les éléments du logiciel dans la même couche peuvent aussi échanger des données. Ce type d'échange de données est habituellement appelé un échange de données *peer-to-peer*.

Donnée : Un signal non interprété.

Écriture (type) : C'est un type de mouvement de données qui déplace un groupe de données se trouvant à l'intérieur d'un processus fonctionnel vers une partie de stockage.

Remarque : Une écriture inclut les manipulations de données.

Entrée (type) : C'est un type de mouvement de données qui déplace un groupe de données depuis un utilisateur à travers la frontière vers le processus fonctionnel où il est requis.

Remarque 1 : Dans COSMIC-FFP, une entrée inclut aussi les manipulations (ex. : validation de la donnée entrante) associées.

Remarque 2 : Une manipulation de données est tout ce qui se passe sur une donnée sauf un mouvement de données.

Environnement opérationnel logiciel : C'est une collection de logiciels exécutés de façon concurrente sur un ordinateur spécifique.

Étude de cas : Comprend plusieurs cas problèmes ayant une logique fonctionnelle du point de vue d'un utilisateur d'un logiciel.

Remarque : Dans le système à base de connaissances de notre recherche nous proposons une centaine de cas problèmes avec les études de cas suivantes : Data Warehouse, Rice Cooker, Valve Control et Report Generator (Annexe H).

Événement déclencheur : C'est un événement qui se produit en dehors de la frontière du logiciel mesuré et initie un ou plusieurs processus fonctionnels. Dans un ensemble de

Fonctionnalités Utilisateurs Requises (FUR), chaque type d'événement qui déclenche un processus fonctionnel est indivisible pour cet ensemble de FUR.

Remarque 1 : Les horloges et les événements temporels peuvent être des événements déclencheurs.

Remarque 2 : Un événement est survenu dans le passé ou pas ; il est instantané.

Expert : C'est la personne qui utilise le prototype pour entrer les différentes informations dans la base de connaissances (mots-clés, concepts topologiques, cas problèmes, thèmes avec réponses et recommandations).

Exigences de l'utilisateur : Dans COSMIC-FFP les exigences de l'utilisateur s'expriment à travers les FUR. Ainsi, les exigences de l'utilisateur peuvent être représentées par des spécifications, un design, etc.

Fait : C'est le constat du mesureur par rapport à un thème et au cas problème relié à ce thème.

Fonction de mesure (COSMIC-FFP) : C'est une fonction mathématique qui assigne une valeur numérique à sa variable sur la base de l'étalon de mesure COSMIC-FFP. La variable de la fonction de mesure de COSMIC-FFP est le mouvement de données.

Fonctionnalité utilisateur requise (FUR) : C'est un sous-ensemble de besoins de l'utilisateur. Le FUR représente les pratiques et les procédures de l'utilisateur que le logiciel doit accomplir pour répondre aux besoins de celui-ci. Le FUR exclut les besoins en matière de qualité et les besoins techniques.

Frontière : C'est une interface conceptuelle entre le logiciel étudié et ses utilisateurs.

Remarque : La frontière d'un logiciel 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 fonctionne ce logiciel.

Groupe de données (type) : C'est un ensemble non vide, non ordonné et non redondant d'attributs. Chaque attribut décrit un aspect complémentaire du même objet d'intérêt (voir définition).

Groupe de données persistant : C'est une qualité indiquant la persistance d'un groupe de données dans le contexte du FUR. Trois types de persistance sont définis : transitoire (pour la durée de vie du processus seulement), courte (au-delà de la durée du processus fonctionnel tant que le logiciel est opérationnel) et durable (au-delà de la durée des opérations du logiciel).

Inférences : Elles décrivent les étapes du raisonnement dans le processus de résolution de problèmes. Les inférences sont aussi appelées des mécanismes d'inférence. Les modèles d'inférence peuvent être aussi appelés "structure d'inférence".

Information : Un signifié transporté par une donnée.

Lecture (type) : C'est un type de mouvement de données qui place un groupe de données de la partie de stockage à la portée de la fonctionnalité qui le requiert.

Remarque : Une lecture inclut certaines manipulations de données associées nécessaires à son accomplissement.

Logiciel : C'est un ensemble d'instructions, de données, de procédures et éventuellement de documentations opérées comme un tout, pour répondre à un ensemble de buts spécifiques, lesquels peuvent être décrits dans une perspective fonctionnelle via un ensemble fini de requis fonctionnels (FUR), techniques et qualitatifs.

Mesureur : C'est la personne qui utilise le prototype pour interroger la base de connaissances en vue d'obtenir une recommandation (solution).

Méthode de mesure : C'est une séquence logique d'opérations, décrite de façon générique, utilisée pour réaliser la mesure.

Méthode de mesure de la taille fonctionnelle : C'est une implantation spécifique de la MTF défini par un ensemble de règles, lesquelles sont conformes aux caractéristiques obligatoires de la partie 1 de la norme ISO/IEC 14143.

Modèle : C'est une description ou une analogie utilisée pour aider à visualiser un concept qui ne peut être visualisé directement.

Mouvement de données (COSMIC-FFP) : C'est un mouvement élémentaire et fonctionnel de données rencontré pendant l'exécution d'un processus fonctionnel. Il y a quatre types de mouvements élémentaires de données : entrée, sortie, lecture et écriture. Un mouvement élémentaire de données réfère à un ensemble d'attributs trouvés dans un groupe de données. Un mouvement de données COSMIC-FFP est l'équivalent d'un CBF ISO. Un mouvement de données COSMIC-FFP exprime seulement la Fonctionnalité Utilisateur Requise (FUR) et exclut les requis de qualité et les requis techniques.

Ontologie : Elle représente la description de la structure et du vocabulaire du domaine de la connaissance statique.

Principes{ XE "Principes" } de la méthode de mesure COSMIC-FFP{ XE "COSMIC-FFP" } : Les principes{ XE "principes" } de la méthode de mesure COSMIC-FFP font partie des règles{ XE "règles" } de la méthode de mesure pour notre système à base de connaissances. Les principes de la méthode de mesure indiquent les contraintes qui régissent l'identification d'un concept COSMIC-FFP.

Principes de la méthode de mesure COSMIC-FFP : Les principes de la méthode de mesure COSMIC-FFP sont les règles de la méthode de mesure pour notre système de connaissances. Ils indiquent les contraintes qui régissent l'identification d'un concept COSMIC-FFP. Dans le contexte du système à base de connaissances nous allons utiliser l'expression "règles de la méthode de mesure" plutôt que "principes de la méthode de mesure". Lorsque nous faisons référence au Manuel de la mesure COSMIC-FFP [2], nous utilisons le mot "principe".

Procédure de mesure : C'est un ensemble d'opérations, décrites de façon spécifique, utilisées pour réaliser une mesure particulière selon une méthode donnée.

Processus de mesure : C'est un processus qui sert à établir, à planifier, à réaliser et à évaluer la mesure du logiciel.

Processus fonctionnel (type). Synonyme de type de transaction : C'est un composant élémentaire d'un ensemble de besoins fonctionnels de l'utilisateur comprenant un ensemble de mouvements de données unique, cohésif et indépendamment exécutable. Il est déclenché par un ou plusieurs événements déclencheurs soit directement ou indirectement, via un "acteur". Il est complet lorsqu'il a exécuté tout ce qui est requis en réponse à un déclencheur.

Règle : Dans le cadre d'un système à base de connaissances, c'est ce qui permet de manipuler les faits.

Règles d'assignation numérique : Dans le Manuel de mesure COSMIC-FFP, c'est la phase qui consiste à assigner une valeur numérique (toujours 1) à chaque mouvement de données.

Règles{ XE "Règles" } de la méthode de mesure COSMIC-FFP { XE "COSMIC-FFP" } : Dans le Manuel de mesure COSMIC-FFP [2], les règles{ XE "règles" } permettent de confirmer le statut d'un concept COSMIC-FFP mais ne constituent pas des contraintes (voir principes{ XE "principes" } de la méthode de mesure COSMIC-FFP).

Règles de la méthode de mesure: Dans le cadre de notre thèse, c'est l'ensemble des définitions, des principes et des règles de la méthode de mesure COSMIC-FFP qui permettent d'identifier les concepts COSMIC-FFP.

Règles locales : Une interprétation des règles de la méthode de la mesure dans le contexte particulier d'une organisation. Une règle locale n'est pas une nouvelle règle de la méthode de mesure, mais l'interprétation d'une règle existante (ex. : une organisation utilise la conception orientée objet et interprète le concept de "classe" comme l'équivalent du concept groupe de données dans la méthode de mesure fonctionnelle COSMIC-FFP).

Sortie (type) : C'est un type de mouvement de données qui déplace un groupe de données d'un processus fonctionnel à travers la frontière vers l'utilisateur qui le demande.

Remarque : Une sortie inclut aussi les manipulations de données associées (ex. : formatage et routage associés à la donnée afin d'être sortie).

Système à base de connaissances : Synonyme de système expert.

Système expert : Un programme d'ordinateur qui représente et raisonne avec la connaissance d'un domaine de spécialité en vue de résoudre un problème ou de fournir un avis.

Tâches : Les tâches sont les activités qui doivent être réalisées pour la résolution d'un problème.

Taille fonctionnelle : C'est la taille d'un logiciel dérivée par la quantification des Fonctionnalités Utilisateurs Requises (FUR).

Thèmes : Dans cette thèse, ce sont les questions que se pose le mesureur pour trouver une solution aux cas problèmes.

UML : Acronyme de Unified Modeling Language, aujourd'hui le plus souvent indissociable de la conception objet puisque c'est le résultat de la fusion de plusieurs méthodes de conception objet.

Unité (de mesure) : C'est une grandeur particulière, définie et adoptée par convention, à laquelle on compare les autres grandeurs de même nature pour les exprimer quantitativement par rapport à cette grandeur.

Utilisateur : C'est une personne qui spécifie des FUR et/ou "choses" (*thing*) qui communique ou interagit avec le logiciel à tout moment.

Valeur (d'une grandeur)¹ : C'est une expression quantitative d'une grandeur particulière, généralement sous la forme d'un étalon de mesure multipliée par un nombre.

1 Adapté de *Merriam Webster's Collegiate Dictionary*, 10th édition et *Le Petit Larousse illustré*, 1996.

INTRODUCTION

Cette thèse porte sur l'application de la mesure fonctionnelle ISO 19761 : 2002 (COSMIC-FFP) [3, 4]. Appliquer une mesure fonctionnelle à un logiciel veut dire mettre en correspondance les artefacts du logiciel à mesurer avec les règles² de la méthode de mesure. Cette tâche est complexe et demande au mesureur une expertise qu'il acquiert, via des connaissances en génie logiciel et en mesure ainsi que par la pratique de la mesure.

Cette thèse décrit, entre autres, la démarche du mesureur, ce qui n'a jamais été fait à notre connaissance dans d'autres travaux de recherche et n'a jamais, de plus, été analysée systématiquement et encore moins traitée dans le cadre d'une approche cognitive. À ce titre, c'est une recherche de type exploratoire. Pour appliquer une mesure fonctionnelle, le mesureur utilise ses connaissances (en mesure fonctionnelle, mais aussi en génie logiciel) ainsi que des informations sur les documents ou les artefacts décrivant le logiciel. Cette démarche est difficilement automatisable puisque les sources d'information ont un formalisme très varié en plus de nécessiter, assez souvent, des interviews avec le personnel de développement ou de production du logiciel. Nous faisons l'hypothèse qu'un système à base de connaissances pourrait aider le personnel de la mesure à acquérir et à maintenir les connaissances nécessaires à la mesure fonctionnelle des logiciels.

Le Chapitre 1 de notre document met en contexte la mesure fonctionnelle et les premières réflexions qui nous ont conduit à cette recherche. Le Chapitre 2 présente une revue de littérature de la mesure du logiciel et, plus particulièrement de la mesure de la taille fonctionnelle du logiciel et son application. Dans ce même chapitre, nous présentons aussi une revue de littérature sur les approches cognitives en lien avec notre sujet. Les autres chapitres traitent respectivement :

- de la méthodologie de recherche (Chapitre 3) ;

² Dans le Guide COSMIC-FFP le terme "principe" est utilisé plutôt que le terme "règle".

- de la formalisation de la démarche du mesureur (Chapitre 4) ;
- des prototypes et des expérimentations (Chapitres 5 et 6).

Pour conclure, nous présentons un sommaire des résultats, la bibliographie et les Annexes. Les annexes ont pour but d'illustrer plus en détail certaines des explications que nous fournissons dans le texte principal (ex. : règles{ XE "règles" } locales, description des différents outils et prototypes). L'Annexe A décrit sommairement le logiciel Help!CPR qui nous a inspiré quand au type d'outil à développer pour la construction de notre système à base de connaissances{ XE "base de connaissances" }. L'Annexe B décrit le premier prototype utilisé pour la base de connaissances. L'Annexe C décrit l'environnement de développement du deuxième prototype. L'Annexe D décrit la procédure de vérification de la base de connaissances du prototype 2. L'Annexe E fournit le curriculum des experts qui ont examiné la base de connaissances du prototype 1. L'Annexe F fournit la liste des associations qui font la promotion de la mesure fonctionnelle à travers le monde. L'Annexe G est un extrait d'un document d'information sur l'application des "règles locales". L'Annexe H fournit les liens URL ou le texte intégral des études de cas{ XE "cas" } utilisés pour construire la base de connaissances. L'Annexe I présente, en pseudo code, les formules utilisées pour le traitement des faits de la base de connaissances. L'Annexe J montre les résultats des analyses de variance et de corrélation pour les expérimentations (SPSS version 8.0). L'Annexe K fournit le détail des résultats des bonnes réponses pour tous les mesureurs avec (O) et sans outil lors de l'expérimentation. L'annexe L fournit les détails des mauvaises réponses pour quelques mesureurs lors de l'expérimentation et, finalement l'Annexe M apporte quelques exemples de spécifications présentées aux mesureurs débutants lors de l'expérimentation.

1. CONTEXTE

Les entreprises dépendent de plus en plus de l'informatique et les budgets des entreprises pour l'informatique devraient atteindre, au niveau mondial 2,460 milliards US en 2003 [5]. Si les organisations veulent pouvoir contrôler de plus en plus les dépenses en logiciels, elles doivent pouvoir évaluer et analyser ces dépenses à toutes les étapes du cycle de développement. Des modèles d'analyse avec les mesures pertinentes sont nécessaires à cette fin. À quoi servent ces mesures ? Dans le Manuel de mesure COSMIC-FFP [2] on écrit :

"Des mesures sont nécessaires pour analyser tant la qualité que la productivité du développement ou de la maintenance des logiciels. D'une part, des mesures techniques sont nécessaires pour quantifier la performance technique des produits ou des services du point de vue des développeurs. Des mesures techniques peuvent être utilisées pour des analyses d'efficacité ; pour l'amélioration de la performance du design entre autres.

D'autre part, les mesures fonctionnelles sont nécessaires pour quantifier la performance des produits ou des services du point de vue de l'utilisateur ; pour l'analyse de la productivité entre autres. Les mesures fonctionnelles doivent être indépendantes des techniques de développement et des décisions d'implantation. Elles peuvent donc être utilisées pour comparer la productivité des différentes techniques et technologies" [2, p.14].

La première méthode de mesure fonctionnelle a été introduite par Allen Albrecht à la fin des années 1970 [6]. Depuis, d'autres méthodes de mesure fonctionnelle ont été introduites dont Mark II [7] et COSMIC-FFP [2], en vue d'améliorer les règles existantes et aussi de fournir la possibilité de mesurer plus adéquatement les logiciels des domaines autres que les logiciels de gestion [8]. L'ajout de nouvelles méthodes de mesure et l'évolution des méthodes de construction du logiciel, ont créé un environnement favorable aux discussions sur la définition de la mesure fonctionnelle et la légitimité ou la pertinence des différentes méthodes de mesures proposées. En 1996, la norme ISO 14143-1 [9] définissait formellement la mesure fonctionnelle et ouvrait la voie à la normalisation de différentes méthodes de mesure

fonctionnelles. Depuis décembre 2002, la norme ISO/IEC 19761 [3, 4] est la norme pour la méthode COSMIC-FFP. La norme ISO/IEC ne comprend que les règles de base de la méthode puisqu'elle n'a qu'une douzaine de pages. Par contre, le guide d'implantation qui comprend plusieurs exemples pouvant aider le mesureur se trouve dans le Manuel de mesure COSMIC-FFP [2] et comprend plus de 80 pages.

1.1 Brève présentation de COSMIC-FFP

Pour notre recherche, la mesure fonctionnelle de référence est COSMIC-FFP. Cette brève présentation de COSMIC-FFP a pour but d'aider le lecteur à mieux comprendre la démarche de COSMIC-FFP et les relations entre les concepts COSMIC-FFP.

La méthode de mesure fonctionnelle COSMIC-FFP vise à mesurer la fonctionnalité de tous les types de logiciels, que ce soit les logiciels temps réel, embarqués ou d'affaires [8]. Un logiciel peut être vu sous différents angles. Dans la perspective de la méthode de mesure COSMIC-FFP, la vision est celle des fonctionnalités livrées aux utilisateurs. La fonctionnalité livrée aux utilisateurs est décrite via les Fonctionnalités Utilisateurs Requises (FUR).

La méthode de mesure comprend deux phases : la phase du processus de mise en correspondance du logiciel mesuré avec la méthode de mesure (Figure 1-1) et la phase de mesure (Figure 1-2).

Le modèle du processus de mesure COSMIC-FFP montre que, préalablement à l'application des règles et des procédures de mesure, le logiciel doit être mis en correspondance avec un modèle de logiciel propre à COSMIC-FFP. Les Fonctionnalités Utilisateurs Requises (FUR) capturent les définitions, les relations et les concepts (structure fonctionnelle) requis pour compléter l'exercice de la mesure de la taille fonctionnelle. La Figure 1-1 montre la méthode générale du processus de mise en correspondance COSMIC-FFP. Les détails des définitions,

des principes{ XE "principes" } et des règles³ de mise en correspondance du logiciel sur le modèle de logiciel de COSMIC-FFP, sont présentés à la section 3 du manuel de mesure [2, p. 31 ss.].

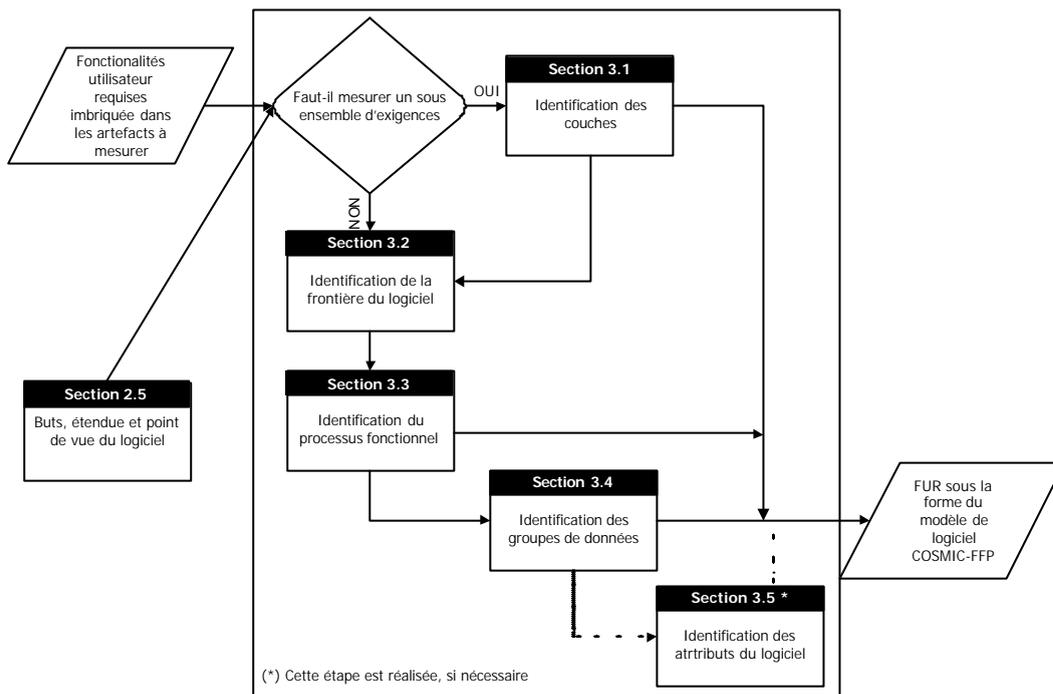


Figure 1-1 : Méthode générale du processus de mise en correspondance COSMIC-FFP

La première étape de la méthode générale du processus de mise en correspondance (2.5) consiste, pour le mesureur, à trouver les FUR à partir des artefacts à mesurer. À cette fin, il se sert des définitions, des principes et des règles tant pour identifier les "couches" du logiciel, la frontière du logiciel, ainsi que les groupes de données et leurs attributs (3.1 à 3.5, Figure 1-1). Le mesureur obtient alors les FUR sous la forme du modèle de logiciel COSMIC-FFP.

³ Il s'agit ici des règles de la méthode de mesure COSMIC-FFP.

Lorsque le logiciel est mis en correspondance avec le modèle de logiciel COSMIC-FFP, la mesure est réalisée en appliquant un ensemble de principes (au sens du Manuel de mesure COSMIC-FFP) et de règles sur le modèle. On obtient ainsi un nombre représentant la taille fonctionnelle du modèle de logiciel. Par convention, ce nombre est ensuite utilisé pour représenter la taille fonctionnelle du logiciel lui-même. La Figure 1-2 montre la procédure générale de la phase de mesure de COSMIC-FFP.

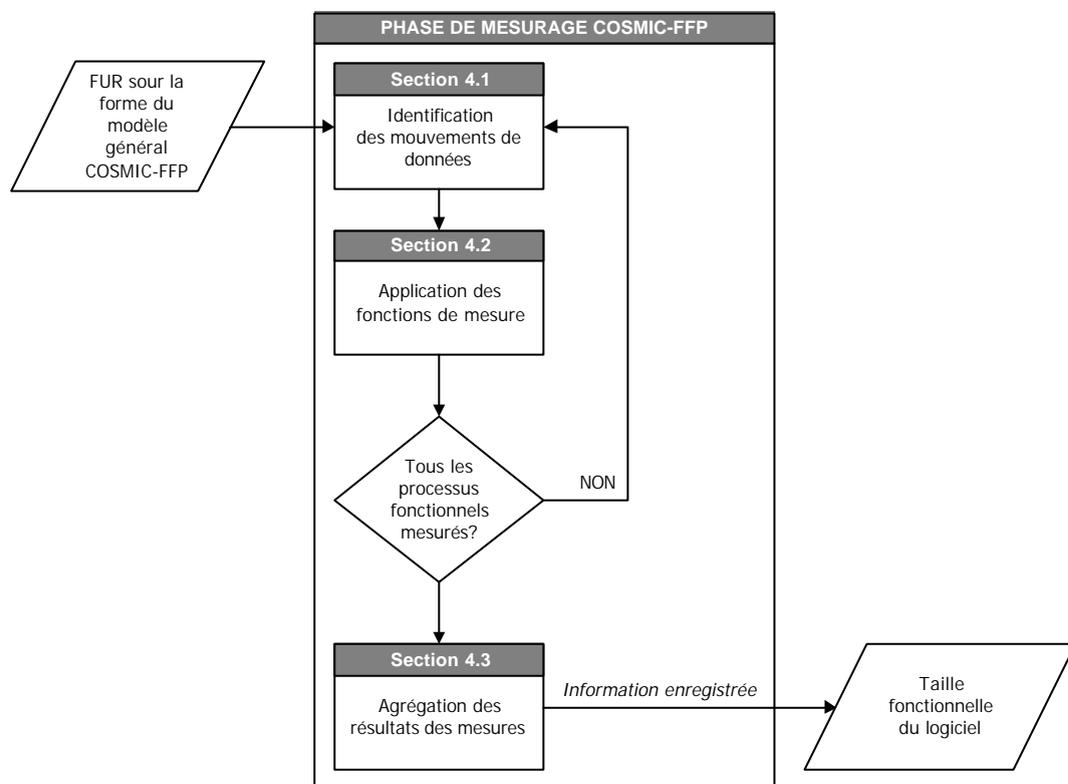


Figure 1-2 : Procédure générale de la phase de mesure COSMIC-FFP

Dans cette phase, il existe aussi des définitions, des principes{ XE "principes" } et des règles (voir note 2){ XE "règles" } nécessaires à l'identification des mouvements de données d'un processus (2.5), ainsi qu'une procédure pour l'application et l'agrégation des fonctions et des résultats des mesures. L'information enregistrée est alors la taille fonctionnelle du logiciel.

Les détails des définitions, des principes et des règles (voir note 2) pour mesurer le modèle de logiciel de COSMIC-FFP{ XE "COSMIC-FFP" }, sont présentés à la section 4 du manuel [2, p.40 ss].

1.2 Les concepts de COSMIC-FFP

Les concepts COSMIC-FFP ont entre eux des relations qui sont schématisés à la Figure 1-3 :

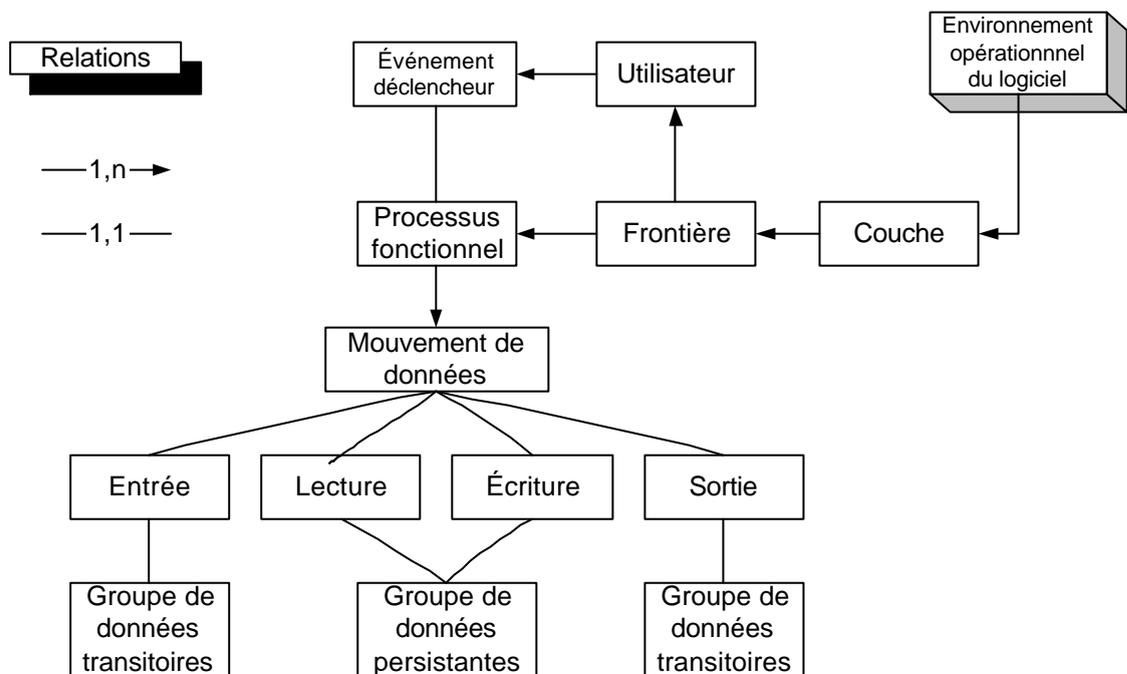


Figure 1-3 : Concepts de COSMIC-FFP [10]

Il y a plusieurs niveaux de concepts dans la méthode COSMIC-FFP : à partir de l'environnement opérationnel jusqu'aux groupes de données en passant par les processus fonctionnels et les mouvements de données.

L'environnement opérationnel du logiciel est une collection de logiciels exécutés de façon concurrente sur un ordinateur spécifique. Un environnement logiciel peut avoir plusieurs couches de logiciels qui interagissent les unes avec les autres via leurs processus fonctionnels respectifs. La couche est donc le résultat du partitionnement fonctionnel de l'environnement du logiciel.

Un logiciel possède des "éléments" qui répondent aux besoins des utilisateurs et qui se traduisent dans des processus fonctionnels. Dans le cadre de la mesure fonctionnelle COSMIC-FFP, l'utilisateur est n'importe quelle personne qui spécifie des FUR et/ou n'importe quelle personne ou chose qui communique ou interagit avec le logiciel à n'importe quel moment.

Le processus fonctionnel est un composant élémentaire d'un ensemble de besoins fonctionnels de l'utilisateur comprenant un ensemble de mouvements de données unique, cohésif et indépendamment exécutable. Il est déclenché par un ou par plusieurs événements déclencheurs soit directement ou indirectement par l'utilisateur.

Les mouvements de données sont des entrées, des sorties, des lectures et des écritures. Une sortie est un type de mouvement de données qui déplace un groupe de données d'un processus fonctionnel à travers la frontière vers l'utilisateur qui le demande, alors que l'entrée déplace un groupe de données depuis un utilisateur à travers la frontière vers le processus fonctionnel où il est requis. Les groupes de données peuvent être transitoires. Une lecture est un type de mouvement de données qui place un groupe de données de la partie de stockage à la portée de la fonctionnalité qui le requiert. Une écriture est un type de mouvement de données qui déplace un groupe de données se trouvant à l'intérieur d'un processus fonctionnel vers une partie de stockage. Les données d'écriture doivent être persistantes. Les données sont persistantes lorsqu'elles persistent au-delà de la durée du processus fonctionnel ou au-delà de la durée des opérations du logiciel.

1.3 Origine de nos recherches

Dans notre pratique des 15 dernières années avec les mesures fonctionnelles, nous avons constaté que l'application de la méthode de mesure n'était pas simple et demandait un apprentissage et une expérience, tant de la méthode que de l'environnement d'application de la méthode.

La compréhension de ces différents concepts et les moyens de les appliquer pour arriver à les mesurer, font, de la mesure fonctionnelle, une tâche complexe et difficilement automatisable [11]. En effet, dans la phase de mise en correspondance, le mesureur doit identifier les couches de l'application à mesurer, la frontière de chaque logiciel, les différents processus fonctionnels du logiciel et enfin les groupes de données. Cette mise en correspondance demande au mesureur de porter un jugement sur un grand nombre de variables qui ne sont pas toujours bien définies. De plus, au niveau de la phase de mesure, il doit identifier les mouvements de données appartenant à chaque processus et appliquer les fonctions de mesure, ce qui n'est pas une tâche triviale. Ceci est particulièrement vrai pour les mesureurs ayant une expérience limitée dans ce type de mesure, ou encore des mesureurs ayant une compréhension limitée du logiciel à mesurer.

Notre intuition au début de ce projet nous a conduit à regarder des approches qui pourraient nous permettre de transférer nos connaissances et notre pratique des règles de la méthode de mesure{ XE "règles" } aux nouveaux mesureurs. Depuis 1988, nous essayons de trouver des moyens d'aider le mesureur débutant à appliquer les règles de la mesure fonctionnelle. À cette époque, une expérience avait été réalisée avec le logiciel VP Expert [12]. En 1995, nous avons proposé et construit, en collaboration avec des chercheurs allemands du GMD, un prototype avec une structure hypertexte permettant aux mesureurs de mieux comprendre le modèle des points de fonction IFPUG [13]. La carte conceptuelle de cette méthode est représentée dans la Figure 1-4 [13, p. 261] :

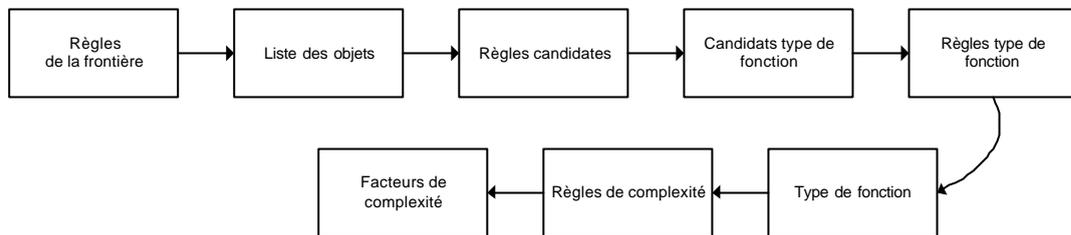


Figure 1-4 : Carte conceptuelle – méthode IFPUG [13]

La carte conceptuelle de la Figure 1-4 met en relation les différents concepts de la méthode de mesure fonctionnelle utilisée en 1995. On retrouve les concepts de frontière, d'objets (ou artefact) et de fonction (ou processus), concepts que l'on retrouve dans cette méthode de mesure et COSMIC-FFP (Figure 1-3). Les types de fonction, les règles candidates{ XE "règles" } et les facteurs de complexité se retrouveraient dans la procédure générale de la phase de mesure si nous l'appliquions à cette méthode de mesure.

L'outil hypertexte a des propriétés intéressantes permettant de lier les différents concepts entre eux sous forme graphique et d'indiquer les étapes de la méthode de mesure, tout comme on les retrouve dans les sections 3 et 4 du Manuel de mesure COSMIC-FFP. Ainsi, le concept d'hypertexte permet de rendre disponibles des informations dans un document sous forme graphique et textuelle. En "cliquant" sur l'un des concepts de la carte conceptuelle on peut retrouver les informations relatives à ce concept. L'outil ne possède pas, par contre, les caractéristiques des systèmes à base de connaissances (ex. : un moteur d'inférence) et nous limite à une présentation de l'information à différents niveaux.

Il faut noter aussi que cette carte conceptuelle présente l'ordre d'apparition des différents concepts lors de la mesure fonctionnelle, mais n'indique pas nécessairement le cheminement de la pensée du mesureur expert lorsqu'il essaie d'identifier les concepts par rapport au logiciel à mesurer.

Lorsque nous avons commencé notre recherche doctorale, la question qui se posait était la suivante : comment le mesureur{ XE "mesureur" } expert en arrive-t-il à reconnaître les fonctionnalités d'un logiciel du point de vue de la mesure et à appliquer les règles{ XE

"règles" } de la méthode de mesure fonctionnelle ? Pour nous, ce que le mesureur devait faire, c'était un diagnostic au même titre qu'un médecin pose un diagnostic à partir des symptômes de la maladie de son patient. Ici, le diagnostic se fait à partir de l'analyse des artefacts du logiciel.

Notre première étape de recherche a consisté à modéliser, dès l'automne 2000, la démarche du mesureur [11], démarche que nous explicitons en détails dans le Chapitre 4 de notre thèse. Nous avons voulu ensuite vérifier si cette démarche pouvait être concrétisée à travers un outil de diagnostic. La même année, l'outil retenu fut Help!CPR de la firme Haley Enterprise. Son interface utilisateur, relativement facile à utiliser, permet le diagnostic. La description de ce logiciel se trouve à l'Annexe A. Sur la base de la modélisation du cheminement du mesureur, nous avons expérimenté le modèle en y incorporant quelques cas de diagnostics typiques au mesureur.

Ce que nous avons appris à partir de Help!CPR se résume de la façon suivante. Avec un logiciel de diagnostic tel que Help!CPR il est possible :

- d'utiliser un index pour retrouver les cas ;
- de créer des cas et de les gérer avec des règles ;
- d'utiliser les formules mathématiques sur la base de la théorie de la certitude et de les intégrer au logiciel ;
- et de générer des recommandations à partir d'un certain nombre de faits.

Nous avons constaté que ce modèle ne permet pas de faire un lien direct entre les différents concepts de la mesure et les cas à résoudre. Cette compréhension des différents concepts de la mesure fait partie de la connaissance du mesureur lorsqu'il applique les règles{ XE "règles" } de la méthode de mesure (voir chapitre 4). Puisque nous n'avions pas le contrôle des sources du logiciel, il ne nous était pas possible de faire des changements au logiciel pour répondre à nos besoins.

2. REVUE DE LITTÉRATURE ET PROBLÉMATIQUE

Ce chapitre présente la revue de littérature et la problématique de notre recherche. Une première section traitera de la problématique du domaine de la mesure en génie logiciel, et plus particulièrement de la mesure fonctionnelle. Dans la deuxième section, nous traiterons de la problématique du domaine cognitif et plus particulièrement des systèmes de connaissances à travers les techniques de diagnostic. La Figure 2-1 schématise, à un haut niveau, la démarche de la revue de littérature.

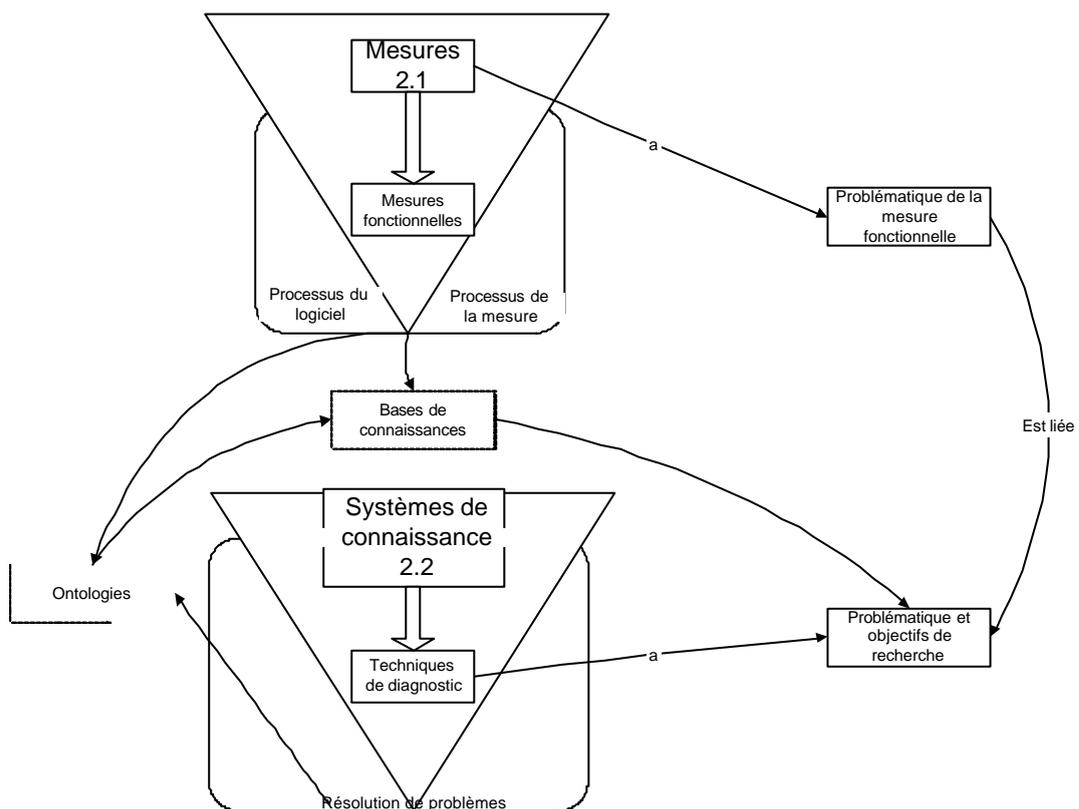


Figure 2-1 : Démarche de la revue de littérature

Notre revue de littérature couvre le domaine de la mesure en génie logiciel et une partie du domaine cognitif à partir des systèmes à base de connaissances. Pour les deux domaines, notre approche de revue de littérature part du générique au particulier c'est-à-dire des mesures en général aux mesures fonctionnelles, des systèmes à base de connaissances aux techniques de diagnostic.

La Figure 2-1 montre les environnements respectifs de la mesure fonctionnelle et des techniques de diagnostic. La mesure fonctionnelle fait partie de l'environnement du processus du logiciel (cycle de vie du logiciel) et du processus de la mesure (application de la mesure). Les techniques de diagnostic sont dans l'environnement de la résolution de problèmes.

2.1 Domaine de la mesure en génie logiciel

La mesure en génie logiciel (plus particulièrement la mesure fonctionnelle) a fait l'objet de plusieurs débats dans les 20 dernières années, à commencer par la définition de l'unité de mesure, des classes de mesure et du schème de classification des mesures. La Figure 2-2 donne une vue d'ensemble de notre démarche de revue de littérature pour la mesure.

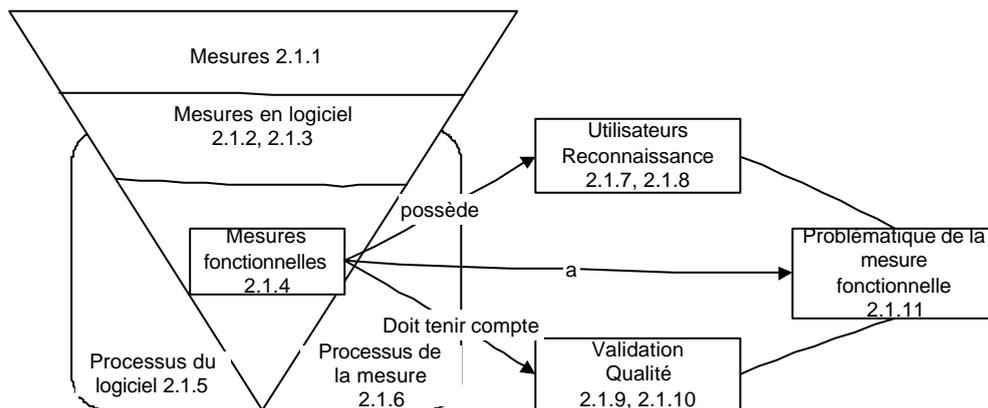


Figure 2-2 : Démarche de la revue de littérature pour la mesure

Dans un premier temps (voir Figure 2-2), la mesure fonctionnelle se présente du point de vue des utilisateurs de la mesure et de la reconnaissance de ce type de mesure dans l'industrie. Dans un deuxième temps, nous traitons de la question de la validation des mesures et de la qualité de la documentation pour générer les mesures fonctionnelles. Ces différents points de vue nous amènent à la problématique de l'application de la mesure fonctionnelle ou du mesurage.

2.1.1 Définition de l'unité de mesure

ISO définit l'unité de mesure comme une "grandeur particulière, définie et adoptée par convention, à laquelle on compare les autres grandeurs de même nature pour les exprimer quantitativement par rapport à cette grandeur" [14].

Dans cette définition, on indique qu'une grandeur particulière est définie et adoptée par convention. Ceci implique que dans toute activité de mesure, il y aura des règles (conventions) à suivre. Fenton écrit [15] :

In any measurement activity, there are rules to be followed. The rules help us to be consistent in our measurement, as well as providing a basis for interpreting data. Measurement theory tells us the rules, laying the groundwork for developing and reasoning about all kinds of measurement.

Ceci n'est pas unique au seul domaine du génie logiciel selon Fenton [15] :

This rule-based approach is common in many sciences.

Dans cette définition, il est aussi question de comparer des grandeurs de même nature. Pour Fenton, il s'agit d'une relation empirique obtenue par un consensus lorsqu'elle s'applique au monde réel :

When the two people being compared are very close in height, we may find a difference of opinion ; you may think that Jack is taller than Jill, while we are convinced that Jill is taller than Jack. Our empirical relations permit this difference by requiring only a consensus of opinion about relationships in the real world. A (binary) empirical relation is one for which there is a reasonable consensus about which pairs are in the relation [15] .

Le dernier élément de cette définition suggère que l'unité de mesure s'exprime "quantitativement". Pour ce faire, il faut partir des conventions et appliquer celles-ci à l'objet que l'on veut mesurer. Le mesurage (en anglais *measurement*) est l'activité qui permet d'exprimer quantitativement l'unité de mesure. Plus formellement, ISO définit le mesurage de la façon suivante :

Un ensemble d'opérations ayant pour but de déterminer la valeur d'une grandeur [14].

Comment se déroulent ces opérations ? Fenton écrit "que la mesure est une mise en correspondance entre le monde empirique et le monde formel" [16].

Le monde empirique de la taille du logiciel, selon Fenton, est composé d'artefacts, de livrables et de documents qui résultent du processus de construction du produit et qui proviennent des spécifications, des designs, des codifications et des tests.

Le monde formel est composé des conventions ou des règles de mesure du logiciel à travers ses artefacts, ses livrables et ses documents. C'est à partir de là que l'on peut définir la mesure de la taille du logiciel.

2.1.2 Les classes de mesure du logiciel

Il existe plusieurs classes de mesure du logiciel selon Zuse. Il en a identifié quelques-unes [17 , p. 17] :

- le schème de classification de Fenton (sur la base de la mesure des ressources, des processus et des produits) ;
- la classification dérivée des modèles de logiciel (*flowgraph*, *call graph*, *source code* ou *LOC*) ;
- les mesures de logiciel intra et inter modulaires. La mesure intra modulaire mesure, par exemple, les programmes, les graphes et les modules. La mesure extra modulaire mesure l'ensemble du logiciel ;
- les mesures de taille et de "complexité" du logiciel. Voici quelques exemples : les mesures de taille du logiciel en lignes de code, les mesures de complexité cyclomatiques de McCabe, les mesures sur les flux d'information d'Henry, les mesures de volume d'Halstead ;
- la classification des mesures des programmes orientés objet.

Notre intérêt porte plus particulièrement sur le schème de classification de Fenton [15] déjà cité dans notre définition de l'unité de mesure.

2.1.3 Schème de classification des mesures

Le schème de classification des mesures de Fenton est intéressant parce qu'il a établi son cadre de référence de mesure du logiciel en rapport avec les buts de la mesure du logiciel. Le cadre de référence est construit en trois étapes : classifier les entités qu'il faut étudier, déterminer les buts pertinents de la mesure et identifier le niveau de maturité que l'organisation a atteint en utilisant ces mesures.

Il y a trois classes d'entités à mesurer selon Fenton [15, Chapitre 3] :

- les processus ou les collections d'activités liés au développement du logiciel ;
- les produits ou les artefacts, les livrables ou les documents qui résultent du processus ;
- les ressources ou les entités requises par un processus.

Pour les fins de notre recherche, nous considérons uniquement les mesures de produits qui résultent du processus. De plus, dans les mesures de produits, nous sommes intéressés uniquement à une seule classe d'attributs, soit la mesure de la taille fonctionnelle du logiciel.

2.1.4 La mesure fonctionnelle

Dans cette sous-section, nous traiterons d'abord de la mesure de la taille du logiciel, puis de la mesure fonctionnelle.

2.1.4.1 Définition de la mesure de la taille du logiciel

En se basant sur les classes d'entités de Fenton, on constate que la mesure de la taille du logiciel est une mesure de produit qui peut provenir de différentes "entités" du logiciel. Selon Fenton, les "entités" des produits sont : les spécifications, les designs, les codifications et les tests. Le choix des "entités" à mesurer détermine la méthode de mesure.

Pour cette raison, le terme "mesure de la taille" est aussi utilisé dans un contexte autre que la mesure de la taille fonctionnelle. C'est le cas pour la mesure du volume d'un logiciel qui est aussi une mesure de la taille du logiciel d'après Zuse [17, p. 20]. Halstead [18] a introduit cette mesure en suggérant un calcul à partir des opérateurs et des opérandes. La mesure de la codification du logiciel se traduit par la mesure des lignes de code. Boehm [19] a fait la mesure de la taille de ses 63 projets en lignes de code (LOC). Il a fourni une brève description de ce qu'est une ligne de code, description reprise par Conte et al. [20], d'une façon plus élaborée, cinq ans plus tard. Les mesures de la taille du logiciel en lignes de code

ou via les opérateurs et les opérands, ont été beaucoup critiquées en raison de leur dépendance à la technique de programmation utilisée.

Les mesures indépendantes des techniques de programmation se basent sur les spécifications et le design. La première mesure indépendante des langages de programmation, reconnue internationalement, est la méthode des points de fonction d'Allen Albrecht [21]. L'analyse du processus de mesure des points de fonction d'Albrecht a fait l'objet d'une thèse de doctorat par le D' Alain Abran [21] au début des années 1990.

Tel que déjà mentionné, au milieu des années 1990, la norme ISO 14143-1 a permis de normaliser la définition d'une méthode de mesure fonctionnelle. Cette norme répondait au besoin de l'industrie, soit d'avoir une définition normalisée de la mesure fonctionnelle. Pour ISO 14143-1, la définition de mesure fonctionnelle est la suivante :

A size of the software derived by quantifying the Functional User Requirements [9].

2.1.4.2 Dimensions du logiciel et méthodes de mesure fonctionnelle

Tom DeMarco [22] suggère que tous les logiciels tentent de solutionner des problèmes qui ont trois dimensions :

- les données et l'interface utilisateur ;
- les processus internes ou fonction ;
- le comportement dynamique ou contrôle.

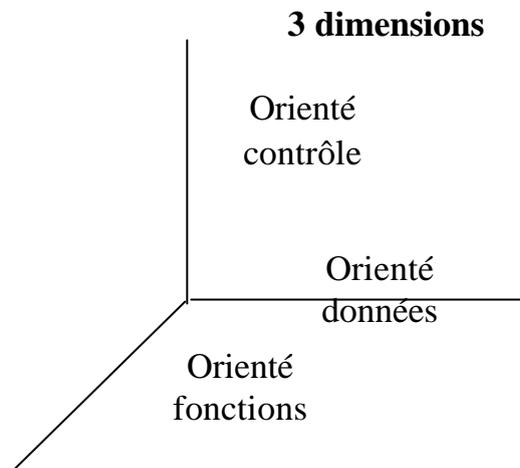


Figure 2-3 : Les trois dimensions du logiciel [22]

À la Figure 2-3, on retrouve des orientations qui exigent des solutions plus liées aux données et aux interfaces tels les "problèmes" de comptabilité et d'inventaire (logiciels de gestion). D'autres sont orientées contrôles et fonctions et exigent des solutions plus axées sur les processus internes et le comportement dynamique des processus (logiciels en temps réel et d'infrastructure).

La mesure fonctionnelle d'Albrecht [23] a été construite essentiellement à partir d'une vingtaine de logiciels d'affaires. Ces logiciels solutionnent des problèmes dont la principale dimension vise les données et l'interface utilisateur. La mesure fonctionnelle d'Albrecht a été reprise et définie de façon plus précise par un groupe d'utilisateurs de différents pays (IFPUG) [24].

La méthode de mesure fonctionnelle COSMIC-FFP a été créée dans le but de combler les besoins d'une mesure fonctionnelle de taille pour les logiciels en temps réel. Pour plus d'information sur les critiques de la méthode IFPUG et sa comparaison avec COSMIC-FFP, voir les articles de Desharnais et Morris [8, 25]. Cette méthode a été présentée à un comité de normes ISO et a été acceptée comme norme ISO en décembre 2002 [3, 4].

D'autres méthodes de mesure fonctionnelle sont utilisées par les entreprises, dont MARK II de Charles Symons [7], utilisée surtout en Angleterre. Cette méthode, tout comme celle d'Albrecht, est orientée données. Elle est plus explicite en ce qui concerne l'interface utilisateur. Symons a critiqué la méthode des points de fonction dans son article publié en 1988 dans *IEEE Transactions of Software Engineering* [26].

- "la classification des composantes en simples, moyennes et complexes ne reflète pas toute la complexité des travaux nécessaires pour réaliser les systèmes que l'utilisateur utilisera ;
- le choix des poids (points par type de composante), bien que justifié et discuté, n'a pas été validé dans des environnements autres que ceux d'IBM ;
- toutes les mesures sont liées à la complexité des processus internes si l'on considère que les fichiers logiques sont liés à la structure physique des fichiers (ex. : IDMS) ;
- on tient aussi compte de la complexité dans les 14 questions relatives aux facteurs d'envergure, ce qui ajoute une certaine confusion à la notion de complexité ;
- le découpage des projets joue un rôle important dans le total des points ;
- le nombre des facteurs est limitatif. De plus, certains se chevauchent (ex. : entrées des données en direct et télécommunications) et d'autres demandent plus de précision (ex. : convivialité, facilité d'utilisation)". Voir aussi [27, p. 14].

2.1.5 Processus de développement et mesure fonctionnelle

La mesure est "une mise en correspondance entre le monde empirique et le monde formel" [15]. Le monde empirique, pour notre projet, est décrit dans les méthodologies de construction du logiciel. Chaque méthodologie décrit les artefacts qui doivent être produits pour documenter et pour construire le logiciel. Le mesureur doit reconnaître ces artefacts puisque ce sont ces artefacts qu'il utilisera pour construire le modèle de logiciel COSMIC-FFP. La reconnaissance des artefacts des différentes méthodologies de développement fait

partie des connaissances nécessaires du mesureur. Ces méthodologies se basent sur un modèle de développement.

Le modèle de processus de développement décrit par Boehm [19] est le plus cité dans la littérature. C'est le modèle de développement en cascade (*Waterfall Model*) qui compte huit phases dans sa version simple. Il reprend les mêmes phases pour la version incrémentale, mais elles peuvent se répéter plus d'une fois. La Figure 2-4 présente les phases sur la base de la version simple. Ces phases sont la faisabilité, les plans et les exigences, l'architecture du produit, l'architecture détaillée, la codification, l'intégration, l'implantation, les activités opérationnelles et la maintenance. Pour chacune des phases, Boehm suggère des activités de validation et de vérification. La validation se trouve dans la définition du produit pour les utilisateurs (construire le bon logiciel), alors que la vérification se trouve dans la construction du logiciel (construire un bon logiciel). La validation recommence au niveau des opérations et de la maintenance.

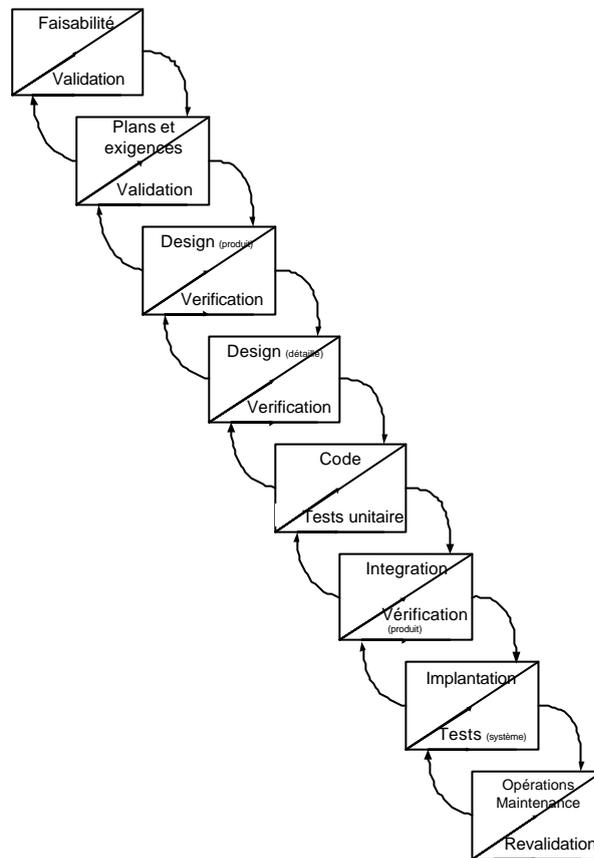


Figure 2-4 : Modèle en cascade [19]

Le modèle de la Figure 2-4 a été repris autant par les chercheurs universitaires que par les entreprises ayant à construire un logiciel.

Plus récemment, des chercheurs de l'UQÀM et de l'ÉTS, en collaboration avec IEEE et avec plusieurs entreprises, ont défini un corpus de connaissances en génie logiciel et proposé cinq (5) phases pour le cycle de vie du logiciel (Requis, Design détaillé, Code, Test, Maintenance) [28]. Ils ont aussi défini cinq (5) phases transversales, c'est-à-dire des phases qui servent à toutes les phases du cycle de vie du logiciel (Gestion de configuration, Assurance qualité, Vérification et qualité, Amélioration des processus, Gestion des processus). Le vocabulaire

que nous avons utilisé pour désigner les différentes étapes du processus de développement s'est inspiré du corpus de connaissances SWEBOK.

Il faut cependant retenir que les modèles et les méthodes de développement et de maintenance sont nombreux et diffèrent d'une organisation à l'autre. Les concepts utilisés par les différents modèles et les méthodes peuvent être définis différemment.

2.1.6 Le processus de mesure et la taille du logiciel

Le processus de mesure a été formalisé par Jacquet et Abran dans un article publié en 1998 [29]. Selon cet article, il y a quatre phases⁴ dans le processus de mesure. Ces phases sont présentées à la Figure 2-5 :

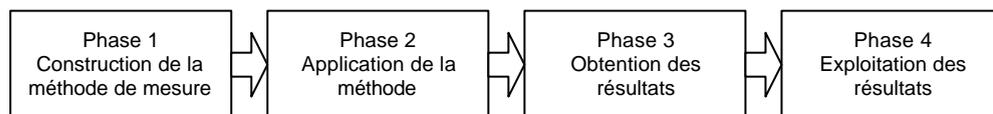


Figure 2-5 : Phases du processus de mesure [29]

La phase 4 constitue l'activité qui représente le plus d'intérêt pour les gestionnaires. C'est la phase finale d'un processus qui permet de construire, par exemple, les modèles de productivité et d'estimation nécessaires à la gestion. La phase 1, celle de la construction de la méthode de mesure, a été appliquée lors de la construction de la méthode de mesure fonctionnelle COSMIC-FFP. Les deux autres phases sont souvent ignorées par les chercheurs sur la mesure. Elles sont cependant essentielles à l'obtention de résultats cohérents de mesures, c'est-à-dire lorsque deux mesureurs, avec la même documentation (ou la même information) obtiennent les mêmes résultats de mesure. Très peu de recherches [30, 31] ont

⁴ Nous remplaçons le mot étape utilisé dans l'article de Jacquet et Abran par le mot phase pour éviter la confusion avec les étapes de la méthode COSMIC-FFP.

été réalisées pour la phase 3. C'est à cette phase que l'on valide les résultats avant de les analyser dans le but de créer des modèles d'estimation et de productivité.

La phase 2 nous intéresse ici plus particulièrement. Cette phase comprend trois activités : la collecte des données, la modélisation du logiciel et l'application des règles d'assignation numérique. À notre connaissance, il n'y a pas eu de recherches théoriques systématiques relatives à cette phase. Notons que Nishiyama et al. [31] ont abordé la question de l'applicabilité des règles de la méthode de mesure du point de vue de la qualité de la documentation.

2.1.6.1 La collecte des données

L'activité de collecte des données est spécifique à une organisation et, en général, elle n'est pas systématisée dans la documentation des méthodes de mesure. Par exemple, il y a des textes génériques sur le type de documentation que doit demander le "mesureur", mais dans la pratique, cela dépend de ce qui est disponible pour un projet spécifique dans une organisation, de l'expérience et des connaissances du "mesureur".

2.1.6.2 La modélisation du logiciel

Le logiciel à mesurer est modélisé en utilisant les règles{ XE "règles" } de la méthode de mesure dans ce qui est appelé la phase de mise en correspondance du logiciel dans COSMIC-FFP [2, p. 18]. Plusieurs étapes sont nécessaires.

Ces étapes sont les suivantes dans la méthode COSMIC-FFP :

- l'identification des couches des logiciels ;
- l'identification de la frontière de chaque logiciel ;
- l'identification des processus.

De plus, les informations peuvent provenir de plusieurs origines.

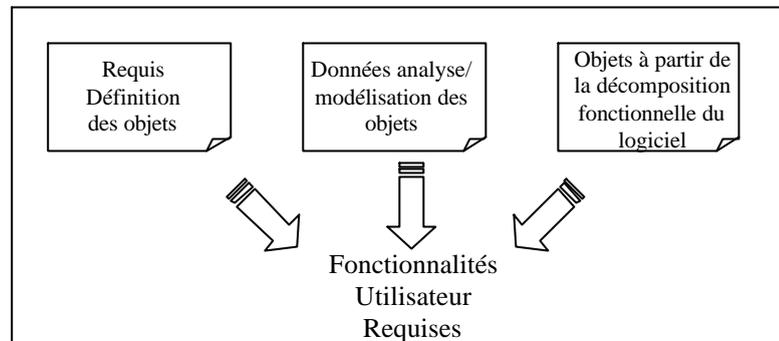


Figure 2-6 : FUR : documentation (de haut en bas) [2, section 2.2]

En effet, un logiciel peut être vu sous différents angles. Dans la perspective de la méthode de mesure COSMIC-FFP, la vision est celle des fonctionnalités livrées aux utilisateurs. Cette dernière est décrite via les Fonctionnalités Utilisateurs Requises (FUR) selon la définition de la norme ISO 14143-1 [9]. En pratique, les FUR existent parfois sous la forme de documents spécifiques (spécifications des besoins requis, par exemple), mais souvent les FUR doivent être dérivés des autres artefacts du génie logiciel. Tel qu'illustré à la Figure 2-6, les FUR peuvent être également dérivés à partir de la documentation avant même l'existence du logiciel (typiquement des artefacts de l'architecture et de la conception). Ainsi, la taille fonctionnelle d'un logiciel peut être mesurée avant son implantation.

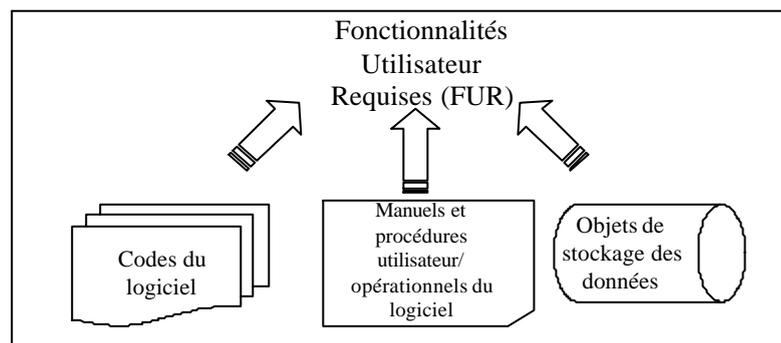


Figure 2-7 : FUR : documentation (de bas en haut) [2, section 2.2]

Dans d'autres circonstances, le logiciel peut être utilisé sans aucun ou avec peu d'objets provenant de l'architecture ou de la conception, et les FUR peuvent ne pas être documentés. Dans de telles circonstances, il est toujours possible de dériver les FUR du logiciel à partir des artefacts que l'on retrouve en exécutant le logiciel, tel qu'illustré à la Figure 2-7.

2.1.6.3 L'application des règles d'assignation numérique

Cette phase est dépendante de la première étape du processus de mesure et plus particulièrement de la définition des règles d'assignation numérique.

Ces dernières sont appliquées par le mesureur à partir de l'identification des mouvements de données (qui se qualifient comme tels – c'est-à-dire 'mesurables' selon le modèle spécifique de la méthode de mesure) dans le cas de COSMIC-FFP.

Dans COSMIC-FFP, l'unité de mesure est un mouvement de données élémentaire ou CFUT (COSMIC-FFP Unité de Taille).

2.1.7 Catégories d'utilisateurs de la mesure fonctionnelle

Les catégories d'utilisateurs pour la mesure fonctionnelle sont les suivantes :

- les entreprises qui appliquent la mesure fonctionnelle pour la mesure des logiciels ;
- les firmes-conseils qui offrent des services en mesure fonctionnelle auprès des entreprises ;
- les associations nationales et internationales de mesures, et plus spécifiquement celles qui maintiennent et diffusent des informations quant à l'interprétation et à l'application des mesures fonctionnelles (Annexe F) ;
- les experts eux-mêmes qui désirent échanger d'une façon formelle des informations quant à l'interprétation des règles{ XE "règles" } de la méthode de mesure fonctionnelles ;

- toutes les personnes qui doivent appliquer les règles de la méthode de mesure fonctionnelles COSMIC-FFP{ XE "COSMIC-FFP" } à un moment ou à un autre du cycle de vie du logiciel.

2.1.8 Reconnaissance de la méthode de mesure fonctionnelle

Dans cette sous-section, nous abordons l'application de la méthode de mesure sous l'angle de son utilisation dans l'industrie.

2.1.8.1 Rôle de l'expert

Pour appliquer la mesure fonctionnelle il faut une "expertise" de la part du mesureur. C'est ce qui nous amène à définir ce qu'est l'expertise humaine. Le mot "expert" ou "expertise" est le plus souvent accolé avec un autre mot (ex. : expert-conseil, expert en sinistre, expert à la cour, etc.). Dans le dictionnaire de l'Office québécois de la langue française [32], il y a plus d'une centaine d'expressions utilisant le mot "expert" et "expertise" accolé à un autre mot.

D'une façon générique, le rôle de l'expert dans l'entreprise est défini de la façon suivante :

"Travailleur dont la compétence et les connaissances dans un domaine particulier lui permettent d'évaluer et de recommander les améliorations à apporter dans le domaine de sa spécialité à l'intérieur d'un établissement commercial, industriel, gouvernemental ou autre" [32].

Il ne faut pas croire cependant que l'expert se doit de connaître son seul domaine. Peter Jackson [33] écrit à ce propos : "Human expertise, even in a relatively narrow domain is often set in a broader context that involves a good deal of commonsense knowledge about the everyday world" [33, p. 5].

C'est ce qui rend difficile la formalisation de l'expertise. Elle n'est pas limitée au strict domaine de l'expert, même si c'est à cette expertise que l'on fait appel.

2.1.8.2 Reconnaissance de la méthode de mesure fonctionnelle : approche industrielle

Il existe, dans un certain nombre de pays, des associations faisant la promotion, l'enseignement et l'interprétation des règles{ XE "règles" } de la méthode de mesure fonctionnelles (Annexe F). Ces associations répondent aussi aux questions des membres sur l'application des méthodes de mesure fonctionnelle. Elles ont toutes les mêmes préoccupations au sujet de la fiabilité des règles de la méthode de mesure entre mesureurs⁵ et de l'application des règles de la méthode de mesure. Pour cette raison, nous avons consulté les organisations suivantes pour leur demander comment elles procédaient pour résoudre les problèmes d'application des règles de la méthode de mesure :

- ASMA en Australie (Pam Morris)
- GUFPI en Italie (Roberto Meli)
- DESMA en Allemagne (Robert Hürten, Günter Büren)
- NESMA en Hollande (Jolijn Onvlee)
- JFPUG au Japon (Mr Kurasige)
- CIM au Canada (Denis St-Pierre)

Ces associations ont, le plus souvent, un comité de pratiques composé de membres reconnus pour leur expertise dans le domaine de la mesure fonctionnelle des logiciels. Ces experts se réunissent à intervalles réguliers (mensuel ou trimestriel) pour discuter de l'interprétation et de l'application des règles{ XE "règles" } de la méthode de mesure. Pour un certain nombre d'associations, les interprétations sont conservées dans un document appelé "règles locales" en référence au fait que les règles sont appliquées dans le contexte de technologies locales.

Nous avons demandé à ces associations leur mode de fonctionnement et les mécanismes utilisés pour s'assurer que leurs membres appliquaient de façon cohérente la méthode de mesure fonctionnelle.

5 La section suivante sur la validation des résultats de la mesure traite de ce problème.

Les résultats de notre consultation auprès des associations locales indiquent que :

- la plupart des associations ont des procédures administratives (nomination des membres du comité, conditions d'acceptation de nouvelles règles{ XE "règles" } pour la méthode de mesure utilisée et autres règles de nature administrative) ;
- la plupart des associations ont un guide d'interprétation des règles{ XE "règles" } de la méthode de mesure avec des exemples. Ce guide est souvent appelé "règles locales". Un document fournissant des exemples de ces "règles locales" se trouve à l'Annexe G ;
- il n'y aurait pas de documents dans ces associations sur la mise en correspondance des règles formelles de la méthode avec différents environnements ;
- la définition du cheminement du mesureur ne semble jamais avoir été discutée, pas plus que l'on peut retrouver un document visant à définir le cheminement du mesureur lors de la mesure fonctionnelle.

2.1.9 Validation d'une méthode de mesure

Pour Abran [21], la validation de l'application d'une méthode de mesure consiste, entre autres, à se poser des questions du type [21] :

- Comment savoir si l'application d'une méthode de mesure a été effectuée correctement ?
- Quel degré de confiance peut-on avoir dans le résultat obtenu après l'application d'une méthode de mesure ?

Il y a trois types de validation des résultats selon Jacquet et Abran [29]. Il s'agit de la validation de la construction d'une mesure (étape 1), la validation de l'application d'une méthode de mesure (étapes 2 et 3) et la validation des modèles analytiques (étape 4).

- validation de la construction d'une mesure (étape 1) : dans la littérature en génie logiciel, ce type de validation a principalement été abordé d'un point de vue

théorique. Une méthode de mesure est valide si elle vérifie le théorème de représentation [34]. Cependant, la validation de toutes les sous-étapes de l'étape de la construction d'une méthode de mesure n'a pas été complètement traitée ;

- validation des modèles analytiques (étape 4) : pour ce type de validation les auteurs se limitent souvent à des propositions d'un seul type de validation de modèles analytiques, soit les modèles dits de type prédictif ;
- validation de l'application d'une méthode de mesure (étapes 2 et 3) : le problème de la validation de l'application d'une méthode de mesure a rarement été abordé en génie logiciel, bien qu'il soit d'importance pour les praticiens.

Le processus de validation présenté par Desharnais [35] et Morris [30] porte à la fois sur les étapes 2 et 3 du processus de mesure. Ce qu'ils appellent la validation porte *a priori* sur l'étape 2 et *a posteriori* sur l'étape 3. Les études *a priori* portent principalement sur la structure de la mesure fonctionnelle. Les études de Desharnais et Morris ont porté sur la méthode de mesure fonctionnelle d'IFPUG. Elles sont cependant moins pertinentes pour la méthode COSMIC-FFP puisque la structure de cette dernière est différente.

Pour la validation *a priori*, quelques études ont été réalisées depuis 1989. Ces études nous intéressent car elles portent sur les résultats de l'application de la mesure fonctionnelle entre mesureurs ayant une information de bonne ou de mauvaise qualité.

Abran [21] cite trois recherches sur la fiabilité, dont la variation des résultats entre les mesureurs. Voici un tableau (2-1) comparatif des trois recherches, tiré de cette thèse :

Tableau 2-1 : Analyses de fiabilité de la mesure fonctionnelle par Abran [21]

Caractéristiques	Rudolph 89 [36]	Low et al. 90 [37]	Kemerer 90 [38]
Sujets et niveau d'expérience	Analystes expérimentés	22 analystes et 7 organisations	De 2 à 4 analystes dans plusieurs organisations
Formation	3 jours de formation	20 individus formés par les chercheurs	Manuel d'instruction et guides de mesure
Nombre d'applications mesurées et mesures	9 applications et 373 mesures	2 programmes et 22 mesures chacun	27 applications et 90 mesures
Méthodes statistiques	Jugement moyen Moyenne, Écart-type et différence des moyennes	Jugement moyen Moyenne, Écart-type et différence des moyennes	Tests statistiques H_0 et H_1 , ERM
Variation inter-mesureurs	Albrecht 79 : 30 % IFPUG 88 : 10 %	30 %	IFPUG 90 : 10.78 % IFPUG E/R : ERM = 17.30 % Inter-méthode : ERM = 8.48 %

Le Tableau 2-1 indique, entre autres, que la variation entre deux mesureurs utilisant la même méthode et les mêmes cas (programme ou logiciel), est de 10 % à 30 %, même s'ils ont reçu la même formation. Les causes de cette variation ne sont cependant pas analysées, sauf que l'on constate dans l'étude de Rudolph que la méthode utilisée a un impact (Albrecht 79 : 30 % et IFPUG 88 : 10 %).

Plus récemment Nishiyama [39] a étudié les différences entre deux mesureurs expérimentés. Dans cette étude, la différence relative moyenne entre les résultats des mesures des deux mesureurs est de 5 % pour les mesures de logiciels avec une documentation non structurée et de 2,8 % pour les mesures des logiciels avec une documentation relativement bien structurée. Nishiyama semble suggérer que ce résultat relativement bas serait lié au fait que les mesureurs avaient beaucoup d'expérience. À partir de notre expérience de la mesure, c'est une opinion que nous partageons, mais elle n'est pas démontrée par l'étude.

Il faut noter qu'elle porte sur seulement deux mesureurs (cinq logiciels), ce qui est un faible échantillon. L'intérêt de cette recherche est lié au fait que ce type d'analyse entre mesureurs expérimentés est relativement rare dans la littérature. L'opinion de Nishiyama a été confirmée par les résultats des recherches de Low et Jeffrey à l'effet que "le niveau d'expertise dans la mesure des points de fonction est un facteur important" [37].

Nishiyama va cependant un peu plus loin dans sa recherche en comparant l'identification des fonctions (EI, EO, EQ, ILF, EIF) entre mesureurs [39, p. 311]. Par exemple, la non correspondance entre la fonction EI pour les mesureurs A et B est respectivement de 57 % et de 60 %. Il constate qu'il y a des différences importantes dans l'identification des fonctions [39]. Les différences auxquelles fait référence Nishiyama sont en rapport avec l'identification des fonctions d'entrées, de sorties et d'interrogations que l'on retrouve dans la méthode de mesure d'IFPUG [24]. Cette différence au niveau de chaque fonction (on dirait, au niveau de chaque mouvement de données avec COSMIC-FFP) n'a jamais été analysée systématiquement à notre connaissance. Nishiyama semble le seul à avoir fait des recherches en ce sens.

La qualité de la documentation joue aussi un rôle important selon une autre étude de Nishiyama [31], mais surtout sous l'aspect de l'analyse des efforts pour la mesure. Dans ses conclusions, Nishiyama indique qu'il faut jusqu'à quatre fois plus d'effort pour mesurer un logiciel dont la documentation est non structurée, par rapport à un logiciel dont la documentation est bien structurée. L'étude des efforts nécessaires pour réaliser la mesure en tenant compte de la qualité des informations, bien qu'intéressante, ne fait pas partie de notre recherche.

2.1.10 Qualité de la documentation

Pour le mesureur dans l'entreprise, il y a aussi un autre facteur non négligeable à considérer : la qualité de l'information. La non-qualité de l'information peut être due au manque de documentation ou à la difficulté d'interpréter la documentation. Les recherches ne font généralement pas une distinction entre la connaissance et l'expérience du mesureur versus la

qualité de la documentation pour la cohérence des résultats de la mesure⁶. Nous avons supervisé une recherche sur la qualité de la documentation pour la mesure fonctionnelle [40] dans le cadre d'un mémoire de maîtrise. Cette recherche, à partir d'une vingtaine de projets informatiques, montre "que la plupart des processus fonctionnels identifiés sont cités ou listés dans la documentation fonctionnelle ; seulement un tiers des projets étaient assez documentés pour les fins de l'identification complète des processus "[40, p. 80].

Selon Prince, la documentation des logiciels (souvent sous forme de modèles) [41, p.86] est souvent incomplète, obsolète et même parfois erronée. De point de vue de la documentation, l'application de la méthode de mesure fonctionnelle est difficile car [11] :

- il y a une grande diversité des sources d'information pour obtenir ce qui est nécessaire à la mesure ;
- les sources ne sont pas homogènes puisque les façons de faire varient d'une organisation à une autre et même d'un projet à un autre ;
- la qualité de la documentation est inégale ;
- la documentation est souvent incomplète.

La Figure 2-8 indique que la qualité de la documentation peut être liée à la difficulté de trouver la bonne documentation (complète, à jour, juste), ou encore à la difficulté de comprendre la documentation (compréhension des modèles et du vocabulaire utilisé).

6 À noter qu'il est difficile de démontrer jusqu'à quel degré la qualité de l'information affecte la qualité des résultats des mesures par rapport à la connaissance et l'expérience des mesureurs.

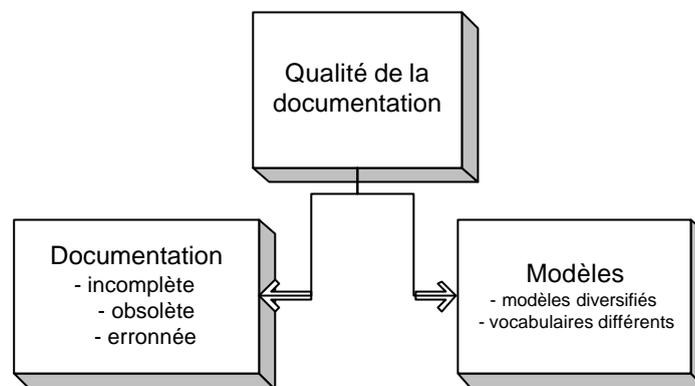


Figure 2-8 : Qualité de la documentation

Ainsi, les problèmes de documentation, lors de la conception et de la mise en place des logiciels, affectent la compréhension des logiciels pour des fins de mesure. De plus, même avec une bonne documentation, il est difficile de comparer les modèles d'une organisation particulière à une autre ; chaque organisation a sa propre méthodologie, ce qui veut aussi dire que les modèles de représentation, ainsi que le vocabulaire, sont différents.

Dans ces situations, le mesureur ne peut se fier complètement à la documentation et doit interroger les spécialistes qui ont réalisé le logiciel afin de compléter l'information manquante pour les fins de mesure. Dans notre recherche, nous n'allons pas aborder le sujet de la qualité des mesures en rapport avec la qualité de la documentation. Le sujet n'a jamais été abordé par les chercheurs à notre connaissance, la définition des concepts liés à la notion de qualité de la documentation et leur expérimentation pouvant faire l'objet d'une autre recherche.

2.1.11 Constatations et problématique de la mesure fonctionnelle

Il ressort de cette première revue de littérature que :

- pour appliquer une mesure de la taille fonctionnelle, il ne faut pas seulement connaître les règles de mesure de la taille fonctionnelle, mais aussi savoir modéliser le logiciel à mesurer à partir de ses artefacts ;
- la mesure est une mise en correspondance entre le monde empirique et le monde formel ;
- la mesure COSMIC-FFP est une mesure de la taille du logiciel et, de ce fait, s'appuie, entre autres, sur les exigences de l'utilisateur et sur le design du logiciel. (Il est aussi possible d'utiliser les autres entités du produit tels la codification et les tests, mais leur utilisation présente plus de difficultés que l'utilisation des exigences de l'utilisateur et du design) ;
- l'application de la méthode de mesure est une phase du processus de mesure à la fois dépendante de la qualité des données à collecter et de la capacité du mesureur à modéliser le logiciel à mesurer ;
- le monde empirique, qui est l'objet de la mesure fonctionnelle, se retrouve à travers le cycle de vie du logiciel ;
- il existe une grande variété de modèles et de méthodes de développement et de maintenance du logiciel ;
- le monde formel de la mesure fonctionnelle se trouve à travers les règles de la mesure fonctionnelle ;
- l'analyse des résultats des mesures fonctionnelles démontre qu'il y a un problème de fiabilité (10 % à 30 %). Ce problème serait moindre avec les experts (5 %) [39] ;
- on ne retrouve pas, dans la littérature, d'analyses systématiques des variations entre mesureurs au niveau de chaque processus ;

- on ne retrouve pas dans la littérature d'analyses des variations entre mesureurs en faisant une distinction entre la qualité de la documentation et les connaissances et l'expérience du mesureur.

Pour ces raisons, il est difficile de mesurer un logiciel à partir de ses artefacts en appliquant la méthode de mesure fonctionnelle de façon cohérente et fiable.

2.2 Domaine cognitif

Dans cette section, nous allons faire une brève revue de littérature des systèmes à base de connaissances, ce qui nous aidera à construire un système à base de connaissances qui peut répondre à la problématique de la mesure fonctionnelle COSMIC-FFP. La Figure 2-9 donne une vue d'ensemble de notre démarche de revue de littérature pour le domaine cognitif.

Cette section est composée des sous-sections suivantes. La sous-section 2.2.1 définit ce qu'est un système à base de connaissances, la sous-section 2.2.2 énumère les différents composants d'un système à base de connaissances en faisant un bref rappel de l'utilité des différentes techniques ou des méthodes utilisées, la sous-section 2.2.3 fait un survol de la nature et des objectifs d'un système à base de connaissances, la sous-section 2.2.4 justifie notre approche conceptuelle d'un système à base de connaissances et son contenu, la sous-section 2.2.5 traite de la modélisation de la connaissance, la sous-section 2.2.6 traite de la vérification et de la validation des modèles de connaissances et enfin la sous-section 2.2.7 indique en quoi notre approche pour un système à base de connaissances est originale.

La problématique et nos objectifs de recherche concluent notre revue de littérature (2.3).

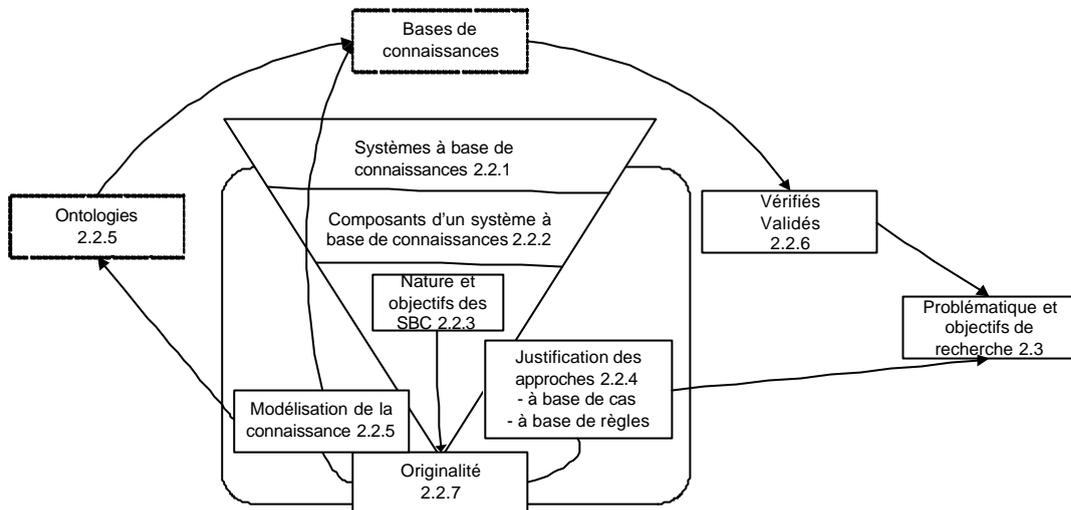


Figure 2-9 : Démarche de revue de littérature pour le domaine cognitif

2.2.1 Définition d'un système à base de connaissances

Dans cette sous-section, nous définissons les concepts de "connaissance" et de "système à base de connaissances" en reprenant notre questionnement sur la nature de l'expertise (voir 2.1.8).

2.2.1.1 Qu'est-ce que la connaissance ?

Pour mieux définir ce que signifie le concept "connaissance", il faut le mettre en relation avec deux autres concepts utilisés depuis quelques décennies en informatique : donnée et information. Pour Prince, le concept "donnée" est défini comme un "signal non interprété" [41, p. 25], le concept "information" est "un signifié transporté par une donnée"[41, p. 25] alors que le concept "connaissance" est "un mode d'emploi permettant de transformer les données en information" [41, p. 26]. Ainsi, selon Prince, les procédures, les règlements et les raisonnements sont des connaissances pour un domaine spécifique. Pour Schreiber et al. [42], le concept de connaissance se distingue du concept information sous deux aspects :

- il est la "machine intellectuelle" pour réaliser un but ;

- il permet potentiellement de générer de nouvelles informations [42].

Schreiber constate aussi que mettre en relation ces différents concepts, d'un point de vue pratique, est parfois difficile puisque "la connaissance est très dépendante de son contexte" [42]. Par exemple, si un consultant conclut que la taille fonctionnelle d'un logiciel en utilisant la méthode de mesure COSMIC-FFP est de 1 000 points de fonction, il peut être difficile de reconnaître s'il s'agit d'une donnée, d'une information ou d'une connaissance sans le contexte :

- pour le non-initié à cette méthode de mesure, c'est probablement une donnée puisqu'il ne peut pas l'interpréter ni l'utiliser ;
- pour une autre personne connaissant bien cette méthode, cela devient une information indiquant que le logiciel est d'une taille plus grande que la moyenne des logiciels développés dans les organisations. Il peut interpréter la donnée ;
- pour le gestionnaire qui a demandé cette mesure au consultant, cela devient une connaissance qui lui permettra de décider s'il va, par exemple, autoriser le développement du logiciel ou encore, demander une augmentation des budgets à la direction pour développer le logiciel.

À partir de cet exemple, selon le contexte et la personne, 1 000 points de fonction peuvent être interprétés comme une donnée, une information ou une connaissance.

Prince utilise aussi le terme "savoir-faire" pour indiquer qu'une personne sait optimiser les données et les informations reçues pour prendre les "bonnes actions" et générer les "bonnes décisions". Nous avons constaté, par exemple, que ce n'est pas parce qu'une personne "connaît" très bien les règles de la méthode de mesure COSMIC-FFP qu'elle est capable de faire une mesure de qualité et cela rapidement. Il lui faut une certaine pratique qui dépasse la simple "connaissance" des règles. Alors que la connaissance est "un mode d'emploi permettant de transformer les données en information", le savoir-faire est l'optimisation de ce mode d'emploi.

Pour Prince, le savoir-faire "peut être parfaitement explicite (perceptible) mais (dont) l'explication ou la description ne correspondent en rien à son application"[41, p. 30]. Le savoir-faire est "l'intuition du vétéran", le "flair de l'expert" ajoute Prince.

2.2.1.2 Qu'est-ce qu'un système à base de connaissances ?

Les expressions "système à base de connaissances" ou "système expert" sont souvent employées comme des synonymes. Les expressions "système de connaissances" et "ingénierie de la connaissance" sont aussi utilisées.

Voici ce que dit Prince au sujet de la distinction entre système à base de connaissances et système expert :

"...des raisons de pertinence, de vérité et de situation d'échec dans les organisations ont entraîné un abandon progressif de cette dénomination de système expert, aussi bien dans les milieux de la recherche que dans les structures industrielles. Celle qui tend à la remplacer est celle de système à base de connaissances initialement adopté pour des raisons de marketing et qui finit par avoir une authenticité académique" [41].

Jackson définit de la façon suivante un système expert :

An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problem or giving advice [33, p. 2].

À partir de la définition de Jackson, nous en déduisons que :

- le système expert possède un certain nombre de connaissances, que ce soit des algorithmes, des listes d'exemples, des questions/réponses, etc. ;
- une bonne partie des connaissances d'un système expert appartiennent à un domaine spécifique. Une collection de connaissances provenant de différents domaines sont rarement l'objet d'une expertise ;

- la connaissance doit permettre de résoudre des problèmes autres que des problèmes d'accès à une documentation en ligne (hyperliens) et peut aider à résoudre des problèmes ou à donner des avis.

Dans les années 1990, les expressions "ingénierie de la connaissance" et "système de connaissances" sont apparues dans la littérature. L'expression "système de connaissances" vient de l'expression "système à base de connaissances". D'autre part, l'ingénierie de la connaissance est l'art de construire des systèmes de connaissances. Voici ce qu'écrit Schreiber à ce sujet :

Knowledge engineering has evolved from the late 1970s onward, from the art of building expert systems, knowledge-based systems, and knowledge-intensive information systems. We use these terms interchangeably, and call them knowledge systems for short [42, p 6].

Dans la prochaine sous-section, nous présentons les différents composants d'un système à base de connaissances en vue d'établir une relation entre les différentes options offertes par les composants et la problématique de la mesure fonctionnelle (2.1.11).

2.2.2 Composants d'un système à base de connaissances

Les principaux composants d'un système à base de connaissances sont la base de connaissances (et la base de faits), le moteur d'inférence et les différentes interfaces utilisateurs. Nous avons ajouté une sous-section spécifique au traitement de l'incertitude [43-46]. Nous passons en revue ces composants afin de les mettre en relation avec la problématique de la mesure fonctionnelle. Cette mise en relation nous permet de faire ressortir les approches et les techniques du domaine cognitif qui nous semblent les plus pertinentes pour résoudre notre problématique. La justification de notre choix se fait dans les sous-sections qui suivent (2.2.3, 2.2.4 et 2.2.5).

2.2.2.1 Base de connaissances

La base de connaissances comprend des faits et des règles. Ce sont les règles qui permettent de manipuler les faits [43]. Pour exprimer la connaissance, il faut un mode de représentation. Les principaux modes de représentation sont : les réseaux sémantiques, les objets structurés, les réseaux neuronaux et les règles de production. C'est aussi ce mode de représentation qui déterminera comment seront formalisées les connaissances dont on dispose au sujet d'un domaine donné.

Voici une brève description de chacun de ces modes de représentation tirée de Karkan et Tjoen [43, Chapitre 3] :

Réseaux sémantiques

Voici ce que disent Karkan et Tjoen sur les réseaux sémantiques :

"Un réseau sémantique est constitué de différents nœuds représentant des concepts reliés par des arcs exprimant les relations qui les lient entre eux. Il s'agit d'un graphe orienté sans cycle où les nœuds représentent les concepts et les arcs décrivent les liens sémantiques entre ceux-ci." [43]

Nous retenons que les réseaux sémantiques sont utiles lorsque la connaissance doit être structurée de façon à ce qu'il y ait des liens sémantiques entre les concepts représentés par des nœuds. Dans le cadre de la problématique de la mesure fonctionnelle, la connaissance peut être encodée sous forme de règles. Nous ne sommes pas intéressés à chercher le lien sémantique entre deux concepts.

Objets structurés

L'approche objet est aujourd'hui utilisée aussi bien pour décrire une application informatique de type gestion (ou en temps réel) que pour décrire un système à base de connaissances. Pour un système à base de connaissances "l'objet est une nouvelle structure de données consistant à mêler dans une même structure des connaissances déclaratives (les données) et procédurales (les procédures ou méthodes)" [43]. La notation UML est aussi associée étroitement à

l'approche objet. La notation UML est utilisée dans CommonKADS [42]. Cette approche peut être combinée à une autre approche de représentation de la connaissance. Elle est utile, entre autres, pour décrire le design d'un système à base de connaissances. C'est d'ailleurs cette notation (UML) que nous avons utilisée dans le cadre de la construction de nos prototypes (voir Chapitres 5 et 6).

Réseaux de neurones

Les approches réseaux de neurones ont comme point d'origine l'analogie au cerveau humain pour proposer une représentation de la connaissance. Aujourd'hui, les chercheurs dans les approches réseaux de neurones n'ont plus la préoccupation d'imiter le cerveau humain. Avec les réseaux de neurones, le pouvoir d'apprendre vient de la capacité d'associer des entrées avec des sorties sans avoir de règles d'associations explicites.

Dans le cadre de la problématique de la mesure fonctionnelle, le monde formel de la mesure fonctionnelle se trouve à travers les règles de la mesure fonctionnelle et celles de l'expert. Ces règles sont disponibles ainsi que la connaissance de l'expert qui peut justifier les résultats obtenus. Cette approche n'est donc pas pertinente dans le cadre de la problématique de la mesure fonctionnelle.

Règles de production

Les systèmes à base de connaissances ont, pour la plupart, des règles de type IF... THEN... [43] pour exprimer la connaissance. Il s'agit de règles de production qui sont utilisées dans les systèmes à base de connaissances où l'on veut représenter au mieux (de façon cohérente et fiable) le raisonnement de l'expert.

Dans le cadre de la problématique de la mesure fonctionnelle, nous avons indiqué que le monde formel de la mesure fonctionnelle se trouve à travers les règles de la mesure fonctionnelle (2.1.11 g). La représentation de la connaissance du mesureur expert peut donc être exprimée sous forme de règles de production. Les règles de production sont manipulées par un moteur d'inférence qui est un outil pour la résolution de problèmes. La prochaine sous-section traite de ce sujet.

2.2.2.2 Résolution de problèmes

Le moteur d'inférence permet d'expliciter la connaissance (voir 4 phases plus loin) selon un mode de représentation spécifique. Le moteur d'inférence est un cas particulier qui permet de faire de la résolution de problèmes en utilisant les règles de production. Tout moteur d'inférence est caractérisé par un cycle de base, une stratégie de recherche et une méthode de chaînage.

Cycle de base

Voici en résumé les quatre phases du cycle de base selon Karkan et Tjoen [43, Chapitre 4] :

- A. la phase de sélection qui a pour objet de trier et de rassembler en un sous-ensemble les faits et les règles de la base de connaissances qui méritent plus d'attention que d'autres ;
- B. la phase de filtrage, au cours de laquelle le moteur d'inférence compare la partie prémisse des règles sélectionnées avec les faits de la base de faits pour déterminer l'ensemble des règles applicables ;
- C. la phase de résolution de conflits qui se concrétise par le choix de la règle à appliquer ;
- D. la phase d'exécution qui consiste à appliquer (activer) la règle choisie lors de l'étape précédente.

Ces phases sont à mettre en relation avec les étapes de la démarche humaine pour résoudre un problème. Nous utilisons la démarche proposée par Laurière [47] pour illustrer cette relation entre la démarche humaine et les phases du cycle de base selon Karkan et Tjoen [43, Chapitre 4].

Pour Laurière [47], un problème est ce qui est ressenti par une personne : "Une personne ressent un problème dès qu'elle se trouve dans une situation où elle désire quelque chose alors qu'elle ne voit pas immédiatement la suite d'actions à accomplir pour obtenir ce

quelque chose" [47, p. 9]. Cette définition d'un "problème" est similaire à celle de Budgen en génie logiciel c'est-à-dire que l'ingénieur "doit travailler d'une façon créative pour identifier les propriétés requises à la solution..." [48, p. 104].

Voici les étapes de la démarche que Laurière [47, p. 19] a tiré du travail du mathématicien et didacticien George Polya [49]. Nous présentons des extraits des questions suggérées à chaque étape :

A – Comprendre le problème

"Quelles sont les inconnues ? Quelles sont les données ? Quelle est la condition ? Est-il possible de satisfaire à la condition ? La condition est-elle suffisante pour déterminer l'inconnue ? Est-elle insuffisante ? Redondante ? Contradictoire ?"

Ceci correspond à la phase A du cycle de base.

B – Concevoir un plan

"Trouvez le rapport entre les données et l'inconnue. Vous pouvez être obligé de considérer des problèmes auxiliaires si vous ne trouvez pas un rapport immédiat entre les données et l'inconnu. Trouvez le plan de la solution. Avez-vous déjà rencontré ce problème ? Ou bien avez-vous rencontré le même problème sous une forme légèrement différente ? Connaissez-vous un problème qui s'y rattache ? Connaissez-vous une théorie qui puisse être utile ?"

Ceci correspond à la phase B du cycle de base.

C – Mettre un plan à exécution

"En mettant votre plan à exécution, vérifiez-en chaque détail l'un après l'autre. Pouvez-vous voir clairement si ce détail est correct ? Pouvez-vous démontrer qu'il est correct ?"

Ceci correspond aux phases C et D du cycle de base.

D – Examiner la solution obtenue

"Pouvez-vous valider le résultat ? Pouvez-vous vérifier le raisonnement ? Pouvez-vous obtenir le résultat différemment ? Pouvez-vous vous servir du résultat ou de la méthode pour quelque autre problème ?"

Les questions qui sont posées, telles "pouvez-vous valider le résultat ?" et "pouvez-vous vérifier le raisonnement ?" indiquent que la solution doit être justifiée. La question "pouvez-vous vous servir du résultat ou de la méthode pour quelque autre problème ?" indique que la solution pourrait aider à trouver d'autres faits. Nous sommes alors en dehors du moteur d'inférence.

En conclusion, le cheminement humain de résolution de problèmes est une démarche qui se doit d'être décrite en plusieurs étapes. Cette démarche resterait sans intérêt pratique si les connaissances nécessaires ne sont pas présentes. Comment se fait ce transfert ? Est-il possible de le modéliser ? C'est à ces questions que nous tentons de répondre dans les prochaines sous-sections.

Enfin, la formalisation de la démarche du mesureur (Chapitre 4) s'appuie sur ces prémisses.

Stratégie de recherche

Le moteur d'inférence a une stratégie de recherche. Il existe un certain nombre de stratégies de recherches qui sont caractérisées par une situation de départ, un but à atteindre et un espace de recherche [43].

Les principales stratégies de recherches sont : la recherche en largeur d'abord, la recherche en profondeur d'abord, la recherche en profondeur limitée et la recherche heuristique.

Les trois premières recherches sont basées sur un arbre de décision avec un ordre de prise en considération suivant un cheminement précis et exhaustif, soit à partir de la racine de gauche à droite, ou à partir de la racine en explorant une branche jusqu'à son terme avant de passer à une autre branche ou enfin en combinant les deux approches précédentes.

Leur efficacité est directement reliée à la configuration initiale présentée. Il s'agit de stratégies de recherche exhaustives. Pour des problèmes relativement complexes avec de nombreuses possibilités, ces stratégies de recherche perdent de leur efficacité.

D'autre part, la stratégie de recherche heuristique vise à exploiter la connaissance spécifique sur le problème traité plutôt que de procéder à des recherches aveugles. Dans le cadre de la problématique de la mesure fonctionnelle, l'heuristique tient compte du fait que le mesureur peut donner les réponses qu'il croit les meilleurs pour solutionner son problème. La valeur de l'heuristique vient de la possibilité que possède le mesureur de fournir des réponses à différentes étapes du processus. Il est donc préférable d'utiliser cette approche puisque la méthode de mesure fonctionnelle COSMIC-FFP est une mise en correspondance entre le monde empirique et le monde formel (2.1.11 b). Enfin, l'application de la méthode de mesure fonctionnelle COSMIC-FFP est une phase du processus de mesure dépendante de la capacité du mesureur à modéliser le logiciel à mesurer (2.1.11 d). Les interactions, entre le mesureur et le système à base de connaissances, sont nécessaires. La stratégie de recherche heuristique permet d'exploiter la connaissance spécifique du mesureur sur le problème traité.

Méthodes de chaînage

Pour les méthodes de chaînage, la stratégie de recherche est basée sur une certaine configuration du problème à traiter. La méthode de chaînage est guidée par les données ou par les buts, selon qu'il s'agisse d'un chaînage avant ou d'un chaînage arrière.

Voici comment Karkan et Tjoen décrivent le chaînage avant :

"Le moteur d'inférence raisonne en partant des données et essaye d'en déduire de nouvelles, qui permettront à leur tour de déduire le but. Ceci signifie que l'on ne sélectionne, au cours de la phase de filtrage, que les règles dont les conditions de la partie gauche (les prémisses) sont vérifiées. La phase de résolution de conflits est ensuite appliquée sur cet ensemble de règles, pour déterminer celles à utiliser en priorité. La base de faits est, lors de la phase d'exécution, mise à jour par l'application de la règle retenue. Ce processus est réitéré jusqu'à ce que le but soit atteint ou qu'aucune règle ne soit plus applicable" [43, p. 38].

Le chaînage avant s'avère une bonne stratégie dans les situations où le système doit interpréter une série de faits proposés.

Voici comment Karkan et Tjoen décrivent le chaînage arrière :

"Cette seconde stratégie de contrôle est dirigée par les buts : le moteur d'inférence prend le but final comme hypothèse et tente de remonter aux faits pour le démontrer. Les règles sélectionnées sont celles dont la conclusion correspond au but recherché. Les conditions inconnues de la prémisse de ces règles deviennent des sous buts à démontrer. Le processus est répété de manière récursive jusqu'à ce que tous les sous buts soient à leur tour démontrés (ce qui implique que le but lui-même est démontré), ou alors jusqu'à ce qu'il ne soit plus possible de sélectionner de règles" [43, p. 43].

Le chaînage arrière s'avère une bonne stratégie lorsqu'il faut remonter aux faits pour en arriver à une conclusion.

Il est aussi possible de réaliser un chaînage mixte en vue de pallier aux inconvénients des deux premières approches tout en alliant leurs avantages. Le choix d'une stratégie est guidé par l'efficacité de la recherche.

Dans le cadre de la problématique de la mesure fonctionnelle, les situations suivantes peuvent se produire :

- le mesureur doit trouver les faits qui s'appliquent à un "problème" spécifique. La meilleure stratégie est de trouver les faits qui s'appliquent à un "problème" plutôt que de trouver tous les problèmes qui s'appliquent à un fait. Dans ce cas le nombre de faits est limité et le nombre de "problème" est beaucoup plus grand. La stratégie de chaînage avant est alors plus efficace ;
- le mesureur cherche une solution optimale sur la base des faits. La stratégie consiste à partir de la solution optimale pour trouver les faits qui s'appliquent à la solution et non de trouver toutes les solutions relatives à un fait. Dans ce cas, c'est la

combinaison des faits qui intéresse le mesureur. La stratégie de chaînage arrière est alors plus efficace.

2.2.2.3 Interfaces utilisateurs

Puisqu'il y a un (ou des) utilisateur(s) des systèmes à base de connaissances, il leur faut normalement une interface et, si possible, conviviale. Dans le cadre de notre projet de recherche, nous avons adopté une approche incrémentale : création d'un premier prototype "local" avec une expérimentation (Chapitre 5) et création d'un deuxième prototype accessible via un site WEB (Chapitre 6). La description du design des prototypes se trouve aux Annexes B et C.

2.2.2.4 Traitement de l'incertitude

Le traitement de l'incertitude se réalise par différentes approches mathématiques.

Les raisonnements à base de règles utilisent parfois des formules mathématiques (MYCIN est un exemple) pour trouver (ou proposer) une solution. Nous avons examiné trois approches mathématiques : la théorie "baysienne", la théorie de la certitude et la logique floue. Nous allons décrire chacune brièvement et faire ressortir leurs avantages et leurs inconvénients.

Théorie "baysienne"

La théorie "baysienne" est une approche par raisonnement inexact. Traditionnellement, le raisonnement inexact en est un de probabilité mathématique. La théorie des probabilités repose sur des essais répétitifs et effectués au hasard permettant de déterminer un nombre $P(E)$ appelé probabilité de l'événement. L'ensemble des résultats possibles pour une expérience est appelé l'espace d'échantillonnage qui est noté par (S) [50, p. 306]. Selon Durkin [50], si "les événements sont indépendants et que l'on désire connaître la probabilité de plus d'un événement, on a une probabilité composée". Si les événements sont dépendants, on utilise la probabilité conditionnelle. La question est : quelle est la probabilité que l'événement A survienne compte tenu que l'événement B est déjà survenu ? [50, p. 309]

Voici la formulation de probabilités composées et conditionnelles :

$P(A \wedge B)$: probabilité composée de A et B

$P(A \wedge B) = P(A) * P(B)$ si A et B sont indépendants

$P(A \wedge B) = P(A) * P(B/A)$ autrement (il y a plusieurs variantes possibles ici).

La théorie des probabilités est une technique qui permet de travailler avec de l'information incertaine ou au hasard [50, p. 306]. Il y a cependant un certain nombre de contraintes [50, p. 329] :

- cette théorie implique que l'échantillon est bien défini et que les probabilités de chaque événement peuvent être obtenues à partir d'un ensemble de données historiques ;
- lorsque les informations changent, il faut recalculer tous les facteurs de probabilités, ce qui implique un effort de maintenance supplémentaire ;
- la somme des probabilités pour et contre une hypothèse, étant donné un événement, doit être égale à 1. Dans les faits, les experts peuvent donner une valeur positive à un événement (ex. : 0,7), mais hésitent à donner une valeur de contrepartie (ex. : 0,3) qui permettrait d'avoir une somme de probabilité égale à 1 pour une hypothèse ;
- pour la théorie "baysienne", on assume l'indépendance conditionnelle des événements, ce qui n'est pas toujours le cas.

Les théories alternatives pour traiter les problèmes avec des informations inexactes et pour lesquelles les informations *a priori* demeurent manquantes sont : la théorie de la certitude et la logique floue.

Théorie de la certitude

La théorie de la certitude a été développée par Shortcliffe et Buchanan lors des travaux de construction du système expert MYCIN. Elle repose sur une mesure de croyance plutôt que sur les probabilités strictes [50, p. 333].

Pour Durkin [50], il y a trois types d'incertitudes à considérer :

- l'incertitude des faits [50, p. 334] ;
- l'incertitude des règles [50, p. 335] ;
- l'incertitude des inférences [50, p. 335].

L'incertitude des faits oblige l'expert à prendre des décisions en composant avec des informations incomplètes et incertaines. Il dira : "il semble que...", "nous croyons que...". Ce sont alors des croyances exprimées par l'expert et non des certitudes. Le facteur de certitude ou *Certainty Factor* (CF) représente alors une croyance. CF peut avoir une valeur entre -1 (totalement faux) et +1 (totalement vrai). Une valeur positive représente un degré de croyance, et une valeur négative un degré de non-croyance. Le jugement d'un utilisateur sur la qualité de l'information (le fait) obtenue (ex. : complète = 1, sommaire = 0,8, assumée = 0,7 ou extrapolée = 0,6) est un exemple de la représentation de l'incertitude des faits.

L'incertitude des règles oblige l'expert à dériver une inférence inexacte à partir de l'information disponible. Autrement dit, l'expert, à partir de faits certains, ne peut croire que partiellement à la conclusion. Dans ce cas, le groupe MYCIN a décidé d'attribuer une valeur (entre 0 et +1) pour le facteur de certitude de chaque règle. Par exemple, le CF aura une valeur de 0,7 pour signifier que la croyance à la règle n'est pas absolue, mais seulement à 70 %.

L'incertitude de la conclusion (provenant de l'inférence) est liée au fait qu'il y a un lien entre l'événement et la conclusion avec les règles. Si la croyance à l'événement diminue, la croyance à la conclusion diminuera (%).

Si l'expert reçoit des confirmations de ses hypothèses de différentes sources, sa confiance dans son hypothèse sera plus grande. Comment évaluer cette confiance ?

Voici un tableau (2.2) avec un exemple :

Tableau 2-2 : Renforcement des croyances (règles)

Règle 1	Règle 2
SI A	SI C
ET B	ET D
ALORS Z	ALORS Z
CF = 0,8	CF = 0,7

Utilité de la théorie de la certitude

L'équipe de développement de MYCIN a introduit dans son système expert des techniques de raisonnement "inexact". Ces techniques sont utiles lorsque l'approche probabiliste est peu appropriée et qu'il n'est pas possible d'obtenir des informations statistiques fiables sur le problème. L'application des règles de la mesure ne repose pas sur des approches probabilistes. Il faudrait pour cela avoir un plus grand nombre de mesures, et ce jusqu'au niveau le plus bas de la mesure. Tel que déjà mentionné dans la problématique de la mesure fonctionnelle "on ne retrouve pas dans la littérature, connue à ce jour, d'analyses systématiques des variations entre mesureurs au niveau de chaque processus". Les techniques de raisonnement "inexact" sont donc à considérer dans notre approche.

Logique floue

En mathématique, le terme *fuzzy* a été utilisé au début du siècle passé par Lukasiewicz [51] lorsqu'il a voulu mettre en relation des termes tels que "grand", "vieux", "haut". Ces termes ont une nature binaire (vraie ou fausse) qui semble simple : cependant, l'appartenance des objets ou des personnes à un de ces termes n'est pas facile à déterminer (ex. : à partir de quel taille est-on grand ?). Il a ajouté un "possible" à "vrai" et à "faux". Ses recherches ont conduit à une technique formelle de raisonnement inexact appelée théorie des possibilités.

Durkin définit la logique floue de la façon suivante : "une branche de la logique qui utilise des degrés d'appartenance à un ensemble, plutôt qu'à une appartenance stricte vraie ou fausse" [50].

La logique floue est donc nécessaire lorsque le degré d'appartenance n'est pas strictement vrai ou faux. Est-ce utile d'utiliser la logique floue dans le cadre de la problématique de la mesure fonctionnelle ? Dans le cadre de cette problématique, le mesureur doit décider s'il considère que l'élément à mesurer fait partie de la mesure ou non. Il peut hésiter et ne pas répondre catégoriquement, mais alors cette "réponse" serait peu utile pour les fins de la mesure.

De plus, pour le mesureur, lorsqu'il utilise la méthode de mesure fonctionnelle, les artefacts à mesurer font partie ou ne font pas partie de la mesure c'est-à-dire que le degré d'appartenance est de 0 ou 1.

Conclusion sur les approches mathématiques

La théorie de la certitude semble appropriée pour l'expert de la mesure dont la problématique de résolution de problèmes est de nature binaire (il doit trouver une solution vraie ou fausse pour qu'elle soit utile) avec peu de données statistiques. L'expert a un grand nombre de décisions à prendre pour mesurer un seul logiciel, et il ne peut se fier qu'à ses connaissances du domaine et à son expérience pour prendre des décisions.

Voici les raisons qui nous font croire que la théorie de la certitude répond à notre problématique de la mesure fonctionnelle pour la partie approche à base de règles :

- il est possible de donner un poids, non seulement aux faits, mais aussi à chaque thème⁷ du point de vue de l'expert. Il serait aussi théoriquement possible pour le mesureur de donner un poids à la qualité de la documentation ;

⁷ Le thème correspond à l'hypothèse dans la théorie de la certitude.

- il est possible de combiner les thèmes pour évaluer un cas problème c'est-à-dire qu'un thème particulier combiné à un autre, peut influencer le résultat global compte tenu des faits apportés à chaque thème ;
- il est possible de tenir compte des valeurs positives et négatives des poids en combinant les thèmes.

2.2.2.5 Conclusion pour cette sous-section

Nous avons examiné les composants du système à base de connaissances et souligné les options qui peuvent être retenues dans le système à base de connaissances que nous devons construire pour résoudre "au mieux" la problématique de la mesure fonctionnelle (2.1.11). Dans la prochaine sous-section, nous examinerons les principaux objectifs des systèmes à base de connaissances et justifierons nos choix pour résoudre "au mieux" la problématique de la mesure fonctionnelle.

2.2.3 Nature et objectifs des systèmes à base de connaissances

Le terme "systèmes à base de connaissances" couvre un grand nombre de systèmes. Une classification de ces systèmes, à partir de leurs objectifs, a été proposée par Hayes-Roth en 1983 (cité par Jackson [33]). Il s'agit de systèmes :

- d'interprétation de situations décrites à partir d'observations ou à partir de senseurs ;
- de prédiction de conséquences possibles à partir d'une situation ou d'un événement ;
- de diagnostic à partir des symptômes ;
- de design d'une configuration d'objets qui satisfait certaines contraintes ;
- de planification d'une séquence d'actions pour réaliser des buts identifiés ;
- de suivi des comportements d'un système dans le temps, en vue de l'empêcher de dévier de ses buts ;
- de "débugage" ou d'identification des fautes de système ;

- de réparation des fautes de système ;
- d'instruction de diagnostic et de traitement des incompréhensions pour un domaine de connaissances ;
- de contrôle de comportement des systèmes en anticipant des problèmes, en planifiant des solutions et en faisant le suivi des actions nécessaires.

Le choix d'une classe de système à base de connaissances à partir de la classification de Hayes-Roth, est lié à la situation du mesureur (cette situation découle de la problématique de la mesure fonctionnelle) :

- règle générale, le mesureur n'a pas accès aux textes sous un format électronique structuré, ce qui exclut les approches à base de données et les approches statistiques brutes ;
- le mesureur doit effectuer un raisonnement impliquant une manipulation explicite de concepts (symboles) ;
- en raison de la diversité des méthodes de développement dans les organisations et de leur constante évolution, le mesureur se trouve fréquemment dans des situations où des cas nouveaux lui sont présentés ;
- les nouveaux "cas problèmes" du mesureur peuvent avoir des solutions existantes ou similaires. Il est possible qu'un thème (question) déjà utilisé pour un problème, puisse l'être pour un autre problème (c'est ce qui rend intéressantes les technologies à base de règles) ;
- il arrive que l'on demande au mesureur de justifier les solutions associées à chaque problème. Cette exigence n'est pas absolument nécessaire, mais obtenir la justification des solutions peut, de plus, aider le mesureur à apprendre plus rapidement le pourquoi du raisonnement de l'expert, ce qui exclut les technologies d'apprentissage automatique.

Les objectifs d'interprétation de situations décrites à partir d'observations (ici la documentation des exigences de l'utilisateur) et de diagnostic à partir de symptômes (ici les

symptômes sont remplacés par des règles déjà fournies par le mesureur expert), sont les objectifs qui permettent de résoudre "au mieux" la problématique de la mesure fonctionnelle.

L'interprétation de situations est traitée par analogie à partir de raisonnements à base de cas. Le diagnostic à partir de symptômes est traité à partir de raisonnements à base de règles. Doit-on choisir l'une ou l'autre technologie ? Pas nécessairement.

Les chercheurs suggèrent la construction de systèmes hybrides lorsque nécessaire [33, 44, 52-54]. Notons que Watson suggère que MYCIN⁸ était un système hybride (raisonnements à base de règles et de cas) avant qu'il ne soit question des raisonnements à base de cas.

La prochaine étape de notre revue de littérature est de nous demander quelles sont les technologies qui correspondent le mieux :

- à l'interprétation de situations décrites à partir d'observations et
- au diagnostic à partir de symptômes,

et de justifier ces technologies en rapport avec la problématique de la mesure fonctionnelle.

2.2.4 Justification d'un système à base de connaissances et son contenu

2.2.4.1 Présentation des technologies liées à notre choix

Les technologies présentées par Watson, en plus des techniques de raisonnement à base de cas (CBR) sont : les bases de données, la recherche d'informations textuelles, les statistiques, les apprentissages automatiques, les réseaux neuronaux et les systèmes à base de règles [44, p. 44-49]. Nous décrivons uniquement les techniques de raisonnement à base de cas et les

⁸ MYCIN n'est pas un acronyme, mais un suffixe pour un agent antibactériel. Le terme MYCIN est utilisé pour signifier plutôt EMYCIN qui est le système expert de MYCIN et qui signifie *Essential MYCIN*.

techniques de raisonnement à base de règles puisque ce sont les technologies les plus pertinentes aux approches de type diagnostique [44, p. 50-51].

Nous allons d'abord définir chacune des techniques de raisonnement et ensuite justifier leur utilisation.

2.2.4.2 Technique de raisonnement à base de cas

Dans cette sous-section, nous allons définir ce qu'est un cas, les limites de la technique, son cycle et quelques exemples de représentations dans la littérature.

Définition (cas)

Pour Kolodner un cas est un "morceau" de connaissance "contextualisé" et représentant une expérience [1, p. 13]. Le cas représente une expérience qui enseigne une leçon fondamentale pour réaliser les objectifs de la personne qui étudie le cas [1].

Kolodner fournit aussi des indications de haut niveau permettant de répondre à des questions potentielles des ingénieurs de la connaissance [1, Chapitre 5] :

- Quels sont les cas à placer dans le système de connaissances ?
- Comment délimiter la frontière d'un cas à un autre ?
- Comment puis-je diviser en différentes parties un cas qui serait très complexe ?

Plusieurs constatations se dégagent de ce que Kolodner [1, p. 8-14] a écrit au sujet des cas :

- un cas représente une connaissance spécifique liée au contexte. Il enregistre la connaissance à un niveau opérationnel ;
- un cas vient selon différentes formes et différentes tailles, couvrant une durée relativement grande ou petite, associant des solutions avec des problèmes, des résultats avec des situations, ou les deux ;

- un cas enregistre des expériences qui sont différentes à chaque fois. Cependant, toutes les différences ne sont pas importantes à enregistrer. Les cas qui méritent d'être enregistrés sont ceux qui donnent une leçon utile ;
- les leçons utiles sont celles qui ont la possibilité d'aider une personne à réaliser un but ou un ensemble de buts plus facilement dans le futur, ou qui avertissent de la possibilité d'un échec ou mettent en évidence un problème imprévu.

Dans notre recherche, nous avons privilégié, entre autres, les "études de cas" représentatives du domaine de la gestion des affaires et du domaine du temps réel. Ces "études de cas" nous ont aidé à produire des "cas problèmes" couvrant tous les principaux concepts de COSMIC-FFP. Dans notre recherche l'expression "cas problème" est synonyme de "cas" (voir 2.2.4.5).

Selon Kolodner [1], l'utilisation d'un CBR repose sur un certain nombre d'hypothèses :

- la régularité : les mêmes actions réalisées sous les mêmes conditions devraient donner les mêmes résultats ou des résultats similaires ;
- la répétition (caractéristiques types) : les événements ont tendance à se répéter ;
- la cohérence : des changements mineurs dans la réalité requièrent des changements mineurs dans notre raisonnement, ce qui veut dire qu'il y aura des changements mineurs dans la solution.

Limites de la technique de raisonnement à base de cas

Watson [44] a comparé différentes technologies en rapport avec la technologie des CBR. Il conclut que la technique de raisonnement à base de cas (CBR) peut fournir une solution adéquate plus rapidement que les autres techniques lorsque c'est possible. La technique de raisonnement à base de cas (CBR) a cependant des limites [44, p. 49] :

- difficultés à manipuler un grand nombre de données numériques ;
- difficultés à fournir des informations précises ou exactes c'est-à-dire non ambiguës.

Une autre limite est de prévoir une structure de contrôle qui décide l'ordre d'application des règles et leur enchaînement.

Cycle de la technique de raisonnement à base de cas (CBR)

Pour Watson, un CBR comprend un cycle. La Figure 2-10, adaptée de Watson [44], décrit le cycle du CBR de la façon suivante :

- identification du problème ;
- retrouver un cas qui correspond au problème identifié ;
- suggérer une solution au problème identifié à partir du cas ;
- réviser, tester et arranger le cas pour qu'il s'adapte au problème ;
- retenir le cas comme un nouveau cas pouvant être réutilisé pour un problème similaire ;
- l'inscrire dans la base de connaissances contenant les cas.

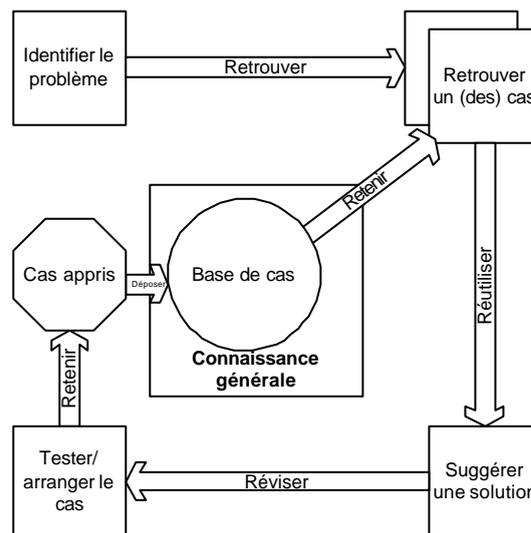


Figure 2-10 : Cycle du CBR (adaptation à partir de Watson)

Selon Watson [44, p. 69], il y a plusieurs classes de CBR. Il distingue entre autres les CBR de type interprétatif et de résolution de problèmes. Un CBR interprétatif implique l'utilisation de cas passés ou précédents, pour analyser et justifier l'interprétation d'un nouveau cas. Un CBR de type résolution de problèmes adapte une solution passée pour rencontrer les exigences de la nouvelle situation. Avec les CBR on peut utiliser, selon Marling, plusieurs techniques et les combiner [52, p. 70]. D'autres auteurs abondent dans le même sens [53-56]. Une telle combinaison peut être utilisée de différentes façons. Par exemple, Golding et al. [54] utilisent d'abord les règles pour obtenir une réponse préliminaire à un problème donné, puis établissent une analogie avec des cas pour mieux "manipuler" les exceptions aux règles. Il est aussi possible de partir de cas et d'appliquer ensuite les meilleures règles possibles au cas retenu.

Méthodes d'indexation

L'indexation, dans la technique de raisonnement à base de cas, est une méthode de recherche des cas dans la base de cas. Il y a deux principaux problèmes associés à l'indexation. Selon Kolodner [1, p. 193], ils consistent :

- à associer la bonne étiquette à chaque cas, de façon à pouvoir les retrouver une fois dans la base de cas ;
- à bien structurer la base de cas afin de permettre une recherche efficace (ex. : rapidité) et précise (ex. : trouver le bon cas), ce qui est moins important avec une centaine de cas, mais peut le devenir avec un plus grand nombre de cas.

Pour le premier problème, Kolodner écrit que "plus la personne qui indexe connaît le domaine, meilleure est l'indexation" [1, p. 193]. Elle ajoute que "l'indexation doit se faire à partir des concepts qui sont normalement utilisés pour décrire les items indexés" [1, p. 193].

Nous retenons que la personne qui fait l'indexation doit anticiper dans celle-ci le vocabulaire de l'utilisateur du CBR. Dans le cadre de la problématique de la mesure fonctionnelle, "la mesure est une mise en correspondance entre le monde empirique et le monde formel". Le vocabulaire est d'une part celui de la mesure fonctionnelle COSMIC-FFP et d'autre part celui

du cycle de vie du logiciel (ex. : SWEBOK). Ce vocabulaire se retrouve à travers les ontologies de chaque domaine. La connaissance des ontologies devrait aussi permettre une meilleure structuration de la base de cas si on peut catégoriser les cas problèmes.

Dans la construction d'un prototype, il est possible de répondre positivement à ces questions. Par exemple, il est possible de catégoriser et d'indexer les cas problèmes. Dans l'explication de la démarche du mesureur, nous utilisons le terme "concept topologique" pour catégoriser les cas problèmes. Pour chaque concept topologique on a :

- une définition ;
- sa relation avec un certain nombre de mots-clés ;
- sa relation avec un certain nombre de cas problèmes ;
- une explication de la raison du choix de ce concept.

Dans le cadre de la problématique de la mesure fonctionnelle, il y a donc une indexation basée sur le concept topologique qui est lié à des mots-clés qui conduisent aux cas problèmes.

Structure pour la représentation des cas dans la littérature

Il existe un grand nombre de systèmes à base de connaissances, tant dans les organisations que dans les universités. La structure de représentation des cas que l'on retrouve le plus souvent dans la littérature est par cadre non structuré et par "formulaire". De plus, en général, la représentation dans les CBR comprend : le problème, la solution et la justification de la solution.

Voici quelques exemples de systèmes à base de connaissances et le type de représentation des cas dans la littérature [1] :

- MEDIATOR utilise une représentation par cadre (*frame*). La représentation est très structurée, mais la solution n'est pas très élaborée. MEDIATOR met l'accent sur l'explication et la réparation des erreurs, la solution n'étant pas sa priorité ;
- CASEY utilise une représentation non structurée. À l'entrée, il reçoit beaucoup d'informations sur les symptômes et l'historique du patient. CASEY cherche à trouver des solutions à différents problèmes en passant d'un problème à l'autre sans recevoir de rétroaction. Pour cette raison, il ne fournit pas seulement la solution mais aussi comment il y est arrivé ;
- REMIND est une coquille qui propose des "formulaires" pour représenter les entrées, les champs calculés et les sorties. Pour une base de cas particulière, l'utilisateur peut visualiser les solutions et corriger ses erreurs avant de relancer le programme.

Dans le cadre de la problématique de la mesure fonctionnelle, le problème, la solution et la justification de la solution sont des éléments également importants. Il faut aussi prévoir une interaction entre le mesureur et le système à base de connaissances. Cette interaction doit être encadrée et structurée. Pour cette raison, nous croyons qu'une présentation sous forme de formulaire serait adéquate.

2.2.4.3 Technique de raisonnement à base de règles

La technique de raisonnement à base de règles est une des plus anciennes techniques pour représenter le domaine de la connaissance dans un système à base de connaissances.

Le raisonnement à base de règles (SI...ALORS RÈGLE) permet entre autres l'abduction, c'est-à-dire que l'on peut partir d'un résultat pour en retrouver les causes. Il permet de traiter des problèmes précis et connus des experts. Les règles sont formalisées par des experts et peuvent servir à résoudre plus d'un type de problème.

Dans le raisonnement à base de règles, la connaissance est présentée comme :

- des faits sur le monde ;
- des règles pour manipuler ces faits.

Il y a une complexité supplémentaire du fait qu'à n'importe quel moment, plus d'une règle peut s'appliquer. De plus, au moment où chaque règle s'applique, plusieurs autres peuvent aussi être applicables.

MYCIN est l'exemple le plus célèbre avec PROSPECTOR pour les techniques de raisonnement à base de règles. Dans le cadre de la problématique de la mesure fonctionnelle, nous avons choisi un raisonnement à base de règles semblable à celui de MYCIN. Nous ne croyons pas qu'il soit nécessaire d'élaborer plus amplement sur le raisonnement à base de règles puisque ce sujet a été traité en 2.2.2.4 (traitement de l'incertitude).

2.2.4.4 Justification de l'utilisation des deux techniques de raisonnement

En quoi l'utilisation des deux techniques de raisonnement se justifie-t-elle dans le cadre de la problématique de la mesure fonctionnelle ?

Raisonnement à base de cas versus raisonnement à base de règles

Dans la technique de raisonnement à base de cas on a une "bibliothèque" de cas "similaires". Dans la technique de raisonnement à base de règles, la connaissance est encodée dans des règles, à partir de la connaissance de l'expert.

Qu'est-ce qu'un cas "similaire" dans le cadre de la problématique de la mesure fonctionnelle COSMIC-FFP ? Il existe un certain nombre d'études de cas résolus par les experts COSMIC-FFP : Warehouse, Rice Cooker, Valve Control et Report Generator (Annexe H). On peut dire que l'étude de cas Rice Cooker est similaire à Valve Control car elles sont toutes les deux des logiciels en temps réels. Est-ce utile pour le mesureur dans le cadre de la problématique de la mesure fonctionnelle COSMIC-FFP ? Un logiciel est mesuré à partir de ses artefacts que l'on met en correspondance avec les concepts COSMIC-FFP (ex. : entrée, sortie, lecture, écriture, processus fonctionnel, frontière). La similarité des cas doit se situer au niveau des artefacts, sinon les cas seraient peu utiles au mesureur. C'est ce que nous appelons un "cas problème".

Il faut donc faire une distinction ici entre une étude de cas et un cas problème.

Étude de cas : plusieurs cas problèmes ayant une logique fonctionnelle du point de vue d'un utilisateur d'un logiciel.

Cas problème : une situation dans laquelle le mesureur doit utiliser un processus de diagnostic pour identifier correctement un concept de l'ontologie COSMIC-FFP (ex. : identifier un groupe de données ou encore identifier un processus).

Les cas (problèmes) qui sont présentés dans la littérature [1, 33, 44, 57] ne sont pas généralement liés à un seul concept, mais décrivent une situation qui peut évoluer dans le temps. Dans le cadre de la problématique de la mesure fonctionnelle COSMIC-FFP, un cas problème se doit d'être lié à un concept de l'ontologie COSMIC-FFP (2.1.11 a). Ces concepts sont importants pour l'indexation.

Est-ce que les cas problèmes de notre systèmes à base de connaissances rencontrent les prémisses de ce qui est défini comme un cas dans la littérature sur les CBR ?

Selon Kolodner, l'utilisation d'un CBR implique l'intuition que certaines situations reviennent avec régularité et peuvent être appliquées dans une situation similaire [1, p. 8-14]. Les cas problèmes de notre systèmes à base de connaissances reviennent avec régularité (ex. : il y a de nombreuses instances de processus fonctionnels dans les artefacts d'un logiciel) et peuvent être appliqués dans une situation similaire (ex. : la définition d'un processus fonctionnel ne change pas).

Est-ce que les règles de COSMIC-FFP peuvent être encodées ? Oui, ces règles sont facilement identifiables et peuvent être encodées. Par exemple, l'identification d'un processus fonctionnel requiert les règles suivantes [2, p. 37] :

- un processus fonctionnel est lié à au moins un besoin fonctionnel de l'utilisateur ;
- un processus fonctionnel est exécuté lorsqu'un événement déclencheur identifiable survient ;
- un processus fonctionnel contient au moins deux mouvements élémentaires de données, une entrée et une sortie ou une écriture ;
- un processus fonctionnel appartient à une et à une seule couche ;
- un processus fonctionnel se termine lorsqu'un point asynchrone est atteint. Un point asynchrone est atteint lorsque le mouvement de données final dans une séquence de mouvement de données n'est synchronisé avec aucun autre mouvement de données.

Cet exemple montre que le mesureur qui utilise la technique de raisonnement à base de règles peut retrouver la règle exacte pour les différents concepts de COSMIC-FFP [2]. Il est même possible de quantifier ces règles en demandant au mesureur si, pour son cas problème, la règle s'applique ou non.

Il est possible de construire plusieurs centaines de cas problèmes. Cependant, Kolodner [1] suggère de limiter leur nombre et de s'en tenir à ceux qui sont représentatifs des différents "problèmes" que peut rencontrer le mesureur. Par exemple, "ajouter un fournisseur" est un cas problème représentatif pour identifier un processus fonctionnel d'ajout dans un logiciel d'affaires. Par contre, on pourrait se demander quelle est la pertinence d'ajouter un cas

problème qui serait "ajouter un client" ou encore "ajouter un utilisateur" provenant du même logiciel ou d'un logiciel similaire. Pour un mesureur, le cas problème "ajouter un..." est toujours le même en autant que "fournisseur", "client" et "utilisateur" sont des entités de la base de données du même logiciel ou d'un logiciel très similaire.

Pour construire un prototype de système à base de connaissances, nous devons aussi modéliser les connaissances. C'est le sujet de notre prochaine sous-section.

2.2.5 Modélisation de la connaissance

2.2.5.1 Difficultés de la modélisation de la connaissance

Selon Jackson [33, p. 4], la productivité du transfert de la connaissance est habituellement relativement faible. Les raisons en sont les suivantes :

- le spécialiste a son propre jargon et il est difficile pour lui de le communiquer dans le langage de tous les jours ;
- les règles et les faits sous-jacents ne peuvent être caractérisés précisément, c'est-à-dire en utilisant, par exemple, des théories mathématiques ou des modèles déterministes ;
- les experts ont besoin de savoir plus qu'un certain nombre de règles ou de faits particuliers d'un domaine pour résoudre un problème. Par exemple, ils savent généralement quelles sortes d'informations sont pertinentes pour quelles sortes de jugements, la fiabilité des sources d'information, comment diviser un problème difficile en sous-problèmes pouvant avoir des solutions plus ou moins indépendantes. Obtenir ce type de connaissances, qui est normalement basé sur l'expérience personnelle plus que sur la formation formelle, est beaucoup plus difficile que d'obtenir des faits particuliers ou des règles générales ;
- enfin, l'expertise humaine, même relative à un domaine très étroit, est souvent liée à un contexte beaucoup plus large impliquant une bonne connaissance d'autres

domaines. Par exemple, l'application d'une méthode de mesure fonctionnelle pour obtenir la taille d'un logiciel est un domaine où l'expertise requise est très pointue. Cependant, cette tâche implique que le mesureur doit avoir non seulement une bonne connaissance des processus de développement en génie logiciel, mais aussi la capacité d'acquérir très rapidement une certaine connaissance des fonctions du logiciel qu'il mesure du point de vue de son utilisateur (par exemple : acquérir des connaissances sur le fonctionnement d'une banque pour des systèmes bancaires, etc.).

2.2.5.2 Questions sur la modélisation de la connaissance

En 1977, Faigenbaum, alors conférencier invité à l'*International Joint Conference on Artificial Intelligence*, faisait cette observation sur les systèmes experts :

"La puissance d'un système expert provient plutôt des connaissances qu'il possède que des formalismes particuliers ou des schémas d'inférence qu'il utilise" [46].

Dans cette perspective le transfert des connaissances est une part importante de la construction d'un système à base de connaissances.

Traditionnellement, le transfert de la connaissance est généralement accompli par une série d'entrevues entre l'ingénieur de la connaissance et l'expert du domaine en autant qu'il puisse articuler son expertise pour les fins du système à base de connaissances. Dans les dernières années, des préoccupations de "modélisation de la connaissance" se sont ajoutées [58, p. 2]. Donc, les processus d'acquisition de la connaissance ne sont plus vus comme le transfert de la connaissance dans un ordinateur, mais plutôt comme un processus de construction d'un modèle [58].

Quelles sont les raisons qui ont incité les chercheurs à passer d'une vue de "transfert de la connaissance à une vue de la modélisation de la connaissance" [58, p. 162] ? Est-ce que les méthodologies de résolution de problèmes en génie logiciel s'apparentent aux méthodologies de modélisation de la connaissance ?

À ces questions, Studer et al. [58, p. 162] répondent que le but de la discipline appelée "ingénierie de la connaissance" est similaire à celui de l'ingénierie du logiciel (ou génie logiciel) : faire en sorte que la construction d'un système de connaissances se fasse à partir d'une discipline rigoureuse plutôt que de demeurer un art.

Selon Studer, le passage d'une approche "transfert" à une approche "modélisation", ou passage d'une première génération de système expert à une deuxième génération de système expert, se serait fait en raison de difficultés de représenter adéquatement différents types de connaissances :

However, a careful analysis of the various rule knowledge bases showed that the rather simple representation formalism of production rules did not support an adequate representation of different types of knowled : e.g. in the MYCIN knowledge base strategic knowledge about the order in which goals should be achieved (e.g. "consider common causes of a disease first") is mixed up with domain specific knowledge about for example causes for a specific disease. This mixture of knowledge types, together with the lack of adequate justifications of the different rules, makes the maintenance of such knowledge bases very difficult and time consuming. Therefore, this transfer approach was only feasible for the development of small prototypical systems, but it failed to produce large, reliable and maintainable knowledge bases [58].

Nous retenons aussi de cette remarque que la modélisation étant un processus cyclique, de nouvelles observations peuvent conduire à des raffinements du modèle. Enfin, le modèle étant dépendant de l'interprétation de l'ingénieur de la connaissance, c'est le modèle qu'il faut évaluer à ses différentes étapes. La description détaillée des différentes étapes permettra de les évaluer et éventuellement (avec un plus grand nombre d'expérimentations) de les réviser. À cette fin, il existe des cadres de référence pour la modélisation de la connaissance (ex. : PROTÉGÉ-II [58, 59]). Ces cadres de référence suggèrent la création d'une ontologie du domaine de la connaissance.

2.2.5.3 Ingénierie des ontologies

La création d'une ontologie n'est qu'une des étapes de la modélisation de la connaissance (COMMON-KADS [60] et aussi Roy [61]). Il y a plusieurs définitions du mot ontologie. Une

définition d'ontologie se trouve dans Mizogouchi [62, p. 17 ss] et Uschold [63, p. 540 ss]. Pour Mizogouchi, "une ontologie consiste en une spécification explicite d'objets et de relations dans un domaine donné, permettant des échanges dans une communauté ainsi que la construction d'un modèle du domaine visé" [62, p. 3]⁹.

Plus récemment, le mot a été utilisé d'une façon plus large pour inclure des ensembles de concepts qui n'ont pas nécessairement une définition explicite et qui n'ont pas de relations clairement définies entre eux. Mizogouchi donne un exemple à partir de la classification hiérarchique des sujets de Yahoo, classification que l'on décrit comme étant une ontologie [62].

Uschold propose une définition minimale du terme ontologie qui contiendrait au minimum un vocabulaire et une indication de ce que signifie les mots dans le vocabulaire [63, p. 551]. On évacue alors la définition explicite des termes et les relations clairement définies entre ces termes.

2.2.5.4 Transfert et modélisation de la connaissance

Buchanan et al. [64] proposent une analyse de l'acquisition de la connaissance en termes de modèle de processus qui servira à construire un système expert. Les différentes étapes de Buchanan sont résumées par Jackson [33, p. 183-184] : identification, conceptualisation, formalisation, implantation et test. Les étapes de Buchanan ont été reprises dans un cadre de référence appelé KADS par Wielinga et al. [65]. Ces travaux ont inspiré ceux de van Heijst [60] et Schreiber avec CommonKADS [42] que nous utilisons dans notre recherche.

⁹ *An ontology is an explicit specification of objects and relations in the target world intended to share in a community and to use for building a model of the target world.*

Van Heijst et al. écrivent que différents éléments de connaissance jouent un rôle dans le raisonnement. Ils distinguent cinq différents types de connaissances et suggère un méta-modèle des connaissances [60, section 2.1] comprenant :

- les tâches : correspondent aux buts qui doivent être réalisés lors de la résolution du problème ;
- les méthodes de résolution de problèmes : ce sont les façons de réaliser les buts dans les tâches. Dans certains cadres de référence de la modélisation de la connaissance, les méthodes de résolution de problèmes définissent des sous-tâches pour lesquelles d'autres méthodes de résolution de problèmes peuvent être appliquées ;
- les inférences : décrivent les étapes du raisonnement dans le processus de résolution de problèmes. Les inférences sont aussi appelées des mécanismes d'inférences. Les modèles d'inférences peuvent être aussi appelés "structure d'inférence" ;
- les ontologies : décrivent la structure et le vocabulaire du domaine de la connaissance statique ;
- la connaissance du domaine : ensemble des énoncés tenus pour vrai sur le domaine.

Le but des tâches à réaliser pour van Heijst et al. [60] serait les tâches telles que définies par Hayes-Roth [66] (diagnostic, réparation, planification, etc.). Dans notre problématique de recherche, nous sommes à un niveau plus bas. Les tâches du mesureur sont des tâches de diagnostic. Ainsi, dans notre problématique de recherche, nous aurons les tâches suivantes : diagnostiquer un processus fonctionnel, une frontière, une sortie, etc. Van Heijst propose l'utilisation d'un grand nombre d'ontologies, un de ses objectifs étant la réutilisation des différentes théories des ontologies [60, *abstract*] qui peuvent ensuite être configurées dans une application ontologique¹⁰, alors que nous nous limitons à la mesure fonctionnelle COSMIC-FFP. La perspective est donc très différente à ce niveau.

¹⁰ Il ajoute: *This application ontology is then exploited to organize the knowledge acquisition process and to support computational design* (idem).

Ces types de connaissances sont directement inspirés de la méthodologie CommonKADS. Schreiber et al. définissent trois types de connaissances dans la méthodologie CommonKADS [42, p. 89] : connaissance des tâches, connaissance des inférences et connaissance du domaine. Notre compréhension de leur approche, par rapport à celle de van Heijst, est que la connaissance des tâches regroupe les tâches et les méthodes de résolution de problèmes, alors que la connaissance du domaine regroupe les ontologies et la connaissance du domaine. Van Heijst a seulement introduit des distinctions plus fines aux différents types de connaissances.

Nous reprendrons les différents types de connaissances suggérés par van Heijst [60] dans le Chapitre 4 en les adaptant aux domaines du génie logiciel, de la mesure fonctionnelle COSMIC-FFP et de la démarche du mesureur.

2.2.6 Vérification des modèles de connaissances

Dans le Chapitre 6, nous traitons de la vérification du prototype 2. Nous avons constaté que, dans le domaine de l'ingénierie de la connaissance, les auteurs utilisent les définitions classiques du génie logiciel pour la vérification, définitions qui ne vont pas sans comparer vérification à validation.

Preece [67] considère que la vérification et la validation (V et V) dans le domaine de l'ingénierie de la connaissance est une instance du génie logiciel :

Knowledge engineering can be viewed as a special instance of software engineering [67, p. 2].

Boehm écrivait en 1984 :

Verification is building the system right. Validation is building the right system [68].

Coenen [69] traite de la question de la vérification et de la validation des bases de connaissances en la comparant aux approches déjà utilisées en génie logiciel avec les bases de données des systèmes à base de règles. Pour Coenen [69, p. 4], une base de données peut avoir un problème structurel d'inférence (ex. : connexion entre les règles) et de design (ex. : duplication, incohérence), ce qui correspond aux concepts de vérification et de validation tels que définis plus haut.

Pour Coenen, une des approches pour valider un système à base de règles est de demander à un expert du domaine si le résultat est conforme à ce qui devrait être normalement [69, p. 5].

Pour Coenen, on doit tenir compte du type d'erreurs lors de la vérification. Il écrit que la façon la plus directe de structurer les erreurs et les anomalies est de faire une distinction entre les erreurs sur les inférences et celles sur le design de la "base de règles" [69, p. 4].

Dans le cas des erreurs d'inférence, il peut s'agir d'un problème de connexions entre les règles, alors que dans l'autre cas (architecture) il s'agit, par exemple, d'incohérences et de duplications. Il y aurait une erreur d'inférence si un cas problème n'était pas relié à au moins un thème. Il y aurait une erreur de design si le même cas problème était deux fois dans la base de connaissances.

Il faut noter que dans notre approche de vérification, l'aspect architectural a été traité par la définition d'un certain nombre de structures de fichiers (XSD), ce qui a permis de réduire les duplications de fichiers et d'éviter les incohérences structurelles.

2.2.7 Originalité de l'approche cognitive choisie et conclusion

L'originalité de notre approche cognitive se trouve entre autres dans l'exploitation du lien entre l'ontologie de la méthode de mesure COSMIC-FFP et le cycle de vie d'un logiciel. Comment ? En proposant la mise en relation du vocabulaire du cycle de vie du logiciel et des concepts de la méthode de mesure COSMIC-FFP (concepts topologiques). Dans le Chapitre 4, dans le cadre de la modélisation de notre système à base de connaissances, nous présentons

l'ontologie COSMIC-FFP. Nous avons aussi commencé à nous interroger sur la création d'une ontologie pour le cycle de vie du logiciel SWEBOK [70].

À partir des concepts topologiques, il est possible de trouver les cas problèmes qui conviennent le mieux à une situation. C'est l'application du raisonnement à base de cas. Lorsque les cas problèmes ont été identifiés, il est possible pour le mesureur de se référer à des règles pour trouver la solution. C'est l'application du raisonnement à base de règles.

Autrement dit, notre originalité est au niveau de l'utilisation d'une approche cognitive hybride pour permettre aux mesureurs l'application des règles de mesure COSMIC-FFP en s'inspirant de l'ontologie COSMIC-FFP pour l'indexation et éventuellement de l'ontologie SWEBOK.

Notre revue de littérature dans le domaine cognitif nous amène à conclure que les raisonnements à base de cas et à base de règles peuvent nous aider à résoudre notre problématique de la mesure.

Dans la prochaine section, nous présentons notre problématique générale et nos objectifs de recherche.

2.3 Problématiques et objectifs de la recherche

En conclusion à cette revue de littérature, nous présentons deux niveaux de problématique, une pour l'expert en général, et une autre pour le mesureur qui, dans le cadre de la problématique de la mesure fonctionnelle COSMIC-FFP, doivent appliquer leurs connaissances. Ceci nous amène à proposer une approche cognitive à cette problématique et à formuler notre objectif de recherche.

En général, la problématique de l'expert peut se présenter de la façon suivante :

- les experts ont besoin de connaître plus qu'un certain nombre de règles ou de faits particuliers à un domaine pour résoudre un problème ;

- l'expertise humaine, même relative à un domaine très étroit, est souvent liée à un contexte beaucoup plus large impliquant une bonne connaissance d'autres domaines. La diversité des méthodes de développement et de maintenance, et aussi l'utilisation d'un vocabulaire dont les mêmes termes ont une signification différente selon l'entreprise, rendent cette tâche encore plus difficile.

Dans le cadre de l'application de la mesure fonctionnelle COSMIC-FFP, la problématique est la suivante :

- il n'existe pas de méthodes systématiques et reconnues pour réaliser la mise en correspondance des règles formelles avec les différents environnements de développement (monde empirique).
- les connaissances d'experts en mesure, connaissances acquises après plusieurs années de pratique, reflètent deux ensembles d'habiletés ou de connaissances nécessaires au mesureur :
- la compréhension des artefacts du logiciel pour situer le problème (ou cas problème) à résoudre (comprendre et interpréter) ;
- la compréhension des règles de mesure pour les appliquer aux cas problèmes à résoudre (utiliser et solutionner).

Notre recherche tient compte de ces deux problématiques. Nous allons utiliser un certain nombre d'études de cas pour créer des "cas problèmes". Ces études de cas sont une partie de la réalité du monde du logiciel qui est un contexte plus large que seulement le domaine de la mesure fonctionnelle. Nous devons aussi nous assurer que les concepts utilisés dans les études de cas ont une signification correspondante au concept utilisé dans COSMIC-FFP. Par exemple, le terme "entité" ne correspond pas nécessairement au concept de "groupe de données" dans COSMIC-FFP, même s'il est le plus souvent synonyme.

Dans notre recherche, nous traitons aussi de la connaissance des règles de mesure pour les appliquer aux cas problèmes à résoudre. Dans le "cheminement du mesureur", ceci consiste à

mieux comprendre les étapes relatives à l'utilisation des règles et à trouver la bonne solution (ou réponse) aux "cas problèmes" lorsque ceux-ci ont bien été compris et interprétés.

Globalement, notre objectif de recherche est :

- d'encapsuler et de structurer la démarche des experts (Chapitre 4), et
- de fournir aux mesureurs des règles et des recommandations qui permettent, en tenant compte de l'expérience du mesureur et de la qualité des informations obtenues, d'améliorer les résultats des mesures (Chapitre 5).

-

Le prochain chapitre explique notre démarche pour réaliser notre objectif de recherche.

3. MÉTHODOLOGIE DE RECHERCHE

Nous avons développé une méthodologie de recherche afin de répondre aux différentes questions de la problématique de recherche. Cette méthodologie comprend plusieurs étapes (2.5) illustrées à la Figure 3-1. La première partie de la méthodologie (partie gauche de la Figure 3-1) consiste à trouver une solution théorique à notre objectif d'encapsuler et de structurer la démarche des experts en mesure fonctionnelle. La deuxième partie de la méthodologie (partie droite de la Figure 3-1) consiste à fournir aux mesureurs des règles et des recommandations qui devraient lui permettre d'améliorer les résultats des mesures, ce qui se traduira par la création de deux prototypes.

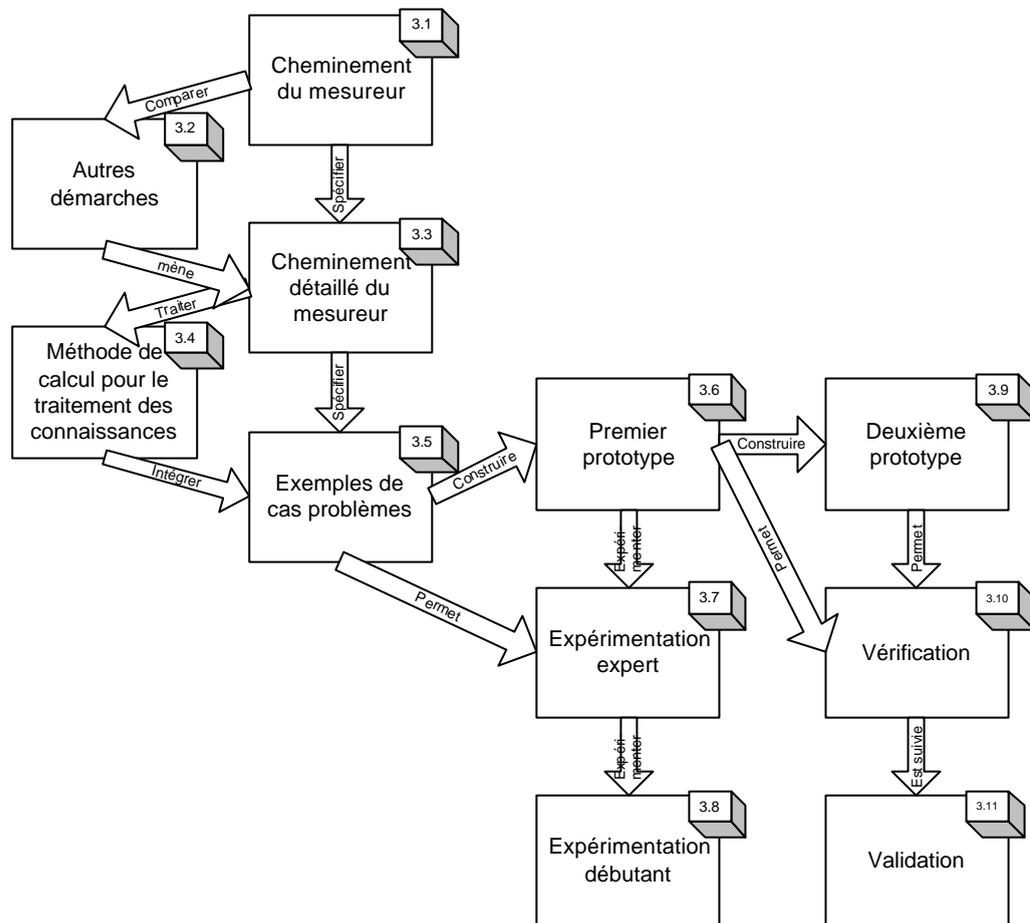


Figure 3-1 : Étapes de la méthodologie de recherche

3.1 Cheminement du mesureur

En tant que praticien de la mesure fonctionnelle pendant plus de 15 ans, nous avons pu observer le comportement de plusieurs mesureurs et les questionner sur les raisons qui les incitaient à privilégier une solution plutôt qu'une autre. Nous avons retenu que les mesureurs avaient à prendre de nombreuses décisions en rapport avec l'interprétation des règles de la mesure et à arriver, selon les artefacts en leur possession, à choisir une solution pour mesurer les logiciels. Dans un premier temps, nous voulons mettre en évidence les mécanismes de

raisonnement du mesureur pour retracer le cheminement théorique de ce raisonnement. Pour retracer le cheminement de raisonnement du mesureur, il nous faut d'abord comparer nos observations sur l'application de la méthode de mesure fonctionnelle, les analyser et les modéliser.

Les sources d'observations sur l'application de la méthode de mesure fonctionnelle sont les suivantes :

- les règles formelles de la mesure (Manuel COSMIC-FFP) ;
- les différentes interprétations des règles formelles écrites par les experts dans des documents appelés "règles locales" ;
- notre propre connaissance en tant qu'expert de la mesure fonctionnelle qu'il faut comparer avec les connaissances des autres experts ;
- des discussions sur les règles de la méthode de mesure{ XE "règles" } avec les experts à travers le monde, dans le cadre de nos interventions en industrie, ou de forums de travail (*workshop*) ;
- les outils de diagnostic (Help!CPR) ;
- les études de cas.

L'analyse de ces sources d'observation, accumulées progressivement et ayant exigé plusieurs années de travail, devrait nous permettre de mettre en évidence les mécanismes de raisonnement qui permettent au mesureur d'interpréter un artefact pour ensuite arriver à une solution. Dans la pratique, le mesureur peut faire toutes ces opérations en quelques secondes, ce qui peut rendre difficiles nos analyses sur la base de nos observations du mesureur. La lecture des "règles locales" permet de mieux analyser le raisonnement, mais est-ce bien le raisonnement idéal d'un mesureur expert ? C'est pour cette raison que nous allons aussi devoir utiliser notre expérience personnelle et nous demander : est-ce bien de cette façon que je raisonne en tant qu'expert ? Cette réflexion est alors enrichie par des discussions avec des experts de la mesure.

La formalisation du cheminement idéal du mesureur sera l'aboutissement de nos observations et de nos analyses. Nous allons aussi nous inspirer de la littérature et, entre autres, de la démarche des chercheurs pour résoudre un problème. Notre méthode consistera à vérifier, à partir de différentes propositions des chercheurs dans le domaine de la mesure fonctionnelle, comment le raisonnement du mesureur s'y inscrit. Arriver à la modélisation du raisonnement de l'expert est un travail de réflexion à partir d'essais et d'erreurs en utilisant différents modèles connus. L'analyse d'un outil de diagnostic (Help!CPR) en tant que modèle de la démarche d'un expert qui doit poser un diagnostic est aussi une partie importante de nos observations.

La description du raisonnement du mesureur sera l'actualisation de nos réflexions sur les différents modèles. Il nous faudra décrire les différentes étapes du cheminement du mesureur à partir des informations en notre possession. Ce processus est itératif. Voici des exemples de questions pour chaque itération :

- est-ce que le raisonnement décrit correspond à mon expérience ?
- est-ce qu'un autre expert pourrait vérifier ce raisonnement et être d'accord ?
- est-ce que je puis me servir de la démarche pour résoudre un problème de mesure ?

3.2 Autres contributions théoriques

Nous aurons aussi à comparer le cheminement du mesureur à d'autres démarches. Les démarches que nous utiliserons sont celles de COSMIC-FFP [2] et de Jacquet et Abran [29].

Notre thèse porte sur la mesure fonctionnelle COSMIC-FFP. La comparaison du cheminement du mesureur avec les étapes de la mesure fonctionnelle proposée dans le Manuel de mesure COSMIC-FFP [2] fait partie de nos préoccupations de recherche.

La comparaison des étapes de l'application de la mesure fonctionnelle proposée par Jacquet et Abran [29] avec les étapes du cheminement du mesureur font aussi partie de nos préoccupations de recherche puisque ce sont ces étapes qui ont inspiré notre sujet de thèse.

3.3 Cheminement détaillé

Le cheminement du mesureur sera détaillé (Figure 3-2). Les différentes étapes (2.5) sont de :

- décrire les tâches du mesureur à travers la méthode COSMIC-FFP ou les tâches de la méthode ;
- adapter les types de connaissances aux tâches de la méthode ;
- définir en quoi consiste la connaissance du domaine (ingénierie du logiciel et COSMIC-FFP) ou encore définir l'ontologie du domaine ;
- décrire les étapes et les sous-étapes de la méthode de diagnostic.

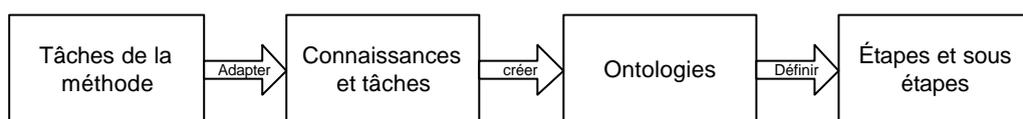


Figure 3-2 : Cheminement détaillé

Pour décrire les tâches de la méthode nous devons construire la carte conceptuelle de COSMIC-FFP, de la même façon que nous avons construit la carte conceptuelle de la méthode IFPUG [13, p. 261]. Ceci nous permettra de faire ressortir les différents concepts qui doivent être diagnostiqués par le mesureur.

L'adaptation des types de connaissances aux tâches du mesureur consiste à mettre en correspondance la tâche à exécuter et les connaissances nécessaires pour exécuter cette tâche. Par exemple, la tâche peut consister à reconnaître un processus fonctionnel et ce qu'il est nécessaire de connaître, entre autres la définition d'un processus fonctionnel, ses principes et ses règles (voir note 2). Il peut y avoir plusieurs niveaux de connaissances et la détermination de ces niveaux de connaissances fait aussi partie de notre recherche. Des types de connaissances sont proposés dans la littérature du domaine cognitif, tels ceux de van Heijst et de Schreiber et Studer [58, 60]. L'adaptation des types de connaissances aux tâches du mesureur devrait nous permettre de proposer une procédure d'application.

Le mesureur utilise les connaissances du domaine de l'ingénierie du logiciel et plus spécifiquement les connaissances de COSMIC-FFP. La littérature du domaine indique que, pour la modélisation des connaissances, la création d'une ontologie est une étape nécessaire. Nous allons donc faire une première exploration des ontologies des domaines étudiés. Il s'agit ici des domaines de l'ingénierie du logiciel et de la méthode de mesure COSMIC-FFP. Nous allons examiner une méthodologie particulière en ce qui concerne le domaine de l'ingénierie, soit le corpus de connaissances de SWEBOK et trouver des pistes de recherche. Nous allons examiner la méthode de mesure COSMIC-FFP (Figure 1-3) pour déterminer si ce modèle a sa propre ontologie.

Nous avons constaté, à partir de notre revue de littérature, que le mesureur posait un diagnostic. Nous avons aussi écrit qu'il s'inspirait à la fois des systèmes à base de cas (CBR) et aussi des systèmes à base de règles. Nous allons définir les étapes et les sous-étapes de la méthode de diagnostic en tenant compte de ces types de systèmes, des connaissances et des tâches du mesureur ainsi que du cheminement du mesureur.

3.4 Traitement des connaissances

Les systèmes à base de cas et de règles impliquent des méthodes de calculs. Quelles seront ces méthodes de calcul ? Dans la revue de littérature, nous avons retrouvé un certain nombre de ces méthodes, dont la théorie de la certitude. Dans notre recherche, nous allons expérimenter ces méthodes en utilisant des cas provenant du domaine de la mesure fonctionnelle (2.5).

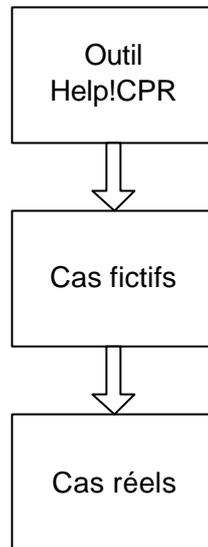


Figure 3-3 : Traitement des connaissances

Nous allons le faire en trois étapes (Figure 3-3) : avec un outil existant et ses propres méthodes et formules, en utilisant nos méthodes et nos formules avec plusieurs simulations fictives, en utilisant nos méthodes et nos formules avec des cas se rapprochant de la réalité du mesureur. Voici la description de ces étapes :

- la première étape consistera à utiliser un outil déjà décrit dans notre revue de littérature soit Help!CPR pour vérifier si les résultats sont conformes à ceux attendus. Help!CPR a ses propres méthodes et formules ;
- la deuxième étape consistera à utiliser nos méthodes avec différentes formules à partir d'un tableur de type Excel. Dans cette vérification, nous allons modifier les formules et les entrées de données pour chaque formule. Le but est de simuler différents résultats afin de vérifier comment seront affectés les résultats à partir de méthodes, de formules et de données. Ceci nous permettra de choisir les méthodes et les formules les plus appropriées aux objectifs de traitement de la connaissance ;

- la troisième étape consistera à utiliser ces formules et quelques cas réels pour trouver la meilleure combinaison possible des pourcentages pour en arriver aux recommandations souhaitées.

3.5 Design d'exemples de cas problèmes

Le design d'exemple de cas problèmes demande que l'on consulte des experts et des associations reconnues à travers le monde afin de savoir comment on définit et on traite les cas problèmes. C'est ce qui devrait nous permettre de créer des cas problèmes (une trentaine serait suffisant pour débiter) en nous inspirant des façons de faire des experts, du cheminement détaillé du mesureur et des formules de traitement des connaissances retenues.

- le choix des cas problèmes s'inspirera des cas problèmes "types" reconnus par les experts ;
- le design de ces cas{ XE "cas" } problèmes sera réalisé en nous inspirant des définitions et des principes{ XE "principes" } de COSMIC-FFP{ XE "COSMIC-FFP" }, ainsi que des "règles{ XE "règles" } locales" et de notre expérience ;
- chaque cas problème devra permettre de vérifier le cheminement du mesureur, à partir de la recherche du cas problème approprié, jusqu'à sa solution ;
- la documentation de chaque étape devra être définie précisément et suivre un modèle qui pourra se répéter pour chaque cas problème. Par exemple, pour le cas problème, il nous faudra définir les différentes parties de sa documentation, soit une problématique, une mise en contexte, etc.

3.6 Design d'un premier prototype

La construction d'un prototype est nécessaire pour expérimenter notre approche d'un système à base de connaissances de la mesure fonctionnelle COSMIC-FFP. La méthode de construction du prototype sera basée sur la méthodologie de développement UML [71] :

- construction d'un diagramme de cas d'utilisation ;
- description des scénarios des différents agents ;
- spécifications, design du logiciel et inspections ;
- construction du prototype ;
- tests unitaires et de systèmes ;
- implantation.

Pour chacune des étapes décrites plus haut, il faut prévoir de nombreuses consultations auprès de différents experts. Uniquement pour les inspections des spécifications et du design, il faudra prévoir la collaboration de plusieurs personnes et une mise à niveau de ces personnes sur la nature du projet.

Il faudra ajouter à ces étapes, celles nécessaires à la définition de la structure des fichiers de référence pour la base de connaissances. Il faudra prévoir différents types de références et une description de la structure de chaque type de référence.

3.7 Vérification auprès des experts

Notre objectif est d'obtenir, avec un jugement le moins biaisé possible, des opinions sur les recommandations produites par le système de connaissances par des experts. Pour ce faire, nous définirons et nous utiliserons une trentaine de cas problèmes. L'expert utilisera le prototype pour examiner ces cas problèmes et faire ses commentaires. Une procédure de vérification sera présentée aux experts.

Les activités de cette vérification seront les suivantes :

- sélectionner un certain nombre d'experts sur la base de leur expérience de la mesure COSMIC-FFP, de la mesure fonctionnelle et de leur disponibilité ;
- demander aux experts d'utiliser le système de connaissances pour chaque cas problème ;

- demander aux experts de répondre de différentes façons aux thèmes ;
- demander aux experts s'ils sont d'accord avec les recommandations produites par le système de connaissances ;
- assurer le suivi auprès des experts ;
- inscrire les résultats par cas problème dans un tableur ;
- compiler les résultats reçus par tous les experts.

Pour cette vérification, nous avons besoin d'un nombre restreint d'experts du domaine de la mesure fonctionnelle. Ces experts peuvent provenir de différents pays (Annexe E). Il nous faudra faire leur suivi sur les quelques mois que dure l'expérience. Ces experts devront avoir une connaissance{ XE "connaissance" } théorique des règles{ XE "règles" } de la méthode de mesure fonctionnelle ainsi qu'une expérience pratique de cette même mesure.

3.8 Expérimentation avec les débutants

Nous faisons l'hypothèse que les mesureurs qui utiliseront notre méthode pour mesurer un logiciel à partir de sa documentation, sur la base de la méthode de mesure fonctionnelle COSMIC-FFP, auront de meilleurs résultats¹¹ que ceux qui utilisent seulement le manuel de mesure COSMIC-FFP.

Pour l'expérimentation avec les débutants, il nous faudra définir les étapes de l'expérimentation, les connaissances et les expériences des mesureurs, la composition des groupes et les caractéristiques de formation (Figure 3-4).

11 La référence est le résultat fourni par les experts pour les cas prévus.

Étapes de l'expérimentation	Connaissances et expériences
Composition des groupes	Caractéristiques de la formation

Figure 3-4 : Expérimentation avec les débutants

3.8.1.1 Étapes de l'expérimentation

Voici les étapes de l'expérimentation :

- choix de deux groupes de mesureurs, un groupe qui utilisera le prototype et un groupe qui n'aura que le manuel de mesure COSMIC-FFP ;
- formation sommaire à la mesure COSMIC-FFP pour les deux groupes ;
- mesure d'un logiciel (ou partie de logiciel) par les mesureurs ;
- analyse des résultats.

3.8.1.2 Connaissances et expériences des mesureurs

Pour les mesureurs :

- ne pas avoir d'expérience (ou très peu) dans la mesure fonctionnelle, aussi bien celle de COSMIC-FFP que les autres mesures fonctionnelles sur le marché ;
- avoir leur première formation COSMIC-FFP le matin même de l'expérimentation ;

- avoir des mesureurs ayant à peu près le même niveau de connaissance en génie logiciel.

3.8.1.3 Composition des groupes

Cette expérimentation devrait être réalisée deux fois avec des mesureurs débutants afin de nous assurer que les résultats de la première expérimentation ne sont pas dus au hasard.

La composition des groupes devra suivre les critères suivants :

- le nombre idéal pour chaque groupe devrait être de cinq à dix mesureurs, ce qui veut dire de 10 à 20 mesureurs pour chaque expérience ;
- les mesureurs ne sauront pas s'ils auront accès ou pas au système de connaissances avant le début de la mesure ;
- la méthode aléatoire pour l'assignation des mesureurs ne devrait être connue qu'au début de la séance de mesure (ex. : choix par ordre alphabétique des noms des personnes).

3.8.1.4 Caractéristiques de la formation :

- la formation devra être dispensée aux deux groupes en même temps ;
- sa durée sera d'environ trois heures ;
- le but de la formation sera de permettre aux participants de comprendre la démarche, les définitions et les principes{ XE "principes" } COSMIC-FFP{ XE "COSMIC-FFP" } ;
- idéalement les mesureurs devraient déjà connaître la démarche et les définitions du processus de développement du logiciel ;
- la formation devra être dispensée par un expert de la mesure fonctionnelle ;
- la formation devra aussi comprendre une formation sur le prototype pour tous les participants.

3.9 Construction du deuxième prototype

Nous croyons nécessaire la construction d'un deuxième prototype pour permettre la vérification du système à base de connaissances. Pour mieux évaluer les mécanismes de vérification du deuxième prototype, nous prévoyons passer d'une trentaine à une centaine de cas, ce qui exige quelques mois de travail supplémentaire. Idéalement, la construction du deuxième prototype devrait permettre de rendre disponible le système de connaissances sur Internet pour des expérimentations plus poussées auprès des mesureurs et des experts.

Le deuxième prototype devra d'abord être construit en tenant compte des objectifs de vérification avec plus d'une centaine de cas. Dans un deuxième temps, si les objectifs ne sont pas conflictuels, voir à le construire de façon à ce qu'il soit disponible sur Internet. Rendre disponible le prototype sur Internet même si les expérimentations plus poussées ne font pas partie de notre recherche.

Les fonctionnalités du deuxième prototype devraient demeurer les mêmes pour le mesureur, l'expert et l'administrateur. La définition des cas d'utilisation et des exigences ne devrait donc pas changer. Par contre, il faut prévoir quelques mois pour réaliser une réingénierie du prototype au niveau de l'accès à la base de connaissances. Cette réingénierie sera, entre autres, rendue nécessaire en fonction des besoins de vérification.

3.10 Vérification

La vérification consiste à s'assurer que l'on fait les choses correctement (*doing the things right*) [68]. La vérification que nous voulons faire sera appliquée à la base de connaissances et non au prototype en tant que logiciel.

Notre revue de littérature a fait ressortir que la vérification du système de connaissances pourrait être facilitée par une meilleure structuration des fichiers de référence des bases de connaissance [69]. Cette structuration pourrait nous amener à mieux réutiliser les différents fichiers et aussi à diminuer le nombre de fichiers utilisés.

Lorsqu'une base de connaissances est composée d'un grand nombre de fichiers, il faut revoir leur composition pour réduire les redondances de la documentation et faciliter la vérification de la qualité des références. Nous allons regarder cette question en considérant deux possibilités :

- la documentation de référence pourrait être considérée comme une source indivisible de données (un seul fichier) et chaque type de document a une et une seule vue ;
- la documentation de référence pourrait aussi permettre une visualisation des références à partir de plusieurs fichiers tout en permettant une seule vue virtuelle par type de document.

La réalisation de ces possibilités se fera à partir de l'examen des normes de structuration des bases de connaissances et plus spécifiquement celles proposées par le W3C¹². Du point de vue contenu, il nous faut aussi respecter la norme COSMIC-FFP.

Nous allons donc vérifier :

- si tous les fichiers respectent les normes de structuration (W3C ou autre) ;
- si pour chaque concept nous respectons la définition fournie dans l'ontologie de COSMIC-FFP ;
- la syntaxe et la grammaire des fichiers.

Nous prévoyons qu'avec une centaine de cas il y aura près de 1 000 fichiers à vérifier et à transformer selon les normes que nous aurons choisies.

3.11 Validation

La validation consiste à s'assurer que l'on a le bon produit (*the right product*) [68].

12 <http://www.w3.org/>

La démonstration de la validation est subjective. La preuve de la validation vient du client (obtenir le bon produit) à travers sa propre subjectivité. Pour commencer, il nous faudra définir qui est le client de notre démarche et plus particulièrement dans le contexte du prototype 2. Sont-ce les experts ? Les mesureurs sur le terrain ? Les apprentis mesureurs ? Les éducateurs ? Nous voulons, idéalement, fournir un "bon produit" (*the right product*) à toutes ces personnes. Dans cette perspective, nous devrions rejoindre toutes ces catégories de clients et leur demander si l'on a construit le bon produit.

Nous allons définir ce que nous voulons valider. Voici quelques éléments à considérer :

- ce sera nécessairement un sous-ensemble des utilisateurs potentiels, probablement le mesureur débutant ;
- la validation devrait pouvoir être réalisée de façon objective ;

La validation pourrait, par exemple, être réalisée sur les recommandations du système à base de connaissances. Compte tenu de ce que nous avons mentionné plus haut, il nous faudrait alors définir le "client" (probablement un expert de la mesure fonctionnelle COSMIC-FFP) et réaliser une expérimentation sur les recommandations du système à base de connaissances.

4. FORMALISATION DE LA DÉMARCHE DU MESUREUR

4.1 Cheminement du mesureur et problématique de la mesure

Dans la phase 2 (application de la mesure) du processus de mesure de Jacquet et Abran [29] les activités 1 et 2 ont pour but de permettre aux mesureurs d'interpréter de façon cohérente (selon la définition ISO [14]) les règles de la méthode de mesure en tenant compte de la qualité et de la disponibilité de la documentation pour le logiciel à mesurer. Ces activités sont des activités d'expertise en vue d'une modélisation appropriée des connaissances [11]. Pour en arriver à la modélisation, il faut d'abord bien comprendre les différentes étapes du cheminement du mesureur.

Du point de vue du mesureur, appliquer les règles{ XE "règles" } de méthode de mesure signifie qu'il doit solutionner le "problème" spécifique de mesure qui se présente à lui. Pour en arriver à la solution (soit une assignation numérique), le mesureur doit comprendre les artefacts du logiciel à mesurer, interpréter ce qu'il comprend afin d'identifier ce qui est à mesurer et enfin utiliser les règles de la méthode de mesure et les règles d'assignations numériques.

La Figure 4-1 montre notre modélisation du cheminement du mesureur [11] :

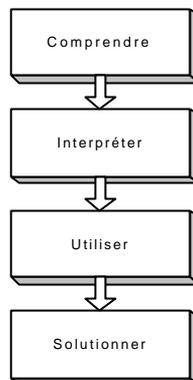


Figure 4-1 : Modèle du cheminement du mesureur

Le cheminement du mesureur comprend quatre étapes. Voici la description de chacune de ces étapes.

4.1.1 Comprendre

Quel est le problème du mesureur ? Le mesureur doit identifier un concept de la méthode de mesure COSMIC-FFP. Le mesureur doit identifier ces concepts à travers les artefacts du logiciel. Par exemple, un artefact du logiciel consiste en un modèle entité/relation. Le mesureur doit comprendre qu'une entité de ce modèle le pourrait correspondre au concept de groupe de données dans la méthode de mesure COSMIC-FFP.

Pour comprendre qu'il y a une relation entre le modèle entité/relation et le concept de groupe de données de la méthode de mesure COSMIC-FFP, il est nécessaire que le mesureur ait une certaine expérience et des connaissances. Tel que mentionné dans la problématique, l'expertise humaine, même relative à un domaine très étroit, est souvent liée à un contexte beaucoup plus large impliquant une bonne connaissance d'autres domaines. La Figure 4-2 montre que le mesureur doit avoir une compréhension de l'ingénierie du logiciel, du logiciel et de la méthode de mesure. Le mesureur doit comprendre suffisamment le domaine de l'ingénierie du logiciel (ex. : ce qu'est un modèle entité/relation), le logiciel à mesurer (ex. :

où se trouve le modèle dans le logiciel à mesurer) et la méthode de mesure fonctionnelle COSMIC-FFP (ex. : lien entre groupe de données et modèle entité/relation). C'est ce qui lui permettra de trouver le "mot-clé" qui déclenchera la recherche. Ici le mot-clé devrait être le concept "groupe de données" ou éventuellement le mot "entité" ou "relation".

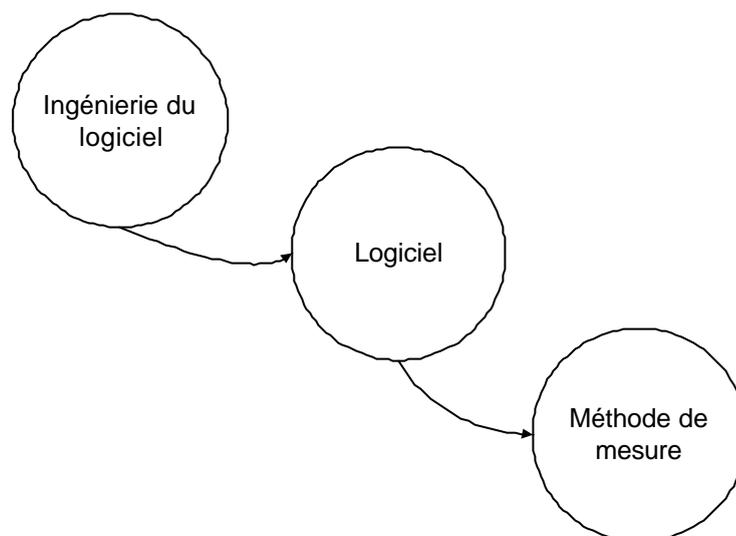


Figure 4-2 : La compréhension du mesureur à trois niveaux d'abstraction

4.1.2 Interpréter

Le mesureur doit interpréter la relation entre les artefacts mesurés et les concepts COSMIC-FFP à partir de sa compréhension du cas problème. Par exemple, il comprend déjà qu'il y a une relation entre le modèle entité/relation et les groupes de données. Il devra alors interpréter à quel type de cas problème appartient le modèle entité/relation mesuré.

Le contexte de la mesure fonctionnelle est très complexe à savoir qu'il existe plusieurs domaines d'application en génie logiciel (administration et gestion, temps réel, embarqués, etc. [8]). De plus, chaque logiciel est différent avec, le plus souvent, de nombreux composants aussi très différents. Cependant, les règles{ XE "règles" } de la méthode de mesure COSMIC-FFP{ XE "COSMIC-FFP" } sont exprimées par des concepts uniques et

structurés (Figure 4-3). Il faut donc, au mesureur à partir de ces cas particuliers et unique, retrouver les concepts génériques du modèle de mesure COSMIC-FFP.

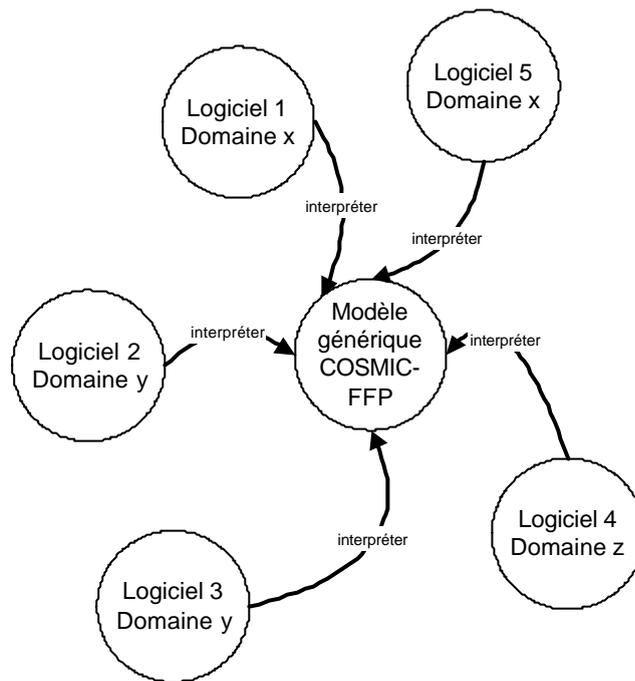


Figure 4-3 : Logiciels et COSMIC-FFP (interpréter)

4.1.3 Utiliser

Il s'agit ici d'utiliser les règles{ XE "règles" } de la méthode de mesure du logiciel. Par exemple, pour identifier un groupe de données, il est requis d'utiliser les règles d'identification d'un groupe de données. Le mesureur{ XE "mesureur" } devra utiliser les règles de la méthode de mesure pertinentes au cas{ XE "cas" } problème à résoudre.

Il est important que le mesureur se pose les bonnes questions (thèmes) en rapport avec le problème particulier à résoudre. Il peut retrouver ces thèmes dans le Manuel de mesure COSMIC-FFP, ainsi que toutes les définitions qui l'aideront, par exemple, à identifier correctement un groupe de données. Cette démarche avec le manuel peut prendre un certain

temps et le mesureur ne peut être certain qu'il a considéré toutes les alternatives, sauf s'il connaît très bien toutes les règles de la méthode de mesure{ XE "règles" } qui doivent être utilisées pour ce problème particulier.

4.1.4 Solutionner

Solutionner le problème, c'est-à-dire identifier, par exemple, si l'entité du logiciel mesuré est bien un groupe de données. La solution consiste à trouver ce que l'expert considère comme étant le meilleur arrimage possible au problème. Le problème, dans un système de connaissances, se traduit par un cas problème. Le mesureur{ XE "mesureur" }, à partir de son expérience ou à partir d'une recommandation d'un système de connaissances, confirme ou infirme les solutions possibles au problème. L'utilisation des règles{ XE "règles" } de la méthode de mesure n'est donc pas suffisante pour résoudre le problème. Il faut aussi que le mesureur choisisse la bonne solution.

4.1.5 Exemples

Voici deux exemples simples illustrant le cheminement du mesureur.

La documentation fournit un modèle entité/relation où le mesureur peut visualiser dix entités et deux relations. Il comprend que ce modèle (cet artefact) peut l'aider à déterminer le nombre de groupes de données¹³ dans le logiciel qu'il mesure. Il doit interpréter ce que relations et entités signifient dans le contexte des règles{ XE "règles" } de la méthode de mesure. Il doit ensuite utiliser les règles de la méthode de mesure relatives aux groupes de données. La solution pourrait être qu'il y a huit groupes de données, ce qui demande à être confirmé ou infirmé par l'expert ou le système de connaissances (recommandations). C'est

13 Identifier un groupe de données est un cas problème.

par ce cheminement que le mesureur peut résoudre un "problème" spécifique d'application de la mesure.

Avec le deuxième exemple, nous présentons le cheminement du mesureur à partir de la formulation des exigences d'un utilisateur pour un logiciel.

Selon les exigences¹⁴, l'utilisateur demande de pouvoir interroger une base de données fournisseur pour obtenir la liste des articles achetés dans le dernier mois pour un fournisseur précis. Pour le mesureur, la mesure de cette demande implique (à partir des données disponibles) une connaissance ou une compréhension :

- du mode de déclenchement du processus ;
- des résultats du processus ainsi que du (ou des) groupe(s) de données lues ;
- des validations et des résultats possibles des validations.

Avec cette information, qu'il peut obtenir par la documentation ou en interrogeant le spécialiste du logiciel à mesurer, il applique les règles{ XE "règles" } de la méthode de mesure.

On pourrait croire que l'application des règles{ XE "règles" } de la méthode de mesure peut se faire facilement par la construction d'un arbre de décision. Cependant, en essayant de vérifier toutes les possibilités, leur nombre peut augmenter de façon exponentielle. Dans ce contexte, l'utilisation d'heuristiques [72, p. 352] est nécessaire.

En conclusion, il y a deux ensembles d'habiletés ou de connaissances qui sont nécessaires au mesureur :

- les habiletés ou les connaissances pour comprendre le logiciel et le problème à résoudre. Le mesureur doit comprendre la documentation, les modèles et les autres artefacts du logiciel pour situer le problème et l'associer à des expériences passées. À

¹⁴ Pour la suite de cet exemple, nous utilisons le terme "exigence" dans le sens d'exigence de l'utilisateur afin de faciliter la lecture du texte.

noter qu'un développeur peut aider le mesureur à mieux comprendre les modèles de logiciel, mais pour les fins de la mesure, c'est le mesureur qui applique les règles{ XE "règles" } de la méthode de mesure ;

- les habiletés ou les connaissances pour appliquer la méthode de mesure au logiciel. Le mesureur qui utilise la méthode COSMIC-FFP doit, par ses connaissances de la méthode, pouvoir appliquer, c'est-à-dire mettre en correspondance, les informations obtenues à partir des artefacts du logiciel et des règles{ XE "règles" } de la méthode de mesure.

Idéalement, pour une mesure cohérente, il faudrait permettre à différents mesureurs d'arriver à la même solution en utilisant la même information. En étudiant de près le cheminement du mesureur, nous avons déjà constaté qu'il a besoin de plusieurs types de connaissances et d'expériences pour arriver à confirmer ou à infirmer une solution. Il lui faut les connaissances et les expériences suivantes pour :

- créer un modèle mental de la requête de l'utilisateur ;
- faire un lien entre la requête de l'utilisateur et la mesure fonctionnelle utilisée ;
- se poser les bonnes questions et utiliser les règles de modélisation du logiciel de façon adéquate ;
- confirmer ou infirmer les solutions possibles.

4.2 Cheminement du mesureur et autres modèles de mesures

Il est possible de comparer le cheminement du mesureur par rapport aux activités de la phase 2 du processus de mesure et les phases de COSMIC-FFP. On constate que les différentes étapes de la démarche du mesureur peuvent être utilisées pour l'arrimage et pour la mesure. C'est ce qui est présenté dans la Figure 4-4.

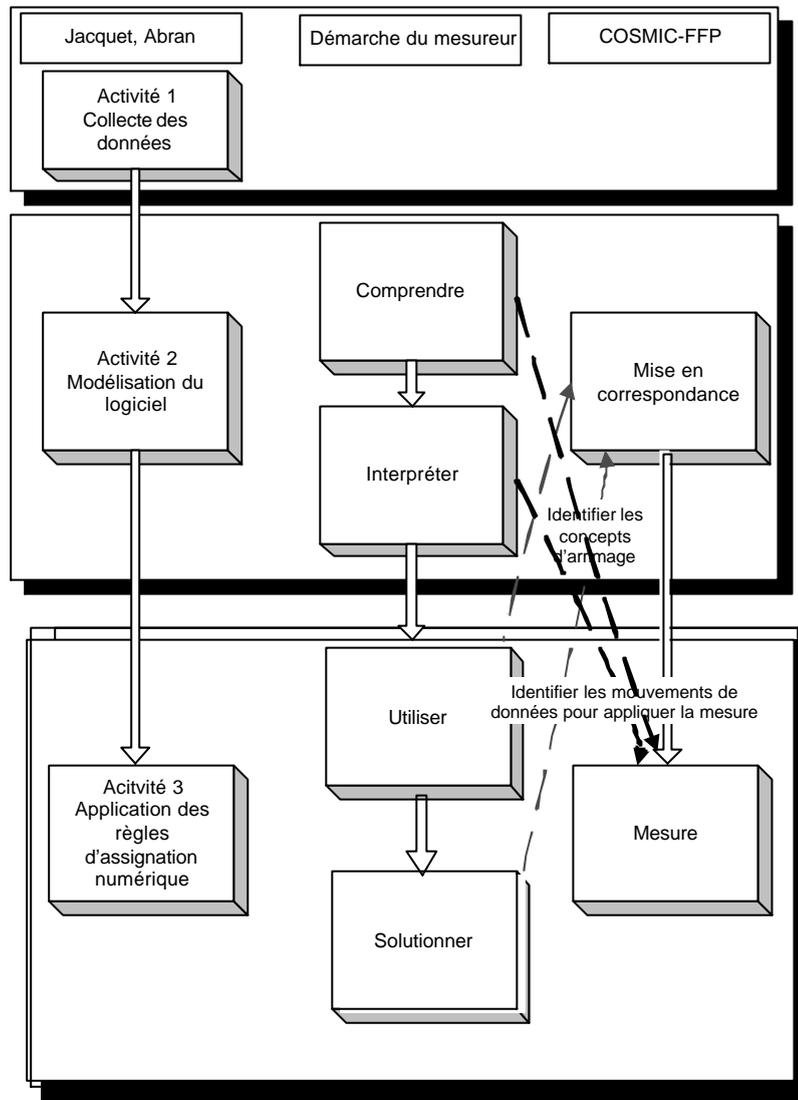


Figure 4-4 : Cheminement du mesureur, activités de la phase 2 et COSMIC-FFP

Dans Jacquet et Abran [29], on retrouve, comme première activité, la collecte des données. Cette collecte des données précède la mesure. Elle ne fait pas partie de la méthode de mesure COSMIC-FFP ni de la démarche du mesureur. Cette activité, si elle est bien organisée, permet de rendre disponible au mesureur, d'une façon ordonnée, les informations nécessaires

à la mesure des logiciels. S'il n'y a pas de collecte des données organisée dans l'entreprise, le mesureur devra trouver un moyen d'obtenir cette information avant de commencer la mesure.

L'activité 2 (modélisation du logiciel) correspond à la phase de mise en correspondance dans COSMIC-FFP et aux étapes de compréhension et d'interprétation dans le cheminement du mesureur. Rappelons que la phase de mise en correspondance dans la méthode de mesure COSMIC-FFP comprend l'identification des couches des logiciels, l'identification de la frontière de chaque logiciel et l'identification des processus. Pour cette activité, le mesureur peut aussi utiliser les règles{ XE "règles" } de la méthode de mesure pour identifier ces concepts.

Enfin, l'activité 3 (application des règles{ XE "règles" } d'assignation numérique) correspond à la phase mesure dans le manuel COSMIC-FFP{ XE "COSMIC-FFP" }. Dans le cheminement du mesureur{ XE "mesureur" }, cette activité correspond à l'utilisation des règles de la méthode de mesure pour identifier les mouvements de données et pour réaliser la mesure du logiciel. Pour identifier les mouvements de données, il doit aussi comprendre et interpréter le mouvement de données (entrée, sortie, lecture ou écriture).

4.3 Le mesureur et la méthode de type diagnostic

Tel que déjà mentionné, la méthode de type diagnostic, appliquée à notre problématique de mesure, nécessite les deux types de connaissances nécessaires au mesureur, soit la connaissance du logiciel mesuré et ses artefacts et la connaissance de la méthode de mesure COSMIC-FFP. La construction d'un logiciel repose sur des méthodes et des modèles qui diffèrent le plus souvent d'une organisation à une autre et même d'un projet de construction à un autre dans la même organisation. La méthode de mesure COSMIC-FFP a ses procédures, ses définitions, ses principes et ses règles{ XE "règles" } (voir note 2). La méthode de diagnostic peut être assez précise en ce qui concerne l'interprétation des règles de la méthode de mesure, puisque l'information est disponible. Pour obtenir la même précision pour les méthodes et les modèles de construction du logiciel, il faudra le même niveau d'information.

4.3.1 Méthodes et modèles de construction du logiciel

Nous savons que les méthodes et les modèles de construction et de maintenance du logiciel en génie logiciel varient, le plus souvent, d'une organisation à une autre, mais aussi à l'intérieur d'une même organisation. Il est donc difficile de se raccrocher à une seule méthode ou à un seul modèle. Nous avons déjà mentionné, dans notre revue de littérature, des modèles bien connus tels, à titre d'exemple, le modèle en cascades décrit par Boehm [19, p. 36-37] et celui du corpus de connaissances SWEBOK [28].

Le fait qu'il n'existe pas de méthode unique de développement et de maintenance du logiciel dans les différentes organisations ne nous permet pas de définir une approche systématique pour la compréhension du problème et son interprétation. L'existence d'une méthode unique et systématique faciliterait non seulement la construction d'un système de connaissances, mais permettrait éventuellement l'automatisation de la mesure, ce qui n'est pas le cas présentement. La méthode de type diagnostic que nous proposons implique que le mesureur devra avoir une connaissance du logiciel et de ses artefacts et qu'il pourra faire un lien entre les concepts de la méthode de mesure COSMIC-FFP et les artefacts du logiciel à mesurer. Il est cependant possible pour les organisations de documenter leur méthodologie en rapport avec la méthode de mesure COSMIC-FFP.

4.3.2 Connaissance de la méthode de mesure COSMIC-FFP

À partir de la méthode de mesure COSMIC-FFP [4], il est possible de décrire les étapes qui seront à la base des principales tâches du mesureur. Le cheminement du mesureur qui peut s'appliquer à chacune de ces étapes est modélisé à la Figure 4-6.

Mise en correspondance

Étape 1a : Sur la base des exigences et des spécifications de l'interaction entre les différents niveaux d'architecture du logiciel, le mesureur doit détecter s'il y a plus d'une couche et les identifier.

Étape 1b : Sur la base des exigences et des spécifications de l'interaction entre l'équipement et le logiciel, le mesureur doit identifier les utilisateurs.

Étape 2 : Sur la base des exigences et des spécifications de l'interaction entre l'équipement et le logiciel, le mesureur doit identifier la frontière du logiciel (étape 2, Figure 4-6).

Étape 3a : À partir des exigences, le mesureur doit identifier tous les processus fonctionnels du logiciel (étape 3, Figure 4-6)

Étape 3b : À partir des exigences, le mesureur doit identifier tous les événements déclencheurs du logiciel. Il est à noter que l'événement déclencheur pourrait aussi être étudié au niveau des mouvements de données, puisque de même niveau. Nous avons opté pour l'étudier au niveau du processus puisque l'événement déclencheur sert surtout à identifier le processus (étape 3, Figure 4-6).

Étape 3c : À partir des exigences, le mesureur doit identifier tous les groupes de données du logiciel.

Mesurage

Étape 4a : Les mouvements de données (4b : Entrée, 4c : Sortie, 4d : Lecture, 4 : Écriture) (4-5) doivent être identifiés pour chaque processus du logiciel.

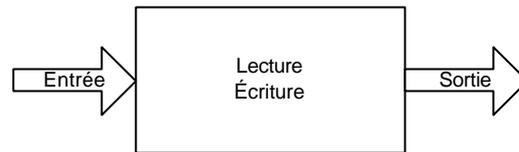


Figure 4-5 : Mouvements de données

Étape 5 : La taille fonctionnelle du logiciel résulte de l'agrégation des résultats de la mesure, c'est-à-dire de l'addition de la taille de tous les mouvements de données. Cette opération étant strictement arithmétique, elle ne requiert pas d'expertise particulière.

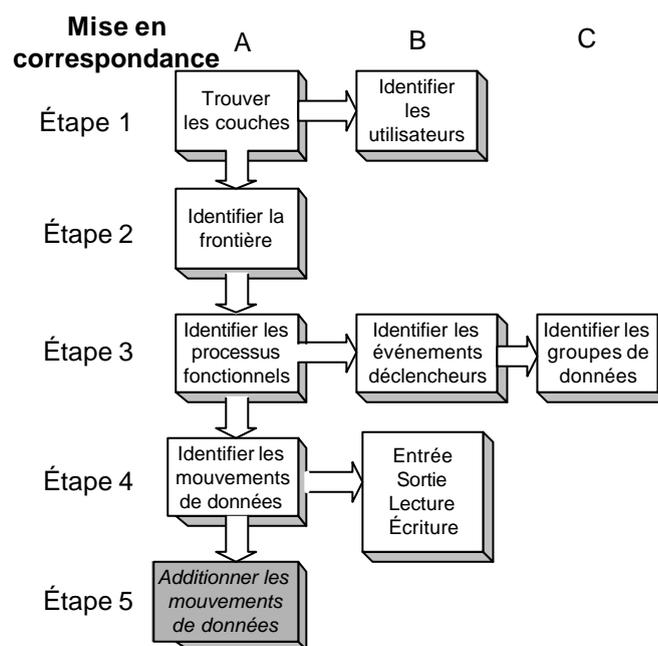


Figure 4-6 : Étapes et tâches de la méthode de mesure COSMIC-FFP

À partir de ces étapes, il est possible de retrouver les principaux cas problèmes rencontrés par le mesureur (Figure 4-6). Chaque bloc peut être identifié à un problème que le mesureur cherche à solutionner. Il faut noter aussi qu'il y a une séquence dans les tâches (Figure 4-6). Une tâche appartenant à l'étape 1 doit être réalisée avant la tâche de l'étape 2, et ainsi de

suite. Par exemple, à l'étape 2, le mesureur doit d'abord identifier la frontière du logiciel s'il veut pouvoir identifier les processus appartenant à ce logiciel.

Ces tâches, mises en relation avec la démarche du mesureur et les artefacts du logiciel à mesurer, nous permettent de proposer une procédure visant à permettre au mesureur de solutionner les différents problèmes de mesure de façon cohérente. Elles sont aussi notre base pour modéliser la connaissance du mesureur. En effet, si nous prenons l'exemple des processus, il n'est pas possible pour le mesureur de les identifier sans avoir des informations sur les processus du logiciel. Ces informations se trouvent dans la documentation (livrable) proposée par la méthode de développement et de maintenance du logiciel. On peut trouver par exemple, dans le corpus de connaissances de SWEBOK, une description (non nécessairement exhaustive) de cette documentation selon les phases de développement et de maintenance du logiciel. Il appartient à chaque organisation de définir ces informations pour faciliter le travail des mesureurs. C'est ce que l'on appelle les "règles locales".

Le travail du mesureur consiste à identifier les documents pertinents à l'identification des "concepts" décrits dans la méthode de mesure COSMIC-FFP. Chaque organisation doit bien identifier les artefacts de sa méthodologie pour aider le mesureur à s'orienter dans la documentation des logiciels. C'est en quelque sorte une indexation de la documentation du logiciel. Par exemple, cette indexation peut aider à répondre aux questions suivantes : dans quel type de documentation peut-on trouver les informations relatives à l'identification des couches ? Des processus fonctionnels ? Des groupes de données ?

4.3.3 Modélisation de la connaissance du mesureur

Nous reprenons, tel qu'illustré à la Figure 4-22, les cinq types de connaissances (modélisation) selon van Heijst et al. [60], et les adaptons pour nous aider à modéliser la connaissance du mesureur.

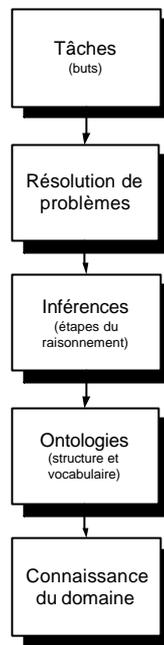


Figure 4-7 : Types de connaissances – inspiré de [60]

Il faut noter que nous allons présenter la connaissance à un niveau plus fin que les exemples de van Heijst [60]. Pour van Heijst, les tâches et les méthodes de résolution de problèmes sont des concepts indépendants des ontologies des domaines. Nous les reprenons ici en sachant que la tâche de haut niveau en est une d'évaluation et les sous-tâches consistent plus concrètement à identifier les éléments pertinents du logiciel et à les diagnostiquer. À partir de la perspective de van Heijst, le mesureur a des tâches de diagnostic. Puisque nous voulons savoir comment le mesureur fera ce diagnostic, il nous faut définir la tâche du mesureur.

4.3.3.1 Tâches

La tâche indique les buts du mesureur et les stratégies employées pour réaliser ces buts. Pour le mesureur, le but est d'identifier les différentes instances des concepts de la méthode de mesure COSMIC-FFP à partir des artefacts de la documentation du logiciel (ou des commentaires des développeurs).

En terme de tâches, on constate que la stratégie employée correspond au cheminement du mesureur :

- le mesureur doit identifier le (ou les) cas problème(s) en rapport avec la tâche à accomplir (i.e. : identifier un artefact du logiciel pour les fins de la mesure) ;
- le mesureur doit aussi pouvoir situer chaque cas problème par rapport aux autres et les interpréter (en fonction de l'artefact qu'il observe) ;
- pour chaque cas problème, le mesureur doit pouvoir utiliser les questions appropriées (les thèmes appropriés) ;
- les réponses aux questions (thèmes) conduisent à la solution cherchée (recommandation) ;
- la solution (recommandation) peut conduire à d'autres cas problèmes ou à une information pertinente qui aidera à trouver une autre solution (recommandation).

La tâche du mesureur est aussi le cheminement du mesureur déjà décrit dans la section 4-1 (Figure 4-1) en y ajoutant une dernière étape qui ramène le mesureur à l'identification d'un nouveau problème à comprendre.

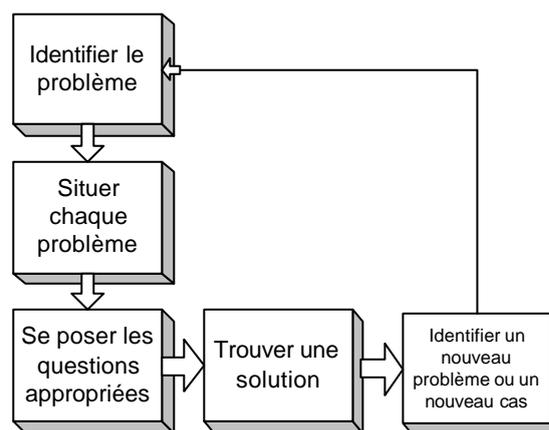


Figure 4-8 : Procédure d'application

Les tâches du mesureur dans la procédure d'application, à la Figure 4-8, les étapes et les sous-étapes de la méthode de type diagnostic que nous décrivons plus loin (Figure 4-11) ont la même logique (en plus détaillé). Ce qui s'explique par le fait que la méthode de type diagnostic de résolution de cas problèmes vise à offrir une solution à la procédure d'application du mesureur.

4.3.3.2 Méthodes de résolution de problèmes

Les méthodes de résolution de problèmes sont des approches pour identifier, pour les fins de la mesure, un concept de la méthode de mesure COSMIC-FFP (but) décrit dans les tâches. Les principales méthodes de résolution de problèmes sont les suivantes :

- la recherche par mot-clé : à partir du vocabulaire du cycle de vie du logiciel et de celui de la méthode de mesure COSMIC-FFP, des mots-clés seront retenus et utilisés pour permettre au mesureur d'identifier l'environnement du problème ;
- la déduction : par exemple, à partir de cas problèmes précis, le système de connaissances déduira quels sont les thèmes à utiliser pour aider à les résoudre ;
- l'induction : par exemple, à partir des réponses aux thèmes, le système de connaissances induira si le concept topologique identifié par le mesureur est le bon ;
- la fouille par chaînage arrière.

4.3.3.3 Inférences

Le terme inférence est généralement utilisé pour désigner un processus qui permet d'obtenir des conclusions [73, p. 176]. Pour Schreiber et al. [42, p. 105], dans le cadre du modèle de connaissance, c'est le plus bas niveau de décomposition fonctionnelle. Nous utilisons aussi le terme "règles d'inférences" pour désigner le processus permettant d'obtenir des conclusions. C'est la "boîte noire" ou le processus interne qui permet d'obtenir l'inférence. La règle d'inférence décrit l'étape du raisonnement permettant de résoudre un problème. Les règles d'inférences sont utilisées par les approches de la résolution de problèmes. Le mesureur effectue deux types d'inférences (Figure 4.24) :

- des recherches permettant de prioriser (avec des %) les cas problèmes à résoudre à partir de mots-clés ou d'autres informations ;
- des recherches permettant de solutionner les cas problèmes ou de recommander d'autres tâches pour mieux circonscrire les cas problèmes à partir de calculs basés sur la théorie de la certitude.



Figure 4-9 : Types d'inférences

4.3.3.4 Ontologies (vocabulaire et structure)

Pour van Heijst [60], une ontologie est un niveau de connaissances explicite spécifiant des concepts, c'est-à-dire un ensemble de distinctions qui sont utiles pour un agent (le mesureur par exemple). La conceptualisation (et ainsi l'ontologie) peut être affectée par un domaine particulier et une tâche particulière.

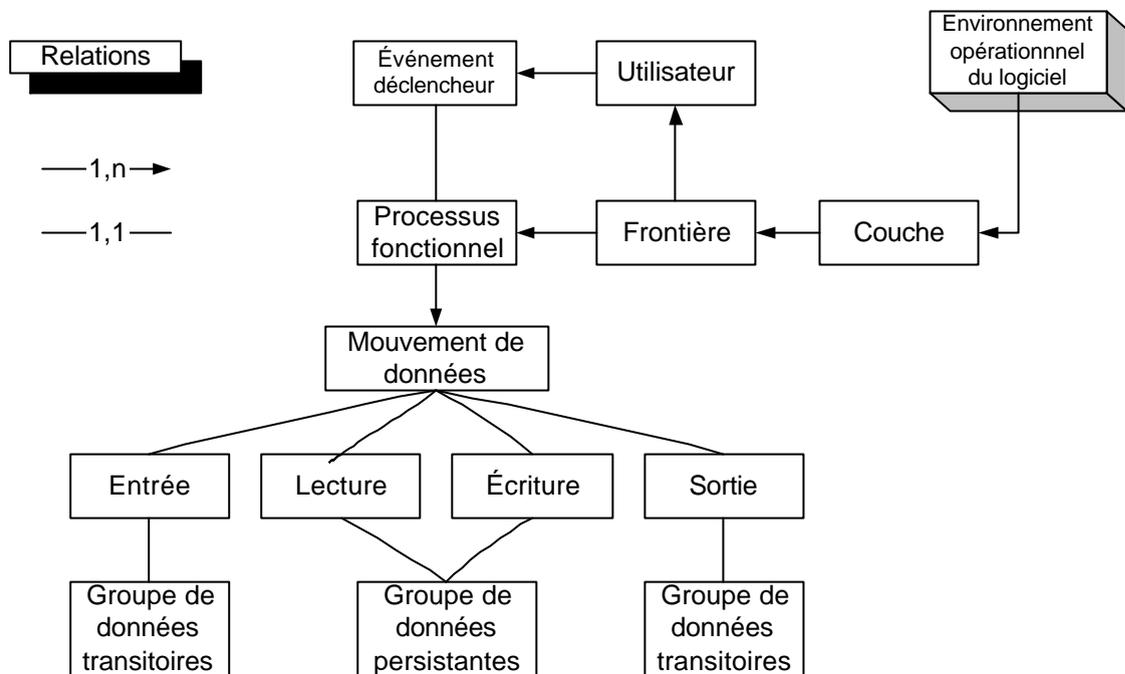


Figure 4-10 : Ontologie COSMIC-FFP [10]

L'ontologie décrit le vocabulaire et la structure du domaine de la connaissance statique. Pour le mesureur, le vocabulaire et la structure nécessaires se trouvent dans le cycle de vie du logiciel et dans la méthode de mesure COSMIC-FFP¹⁵. À partir de l'ontologie COSMIC-FFP que nous reprenons ici (Figure 4-10) à titre de rappel pour le lecteur (Figure 1-3), on peut trouver son vocabulaire (couches, frontières, processus, groupes de données, etc.). Nous avons choisi les concepts de la méthode de mesure COSMIC-FFP pour définir l'ontologie de la méthode de type diagnostic et non ceux des méthodes et des modèles dans les organisations en raison de leur diversité.

15 De ce fait, il est question ici d'une ontologie pour la personne (le mesureur) et non pour la machine.

4.3.3.5 Connaissance du domaine

La connaissance du domaine est un ensemble d'énoncés sur le domaine [60]. Ces énoncés peuvent provenir du mesureur ou de l'expert du domaine de la mesure fonctionnelle pour réaliser la mise en correspondance entre les artefacts du logiciel à mesurer et les règles de la méthode de mesure. Dans les étapes de la méthode de type diagnostic (voir la section suivante), on constate que ces énoncés se situent sur plus d'un niveau.

Ainsi, le mesureur fera un énoncé pour mettre en correspondance le concept de processus dans la méthode de mesure COSMIC-FFP et le modèle de processus dans une méthodologie ou un corpus de connaissances en génie logiciel (ex. : SWEBOK). Il lui faudra trouver un "mot-clé" résumant cet énoncé. L'expert peut suggérer les mots-clés, mais il appartient au mesureur de choisir ce qui lui convient le mieux.

Cet énoncé (mot-clé) permet d'identifier un concept topologique qui s'appuie sur l'ontologie COSMIC-FFP. Le concept topologique permet la mise en correspondance des concepts des méthodes de développement et les concepts de la méthode de mesure COSMIC-FFP. C'est l'expert qui décide des liens (et leur force) entre le mot-clé et le concept topologique.

À un autre niveau, le mesureur devra établir un lien entre un concept topologique et les cas problèmes. Selon le concept topologique, la force des liens (fournie par l'expert) avec les cas problèmes peut varier, mais il appartiendra au mesureur de décider, en dernière analyse, quel énoncé (sous forme de cas problème) correspond le mieux à son "problème".

Au niveau suivant, l'énoncé se présente sous forme de faits appartenant aux thèmes des cas problèmes. En effet, à chaque cas problème appartient un certain nombre de thèmes, ceux-ci pouvant varier selon le type de cas problèmes. Il appartient à l'expert d'énoncer ces thèmes. À ce niveau, la principale tâche du mesureur est de fournir les faits (énoncés) appropriés au thème selon le "problème" qu'il doit résoudre. Cependant, c'est l'expert qui décidera de l'influence de ces faits sur la solution.

Au dernier niveau, la solution ou la recommandation, est un énoncé qui est suggéré par l'expert. Le mesureur devra décider si l'énoncé de solution convient à son "problème".

La méthode de type diagnostic permet de construire une base de connaissances comprenant un ensemble d'énoncés ou de règles sur le domaine, utilisés par le mesureur, de réaliser ses tâches d'identification des concepts de la méthode de mesure COSMIC-FFP et finalement de permettre la mesure du logiciel.

Dans la prochaine section, nous allons décrire les étapes et les sous-étapes de la méthode de type diagnostic.

4.4 Méthode de type diagnostic

4.4.1 Les étapes et les sous-étapes de la méthode

Les étapes de la méthode de type diagnostic sont représentées dans la Figure 4-11 :

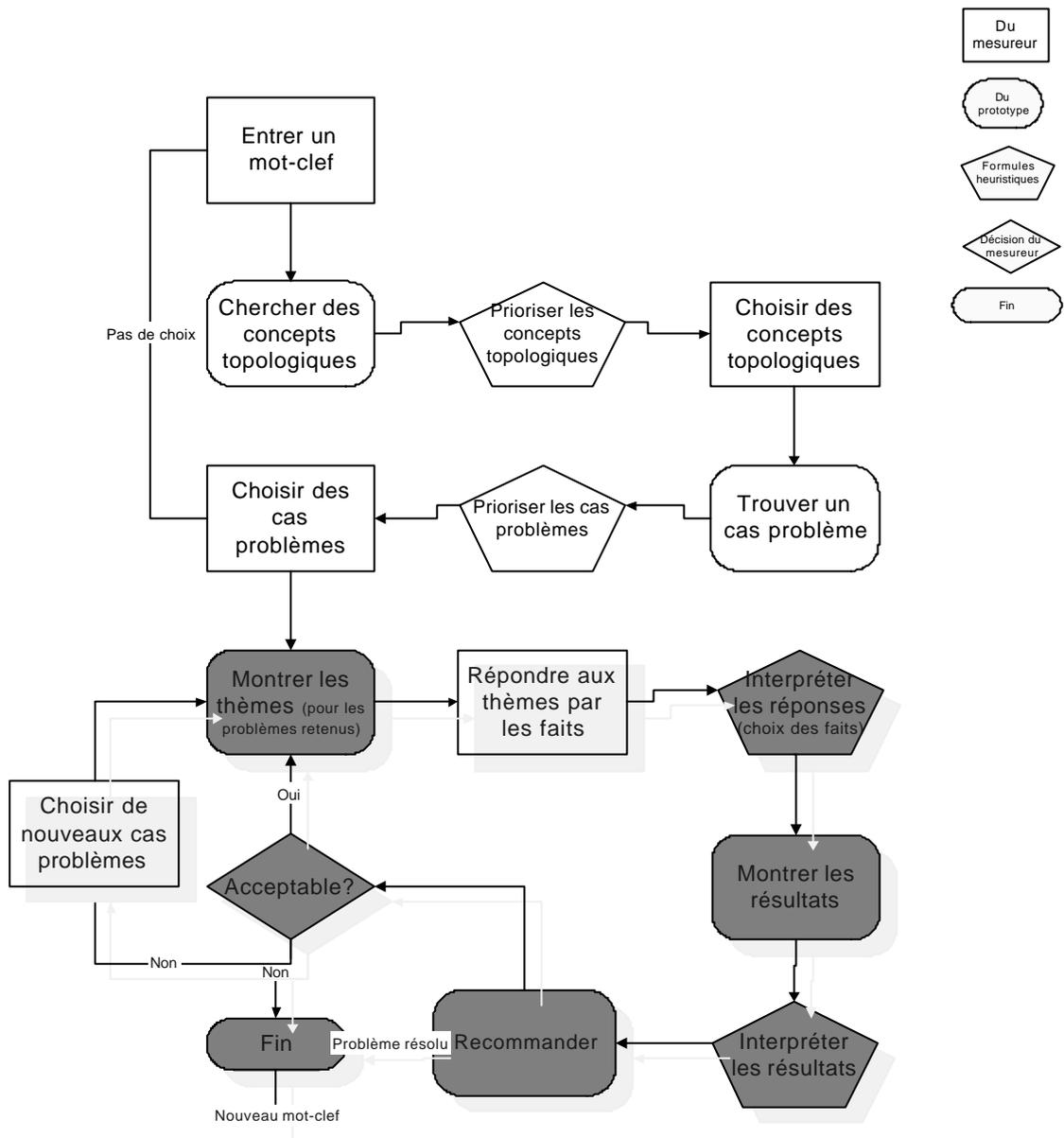


Figure 4-11 : Étapes et sous-étapes de la méthode de type diagnostic

La partie supérieure correspond à la recherche de "cas problèmes" (CBR) alors que la partie inférieure correspond à la résolution du cas problème par des règles (à base de règles). Pour

le mesureur, la partie supérieure correspond à comprendre et à interpréter, alors que la partie inférieure correspond à utiliser (les règles) et à solutionner (recommandation).

4.4.2 Description des étapes de la méthode de type diagnostic

La méthode de type diagnostic comprend plusieurs étapes et sous-étapes. Voici une brève description de ces dernières :

Étape 1 : Identifier le problème

Sous-étape 1a : Entrer un mot-clé¹⁶

Le mesureur doit entrer un mot-clé lui permettant de situer le problème qu'il doit résoudre. Ce mot-clé peut être un concept dans le vocabulaire de la méthode de mesure COSMIC-FFP, tels "processus", "lecture", "écriture", "couche", etc. Il peut aussi être un mot trouvé dans la documentation des logiciels mesurés, tels "fichier", "table", "événement", etc. Ça peut aussi être un concept trouvé dans le vocabulaire de SWEBOK tel "design", "maintenance", "requis", etc. Ces mots-clés devront avoir été prévus par l'expert dans la méthode de type diagnostic.

Sous-étape 1b : Trouver un concept topologique

Cette sous-étape est réalisée par l'outil de diagnostic¹⁷ ou le prototype de diagnostic. Le prototype, à partir du mot-clé, fait une recherche des concepts topologiques associés à ce mot-clé. Le concept topologique est le point d'intersection¹⁸ entre un concept du processus de développement et une partie de la tâche du mesureur. Autrement dit, le concept topologique

16 Nous employons le terme au singulier, mais il pourrait y avoir plus d'un mot-clé.

17 La méthode diagnostic se concrétisera dans un outil de diagnostic ou de prototype de diagnostic. Lors de la description des étapes et des sous-étapes de la méthode diagnostic, nous allons indiquer si elles exigent l'intervention du mesureur ou du prototype. Nous utiliserons le mot prototype pour signifier le prototype de diagnostic, sauf indication contraire.

18 C'est ce qui fait le lien.

permet la mise en correspondance des concepts du processus de développement (dans le cas présent ceux du corpus de connaissances SWEBOK) et des concepts de la méthode de mesure COSMIC-FFP qui lui apportent la solution ou le conduisent à un autre problème.

Sous-étape 1c : Prioriser les concepts topologiques

Le prototype est conçu pour permettre à l'expert de prioriser les concepts topologiques sur la base des mots-clés. À chaque mot-clé sera associé un pourcentage représentant la croyance d'appartenance des mots-clés aux différents concepts topologiques. Un mot-clé peut trouver plusieurs concepts topologiques. Un concept topologique peut être trouvé à partir de plus d'un mot-clé. De plus, pour permettre de déterminer la priorité d'un concept topologique par rapport à un mot-clé, il faut lui associer un pourcentage. Les pourcentages d'association entre les mots-clés et les concepts topologiques permettent de déterminer la priorité des concepts topologiques par rapport à un mot-clé.

Sous-étape 1d : Choisir un concept topologique

Le mesureur peut faire un ou plusieurs choix entre plusieurs concepts topologiques pour la suite des opérations. Notre hypothèse est que le mesureur peut faire ce choix mieux que le prototype puisque c'est lui qui a accès à la documentation du logiciel qu'il mesure. Les options du mesureur sont :

- de faire un choix à partir des suggestions du prototype ;
- de faire plus d'un choix à partir des suggestions du prototype ;
- de ne pas faire de choix et de prendre toutes les suggestions ;
- de refuser tous les choix et de revenir à l'entrée d'un nouveau mot-clé.

Étape 2 : Situer le problème

Sous-étape 2a : Trouver les cas problèmes

À partir du (ou des) choix du mesureur (concept topologique), le prototype va trouver les cas problèmes associés à chaque choix en leur donnant une priorité.

Sous-étape 2b : Prioriser les cas problèmes

Tout comme pour le concept topologique, la priorité des cas problèmes a été donnée par l'expert en termes de pourcentage.

Sous-étape 2c : Choisir les cas problèmes

Tout comme pour le concept topologique, le mesureur peut faire un ou plusieurs choix entre plusieurs cas problèmes.

Étape 3 : Établir les thèmes appropriés

Sous-étape 3a : Montrer les thèmes (pour tous les cas problèmes retenus)

Un certain nombre de thèmes sont associés à chaque cas problème. Il est possible qu'un thème soit associé à plus d'un cas problème apparaissant dans la liste des cas problèmes retenus. Le même thème ne devrait pas apparaître plus d'une fois.

Sous-étape 3b : Établir les faits pour chaque thème

Le mesureur donne une réponse positive ou négative¹⁹ à chaque thème associé à un (ou des) cas problème (s). Il peut y avoir un des trois types de faits suivants pour chaque thème : des choix multiples, oui/non ou un nombre.

¹⁹ Pour les fins de notre recherche, nous nous en sommes tenus à une réponse positive ou négative, mais rien n'empêche d'avoir un multiple choix.

Sous-étape 3c : Évaluer les résultats

Les thèmes seront évalués sur la base des faits par le prototype. Vous trouverez à la section Inférences une explication des formules qui sont utilisées pour évaluer les thèmes à partir des faits.

Étape 4 : Trouver une solution

Sous-étape 4a : Montrer les résultats

Les résultats sont montrés au mesureur en termes de pourcentage. Ce dernier peut se faire une idée sur le cas problème à partir de ces résultats bruts.

Sous-étape 4b : Évaluer les résultats

Le mesureur fournit une réponse à chaque thème et le prototype calcule les résultats sur la base de paramètres d'interprétation fournis par l'expert. L'expert peut, à partir des pourcentages, fournir des seuils qui indiqueront au mesureur si la solution est trouvée ou non.

Sous-étape 4c : Recommander une action

Une recommandation quant à la suite des actions à prendre sera faite par le prototype. Cette recommandation pourra être la suivante :

- indiquer que la solution est trouvée (ex. : le mesureur a clairement identifié un processus) et qu'il n'est pas nécessaire de faire plus de recherches ou recommander d'autres cas problèmes.

Étape 5 : Nouveaux cas

Le mesureur répète les étapes à partir de l'étape 2. Le prototype peut aussi donner une explication sur la recommandation si le mesureur le désire et si l'explication est disponible.

4.4.3 Création des cas problèmes

La méthode de type diagnostic sera utilisée par le mesureur. Une autre question se pose : comment l'expert devrait-il s'y prendre pour créer les cas problèmes à partir de son expérience ? Nous y répondrons après avoir donné un exemple de l'application des règles d'un logiciel de diagnostic.

4.5 Exemple d'application des règles

Voici un exemple (Tableau 4-1) d'application des règles{ XE "règles" } de la méthode de mesure fonctionnelle sur la base de la théorie de la certitude. Dans l'exemple présenté ici, le mesureur veut savoir si l'élément de logiciel est bien un processus fonctionnel.

Dans cet exemple, nous présentons ce que le mesureur peut voir lorsqu'il doit résoudre différent cas problèmes²⁰. Un cas problème comprend deux thèmes (nous présentons deux thèmes pour simplifier l'exemple). Nous avons six (6) colonnes qui se comprennent comme suit :

Colonne 1 : identification du cas problème qui est ici l'identification d'un processus ou notre conclusion

Colonne 2 : identification des thèmes qui sont ici liés uniquement au nombre d'événements déclencheurs et à la nature de l'événement déclencheur

Colonne 3 : les différentes possibilités de faits quant à la nature des thèmes

Colonne 4 : le pourcentage attribué à un fait spécifique en rapport avec la nature du thème de la colonne (2). Le pourcentage indique la confiance de l'expert qu'il y a un lien positif entre

²⁰ Le cas problème est aussi une conclusion dans notre exemple.

un fait spécifique lié au thème et l'existence d'un processus (notre cas problème ou notre conclusion). La confiance de l'expert en l'existence d'au moins un événement indiquant qu'il y a effectivement un processus, est à 80 % pour le fait a).

Colonne 5 : le pourcentage indique la confiance de l'expert qu'il y a un lien négatif entre le thème et l'existence d'un processus (notre cas problème). La confiance de l'expert que la non-existence d'au moins un événement déclencheur puisse indiquer qu'il n'y a pas de processus, est à 60 % pour le fait c).

Colonne 6 : indique le degré de confiance de l'expert que le thème est pertinent au cas problème, ou encore que le thème permet de résoudre effectivement le cas problème. L'expert a confiance à 80 % que le nombre d'événements déclencheurs est un thème qui permet d'identifier un processus.

Tableau 4-1 : Exemple d'application des règles

(1) Cas problème (H)	(2) Thèmes (F)	(3) Faits	(4) CF R	(5) CF R	(6) CF R
Identification d'un processus	Nombre d'événements déclencheurs	a) Au moins un événement déclencheur	0,8		0,8
		b) Il y a un événement déclencheur	0,6		0,8
		c) Il n'y a pas d'événement déclencheur		0,6	0,8
	Nature de l'événement déclencheur	a) Action d'un utilisateur humain	0,8		0,6
		b) Action d'un autre logiciel	0,6		0,6
		c) Données provenant d'une horloge	0,4		0,6
		d) Données d'une autre couche		0,2	0,6

Nous utilisons (Tableau 4-1) la formule $CF(H,E1) = CF(E1) * CF(RULE 1)$, où $CF(RULE 1)$ correspond à la multiplication des colonnes (4) ou (5) multiplié par la colonne (6).

Tableau 4-2 : Thème 1 selon les faits

CF(H,E1) = CF(E1) * CF(RULE 1) conjonctive	
Choix de a) E 1 = a	0,64
Choix de b) E 1 = b	0,48
Choix de c) E 1 = c	-0,48

Tableau 4-3 : Thème 2 selon les faits

CF(H,E2) = CF(E2) * CF(RULE 1) conjonctive	
Choix de E 2 = a	0,48
Choix de E 2 = b	0,36
Choix de E 2 = c	0,24
Choix de E 2 = d	-0,12

Les deux tableaux précédents (4-2 et 4-3) montrent le résultat de la confiance que l'on peut avoir en la conclusion considérant son poids. Le fait a) avec le thème 1 donne une confiance de 0,64, alors que le fait c) pour le même thème donne une confiance de -0,48.

Pour obtenir ces résultats, on multiplie le poids donné au fait (ou confiance par rapport au choix) et le poids donné au thème par rapport au problème.

Tableau 4-4 : Selon deux faits (positifs)

CFcombine(CF1,CF2) = CF1 + (CF2 * (1 - CF1)) conjonctive		Note : Les deux > 0
Choix de a) E 1 et a) E 2	0,81	
Choix de b) E 1 et b) E 2	0,67	
Choix de b) E 1 et c) E 2	0,60	

Le Tableau 4-4 donne différentes solutions selon différents choix de faits aux thèmes 1 et 2. On constate aussi que la combinaison d'un fait positif à un autre fait positif donne un résultat positif encore plus élevé. Ceci confirme l'influence de deux faits positifs, selon deux sources, comme étant plus positive.

Tableau 4-5 : Selon deux faits (positif et négatif)

CFcombine(CF1,CF2) = (CF1 + CF2) / (1 - min (CF1, CF2))		Note : Un < 0
Choix de E 1 = c et E 2 = a	0,00	
Choix de E 1 = b et E 2 = d	0,41	
Choix de E 1 = a et E 2 = d	0,59	

Pour le Tableau 4-5, nous avons pris des réponses avec des thèmes ambigus, soit la combinaison d'un fait positif à un fait négatif. Dans tous les cas, les résultats sont mitigés, passant de 0,00 à 0,59. Il n'est pas certain que l'on ait pu identifier un événement déclencheur.

Tableau 4-6 : Selon deux faits (négatif)

$CF_{combine}(CF1,CF2) = CF1 + (CF2 * (1 + CF1))$		Note : Les deux < 0
Choix de E 1 = c et E 2 = d	-0,54	

Le Tableau 4-6 combine deux pourcentages négatifs. Le résultat est non ambigu : il ne s'agit pas d'un événement déclencheur.

5. PRÉSENTATION ET EXPÉRIMENTATIONS DU PROTOTYPE 1

Nous avons démarré la concrétisation de l'outil de diagnostic par la construction d'un premier prototype (5.1). Nous avons ensuite fait vérifier par des experts les recommandations que nous avons intégrées au premier prototype (5.2.1). Suite à cette vérification, une version plus complète du prototype a été élaborée et celle-ci a été utilisée pour une expérimentation auprès de mesureurs débutants (5.2.2).

5.1 Présentation des exigences du prototype 1

Nous présentons dans cette section les fonctionnalités du prototype 1 (voir Annexe B pour des informations plus détaillées).

Le premier prototype a été construit avec une trentaine de cas problèmes dans la base de connaissances. Il a été ensuite utilisé en mode local pour vérifier les résultats de l'outil de diagnostic.

Il y a trois types d'utilisateurs pour le prototype 1 :

- le mesureur : c'est la personne qui utilise le prototype pour interroger la base de connaissances en vue d'obtenir une recommandation (solution) ;
- l'expert : c'est la personne qui utilise le prototype pour entrer les différentes informations dans la base de connaissances (mots-clés, concepts topologiques, cas problèmes, thèmes avec réponses et recommandations) ;
- l'administrateur : c'est la personne qui gère les accès au prototype en inscrivant les noms des utilisateurs selon leur type (mesureur et expert) ainsi que les mots de passe et la durée des accès.

Le diagramme de cas d'utilisation (Figure 5-1) est une représentation du cheminement du mesureur et montre les différents scénarios utilisés par ces différents types d'utilisateurs avec

le prototype. Le mesureur entre un mot-clé (compréhension) et le prototype fait apparaître des concepts topologiques associés au mot-clé, puis les cas problèmes associés aux différents concepts topologiques et enfin les thèmes qui correspondent aux différents concepts topologiques (UC-2). Dans un deuxième temps, le mesureur peut réduire le nombre de cas problèmes en éliminant les concepts topologiques jugés non pertinents. Le mesureur peut ensuite réduire le nombre de thèmes en éliminant les cas problèmes non pertinents jusqu'à ce qu'un seul cas problème apparaisse à l'écran (interprétation) (UC-3). Enfin, le mesureur peut donner une réponse aux différents thèmes (utilisation) (UC-4) et trouver la recommandation appropriée (solution) (UC-4). Le mesureur a aussi la possibilité de consulter les références relatives à tous les concepts (mot-clés, concepts topologiques, cas problèmes, thèmes, recommandations). Il peut aussi avoir une définition de chaque concept.

L'expert utilise des scénarios (Figure 5-1) qui permettent de mettre à jour la base de connaissances (gestion des mots-clés, des concepts topologiques, des cas problèmes, des thèmes, des faits, des recommandations, du modèle de diagnostic, des informations sur l'utilisation des connaissances, des "boutons" et des rapports). L'expert peut aussi mettre à jour toutes les références.

Seul l'administrateur peut utiliser les scénarios (Figure 5-1) de gestion pour l'ensemble des utilisateurs (nouveaux utilisateurs, mots de passe, statut de l'utilisateur).

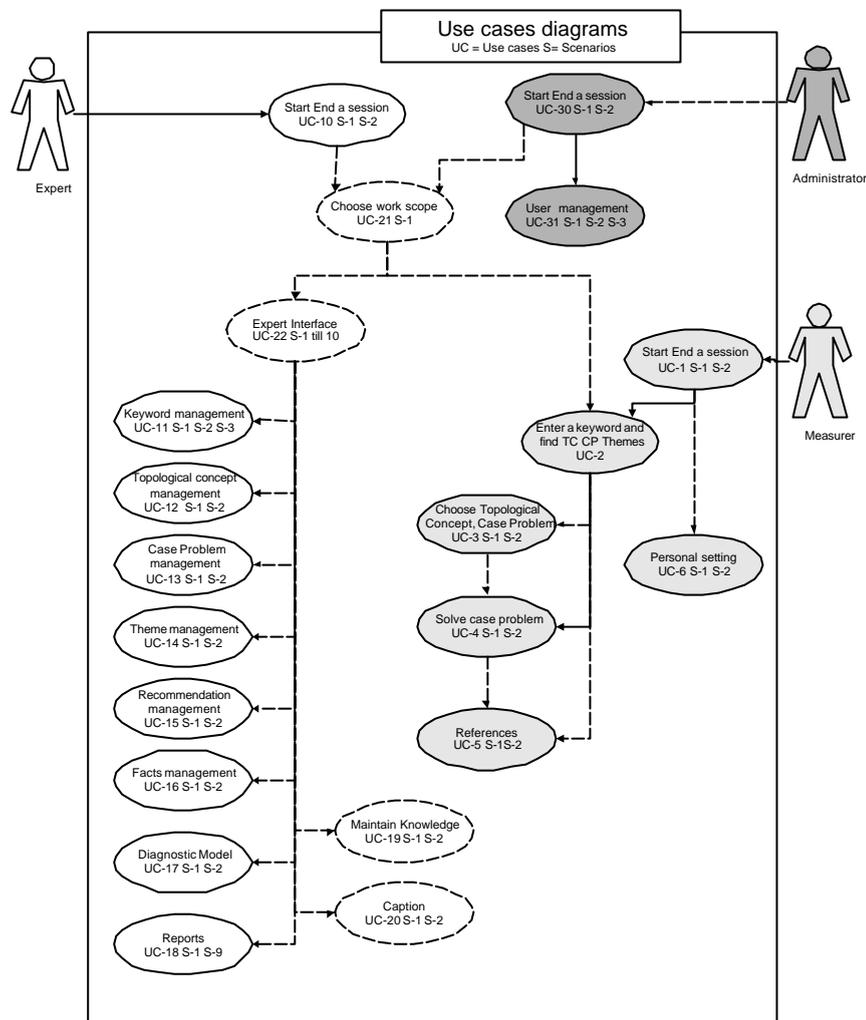


Figure 5-1 : Diagramme de cas

Les listes des scénarios (avec les acteurs, les événements et les entrées/sorties) sont présentées au Tableau 5-1 pour le mesureur et au Tableau 5-2 pour l'expert et l'administrateur.

Tableau 5-1 : Scénarios du mesureur

Acteur	Événement	Entrée/Sortie
Mesureur	Entrer un mot clé	(E)
Mesureur	Rechercher les concepts topologiques et les cas problèmes	(E)
Mesureur	Sélectionner les concepts topologiques et les cas problèmes désirés	(E)
Mesureur	Sélectionner les faits sur la base des informations disponibles via la documentation	(E)
Mesureur	Entrer une information sur la qualité de la documentation en regard d'un thème particulier (Note : a été enlevé dans le deuxième prototype)	(E)
Mesureur	Recommandation appropriée	(S)
Mesureur	Réponse ou valeur sur la qualité de la documentation	(E)
Mesureur	Demander comment compacter la base de données	(E) et (S)
Mesureur	Demander une explication sur un mot clé, un concept topologique, un cas problème, un thème ou une recommandation spécifique	(E) et (S))
Mesureur	Demander une description de ce qu'est une ontologie, un mot clé, un concept topologique, un cas problème, un thème ou une recommandation.	(E) et (S)
Mesureur	Requête sur les paramètres de l'utilisateur	(S)
Mesureur	Modifier le nom de l'utilisateur, le mot de passe et le langage	(E)

Tableau 5-2 : Scénarios de l'expert et de l'administrateur

Acteur	Événement	Entrée/Sortie
Expert	Ajouter, modifier et enlever un cas spécifique de connaissances	(E)
Expert	Modifier les boutons, les en-têtes, les titres et autres des interfaces	(E)
Expert	Demander des rapports tels le suivi du mesureur, la session ou encore les relations entre les connaissances d'expertise	(S)
Expert	Montrer une description sommaire d'un noeud dans un arbre lorsque sélectionné	(S)
Expert	Copier, couper et coller des noeuds dans l'arbre	(E)
Expert	Demander les propriétés des cas spécifiques des connaissances	(S)
Expert	Sélectionner l'outil de diagnostic désiré	(E)
Expert	Maintenir des cas spécifiques de connaissances dans différents langages (Note : pas dans le deuxième prototype)	(E)
Expert	Enlever le rapport de suivi du mesureur ou la session	(E)
Expert	Aller vers les fonctionnalités du mesureur et résoudre un problème	(E)
Administrateur	Ajouter, modifier ou enlever un code utilisateur	(E)
Administrateur	Utiliser les fonctionnalités de l'expert	(E) et (S)
Administrateur	Utiliser les fonctionnalités du mesureur	(E) et (S)

5.2 Vérification et expérimentation du prototype 1

L'objectif de la vérification et de l'expérimentation avec le prototype 1 est de répondre à deux questions :

- Est-ce que les experts de la mesure acceptent les recommandations du prototype ? (Q1)
- Est-ce que le prototype 1 est utile au mesureur débutant ? (Q2)

5.2.1 Vérification par des experts du prototype 1

5.2.1.1 Description de la vérification des recommandations par des experts.

Il a été demandé à quatre experts (Annexe E) de faire un exercice de vérification de la base de connaissances (Annexe D). Tous les experts avaient les caractéristiques minimales suivantes :

- expérience de l'application de la mesure fonctionnelle depuis plusieurs années ;
- participation directe ou indirecte à la construction de la méthode de mesure COSMIC-FFP ;
- expérience avec la mesure fonctionnelle de logiciels avec la méthode de mesure COSMIC-FFP ;
- contribution à la création d'au moins un cas à la méthode de mesure COSMIC-FFP (*Rice Cooker, Valve Control, Gendarme*, etc.) ;
- ils n'ont pas contribué à la préparation du prototype.

La vérification du prototype 1 par les experts a aussi permis de demander à ces experts leur opinion sur la facilité d'utilisation du prototype.

5.2.1.2 Déroulement du processus de la vérification

Voici comment s'est déroulée la vérification avec les quatre experts :

- il a été demandé aux experts d'utiliser le prototype 1 pour retrouver tous les cas problèmes et indiquer, en répondant aux différents thèmes, s'ils étaient d'accord avec la recommandation fournie par le prototype ;
- il a aussi été demandé aux experts d'indiquer (par des commentaires) s'ils comprenaient l'explication du cas problème, les thèmes, les réponses à chaque thème en plus de la formulation de la recommandation ;
- il a été demandé aux experts d'indiquer (commentaires) s'ils n'étaient pas d'accord avec la recommandation ;

- les quatre experts ont tous analysé les mêmes cas problèmes, soit 34.

5.2.1.3 Résultats de la vérification

Le Tableau 5-3 présente les pourcentages du nombre de recommandations où les experts sont d'accord et du nombre de recommandations où ils sont en désaccord. Ils ont aussi fourni des commentaires écrits indiquant en quoi l'explication d'un cas ou sa recommandation n'était pas claire.

Tableau 5-3 : Résultats de la vérification par les 4 experts – prototype 1

Accord ou désaccord avec les recommandations du prototype					
	Expert 1	Expert 2	Expert 3	Expert 4	Moyenne
% en accord	82 %	79 %	85 %	85 %	83 %
% en désaccord	15 %	18 %	3 %	9 %	11 %
% ne sait pas	3 %	3 %	12 %	6 %	6 %

Pour plus de 80 % des cas problèmes, les experts sont d'accord avec les recommandations du prototype. Entre les quatre (4) experts, les pourcentages en accord avec les recommandations des cas problèmes sont assez constants puisque la variation n'est que de 6% entre le plus haut et le plus bas (79 % et 85 %). Entre les quatre (4) experts le pourcentage des experts en désaccord avec les recommandations des cas problèmes a une plus grande variation puisque la variation est de 15 % entre le plus haut et le plus bas (3 % et 18 %). Les experts ont aussi indiqué que, pour certains cas problèmes, ils ne savaient pas s'ils étaient en accord ou en désaccord avec ces recommandations (les pourcentages varient de 3 % à 12 %).

A) Analyse des variations d'opinion entre les quatre experts

Il y a 26 cas problèmes sur 34 (76 %) où les quatre experts sont d'accord avec la recommandation. Autrement dit, dans plus des trois quarts des cas problèmes, lorsqu'un expert est d'accord avec une recommandation, les autres le sont aussi.

Comment peut-on expliquer les désaccords et s'agit-il bien de désaccords ? Nous avons analysé le contenu des commentaires des experts lorsqu'ils indiquaient être en désaccord avec une recommandation dans le prototype 1. Par exemple, pour un cas problème, les experts 1 et 2 ont commenté que l'explication n'était pas claire tout en indiquant un désaccord quant à la recommandation à ce cas problème. L'expert 3 a indiqué que l'explication n'était pas claire tout en répondant "ne sait pas" pour le cas problème.

Voici trois exemples spécifiques : 1) dans le cas problème appelé *Creating a Report RG* l'expert 2 écrit qu'il est en désaccord, mais son commentaire écrit est : "revoir l'explication". 2) il y a aussi un cas problème où trois experts sur quatre ne sont pas d'accord et le quatrième indique qu'il ne comprend pas l'explication. Nous avons analysé le cas problème intitulé *How to Identify an Exit*. Nous avons constaté qu'il était mal formulé, ce qui laissait croire à une mauvaise recommandation. 3) on retrouve cette ambiguïté dans la formulation du cas problème intitulé *Software Users*. Le résultat indique que trois experts ne sont pas d'accord alors qu'un expert est d'accord avec la recommandation du cas problème. En résumé, sa formulation était compréhensible mais ambiguë.

B) Leçons apprises

En référence à notre question (Q1), à la suite de cette vérification avec les quatre experts, nous avons constaté qu'ils étaient généralement d'accord (83 %) avec les recommandations incluses dans le prototype 1. Lorsqu'un expert était en désaccord ou indiquait "ne sait pas", c'était principalement dû à une formulation ambiguë du cas problème et/ou de la recommandation.

Cette analyse des opinions de quatre (4) experts, nous a conduit à enlever de la base de connaissances le cas problème (sur 34) où trois experts sur quatre n'ont pas compris

l'explication du cas problème. Nous avons aussi corrigé (pour sept cas problèmes) les ambiguïtés soulignées par les experts.

Cette étape de vérification par les experts indique que notre approche est acceptable pour ces quatre (4) experts de la mesure fonctionnelle. Étant donné qu'il n'a pas été possible d'avoir plus de quatre (4) experts, on pourrait souhaiter refaire cette vérification avec un plus grand nombre d'experts afin d'obtenir un plus grand nombre de commentaires. Cependant, ces experts internationaux travaillent tous dans le secteur privé ayant une charge de travail considérable ; il est alors souvent difficile d'obtenir leur participation bénévole pour des essais de recherche.

Enfin, au niveau opérationnel, nous avons constaté que l'installation du prototype 1 sur des ordinateurs personnels posait un problème de compatibilité pour les ordinateurs n'ayant pas la version Windows 2000 ou XP. C'est surtout ce qui a occasionné une "perte de temps" que les experts consultés n'ont pas appréciée. Cette constatation nous a amené, d'une part, à revoir notre approche de construction du prototype 1 et, d'autre part, à revoir notre approche d'expérimentation avec les débutants. Ainsi, l'installation du prototype 1 pour les fins de l'expérimentation avec les "débutants" a été réalisée par des techniciens connaissant bien le prototype 1, plutôt que par les mesureurs débutants. Nous expliquons plus en détails, au Chapitre 6, notre approche de construction d'un nouveau prototype, appelé prototype 2.

5.2.2 Expérimentation auprès de mesureurs débutants avec le prototype 1

L'expérimentation qui suit a été conçue pour répondre à la deuxième question (Q2) : est-ce que le prototype 1 est utile au mesureur débutant ?

La vérification avec les experts est une étape préparatoire importante à l'expérimentation auprès des mesureurs débutants avec le prototype 1. Dans les "leçons apprises" nous avons déjà souligné que nous avons amélioré la formulation de sept cas problèmes et enlevé un cas problème trop ambigu. Nous avons aussi réalisé une autre étape préparatoire importante, soit

l'ajout de plusieurs cas problèmes à la base de connaissances du prototype 1. Le nombre de cas problèmes est passé de 34 à 105 dans la base de connaissances du prototype 1.

C'est avec cette base de connaissances "améliorée" que nous avons ensuite entrepris notre expérimentation avec les mesureurs débutants. Les cas problèmes ajoutés viennent pour la plupart d'études de cas déjà résolus par plusieurs experts ayant contribué à construire ces études de cas lors des recherches du Laboratoire de recherche en gestion du logiciel (LRGL) de l'UQÀM et de l'ÉTS pour la méthode de mesure COSMIC-FFP.

Il nous faut donc vérifier l'utilité du prototype 1 pour les mesureurs débutants (Q2). La question est la suivante : les mesureurs débutants ayant utilisé le prototype 1 obtiennent-ils de meilleurs résultats que les mesureurs débutants n'ayant pas utilisé le prototype 1 ? Voici les hypothèses que nous allons vérifier :

- A. l'utilisation du prototype 1 par un mesureur débutant permet d'obtenir de meilleurs résultats en obtenant un nombre moyen de bonnes réponses plus élevé (Hypothèse A) ;
- B. l'utilisation du prototype 1 par un mesureur débutant permet d'obtenir de meilleurs résultats en obtenant un nombre moyen de mauvaises réponses plus faible (Hypothèse B) ;
- C. l'utilisation du prototype 1 par un mesureur débutant ne permet pas d'obtenir de meilleurs résultats en obtenant un nombre moyen de bonnes réponses plus faible (Hypothèse C) ;
- D. l'utilisation du prototype 1 par un mesureur débutant ne permet pas d'obtenir de meilleurs résultats en obtenant un nombre moyen de mauvaises réponses plus élevé (Hypothèse D).

Dans l'expérimentation, nous avons demandé aux mesureurs débutants d'identifier les processus et les mouvements de données décrits dans la méthode de mesure COSMIC-FFP à

partir de la description textuelle des spécifications d'un logiciel connu des mesureurs débutants dans le cadre de leur cours (voir Annexe L).

5.2.2.1 Description de l'expérimentation

Pour que cette expérimentation soit valable, il faut éliminer, le plus possible, les facteurs qui pourraient faire varier les conditions du processus de mesure des mesureurs débutants. Afin de minimiser les possibles biais dans l'expérimentation, nous avons établi un processus visant à nous assurer du bon déroulement du mesurage, des conditions de mesurage, du choix des mesureurs ainsi que de l'analyse des résultats des mesures. Pour assurer le plus d'objectivité possible à l'expérimentation, nous l'avons répétée deux fois en deux jours consécutifs avec presque les mêmes conditions pour les deux groupes de participants (nous expliquons les variations plus loin), mais avec des participants différents pour chacun des deux groupes.

A) Voici le contexte de l'expérimentation :

- Avant le cours sur la méthode des points de fonction, il a été demandé aux étudiants s'ils étaient familiers avec la méthode de mesure COSMIC-FFP et s'ils avaient déjà utilisé cette méthode de mesure. Si des étudiants avaient répondu affirmativement à l'une des deux questions, ils auraient été éliminés de la sélection. Ce ne fut pas nécessaire ;
- tous les mesureurs débutants sont inscrits à la maîtrise en génie logiciel, programme conjoint de l'ÉTS et de l'UQÀM et tous les participants faisaient partie du cours sur les exigences (MGL-801) lors de l'expérimentation (trimestre d'automne 2002) ;
- les spécifications de l'étude de cas (voir Annexe L) étaient déjà connues des mesureurs débutants dans le cadre du cours de maîtrise MGL-801 ;
- l'étude de cas faisant partie de l'expérimentation ne se trouve pas parmi les cas du prototype utilisé par les mesureurs débutants, pas plus que des parties de ce cas.

B) Déroulement de l'expérimentation

- L'expérimentation elle-même comprenait deux étapes :
- une formation d'une durée de trois heures (le matin) ;
- une étape de mesurage qui a duré environ trois heures (l'après-midi du même jour) ;
- l'expérimentation s'est déroulée en deux jours consécutifs avec deux groupes différents ;
- les mesureurs débutants ne savaient pas quelle étude de cas serait utilisée pour la mesure avant le début de l'expérimentation. Ils en ont été informés uniquement au début de l'expérimentation ;
- au total, 25 mesureurs débutants ont participé à l'expérimentation ;
- tous les mesureurs débutants ont eu les mêmes spécifications du logiciel à mesurer ;
- tous les mesureurs débutants ont eu accès, pendant la durée de l'expérimentation, au Manuel de l'utilisateur COSMIC-FFP (copie papier et électronique) ;
- tous les mesureurs débutants étaient dans la même salle ;
- pour chaque groupe, un sous-groupe de mesureurs débutants avait accès au prototype et un autre sous-groupe de mesureurs débutants n'avait pas accès au prototype (voir Tableau 5-1) ;
- la répartition entre les deux sous-groupes a été établie au hasard juste avant l'expérimentation (voir Tableau 5-1 pour la répartition) ;
- les mesureurs débutants ayant accès au prototype ont eu droit à une assistance technique sur son fonctionnement pendant toute la durée de l'expérimentation. Nous voulions éviter que les problèmes techniques interfèrent dans l'expérimentation. L'assistance technique portait uniquement sur le fonctionnement du prototype et non sur l'identification des fonctionnalités ;

- la directive fournie aux deux assistants techniques était de ne pas répondre à des questions sur le contenu de l'étude de cas. Si une question était posée sur le contenu, la réponse était fournie à l'étudiant par l'expert (l'auteur de cette thèse) et transmise aussi aux autres étudiants ;
- le deuxième jour, les deux assistants techniques ont participé à l'expérimentation en tant que mesureurs débutants. Ce sont les mesureurs 24 et 28. Nous les avons exclus de l'analyse puisqu'ils ne correspondent pas à la définition de mesureur débutant tel que mentionné dans cette section.

C) Déroulement du mesurage et instructions aux participants :

- la durée de la partie mesurage de l'expérimentation a été de trois heures ;
- pour chaque groupe, au début de l'expérimentation, afin d'aider à démarrer le processus, l'expert a fourni un exemple de mesures de fonctionnalités du logiciel, soit sur la première étape d'une des 8 spécifications du cas (chaque spécification possède de 2 à 4 étapes) ;
- l'assistance technique se voulait immédiate (moins d'une minute) pour les mesureurs débutants ayant accès au prototype. Si cette condition a été bien respectée pour le premier groupe, elle n'a pu être respectée pour le deuxième groupe (attente de cinq minutes ou plus), les assistants de la première journée n'étant pas disponibles. L'expert a assuré l'assistance technique ;
- nous avons demandé aux mesureurs débutants d'identifier les concepts de la méthode de mesure COSMIC-FFP (frontière, processus fonctionnels, mouvements de données, groupes de données, etc.) à partir des spécifications écrites du logiciel. Cette identification des concepts est une activité d'identification d'exigences de l'utilisateur aux fins de la méthode de mesure COSMIC-FFP ;
- nous n'avons pas demandé aux mesureurs débutants de finaliser la mesure par l'addition des mouvements de données, ce qui équivaldrait à demander la taille du logiciel. En fait, demander uniquement la taille du logiciel limiterait notre

expérimentation puisque, dans la taille du logiciel, seuls les concepts relatifs aux mouvements de données sont additionnés.

D) Critères d'identification des bonnes et des mauvaises réponses :

- nous avons identifié les occurrences de concepts de la méthode de mesure COSMIC-FFP (l'Annexe K fournit le détail des résultats pour les occurrences de concepts de la méthode de mesure COSMIC-FFP) pour le calcul des résultats ;
- une réponse donnée par un mesureur débutant est considérée comme une bonne réponse si elle est en conformité avec les résultats de l'expert de la mesure fonctionnelle, lequel expert était aussi le concepteur de l'expérimentation et le formateur ;
- le mesureur débutant donne une mauvaise réponse lorsqu'il identifie dans le texte des concepts de la méthode de mesure COSMIC-FFP qui n'ont pas été identifiés par l'expert ;
- une mauvaise réponse n'est pas l'absence d'une bonne réponse.

Il faut noter que si le nombre de bonnes réponses est limité (67), le nombre de mauvaises réponses ne l'est pas, du moins en théorie.

5.2.2.2 Analyse des résultats de l'expérimentation auprès des débutants avec le prototype 1

A) Présentation des résultats

Le Tableau 5-1 présente un sommaire des résultats par participant (nombre de bonnes et de mauvaises réponses) selon que le participant avait ou non utilisé le prototype. Les deux groupes sont numérotés 1 et 2. Les mesureurs débutants du premier groupe sont numérotés de 1 à 14, alors que les mesureurs débutants du deuxième groupe sont numérotés de 16 à 27. Le mesureur 15 ne s'est jamais présenté à l'expérimentation, nous n'avons donc pu analyser ses résultats. Les mesureurs 24 et 28 ont été exclus du deuxième groupe puisqu'ils ne correspondent pas au profil du mesureur débutant tel que décrit à 5.2.2.1 D.

La répartition des mesureurs débutants ayant utilisé le prototype par rapport aux mesureurs débutants n'ayant pas utilisé le prototype est égale dans le premier groupe (7-7) et inégale dans le deuxième groupe (6-5).

Tableau 5-4 : Nombre de bonnes et de mauvaises réponses par participant et par groupe avec ou sans prototype

Mesureur	Groupe	Avec prototype	Bonnes réponses	Mauvaises réponses
2	1	Non	25	12
4	1	Non	50	36
5	1	Non	18	40
6	1	Non	24	47
10	1	Non	25	39
11	1	Non	14	44
13	1	Non	12	72
		Moyenne	24	41
		Écart-type	13	18
1	1	Oui	58	16
3	1	Oui	36	16
7	1	Oui	50	31
8	1	Oui	52	14
9	1	Oui	51	17
12	1	Oui	65	12
14	1	Oui	27	12
		Moyenne	48	17
		Écart-type	13	7
17	2	Non	36	48
19	2	Non	22	12
20	2	Non	41	37
21	2	Non	25	26
22	2	Non	44	12
		Moyenne	34	27
		Écart-type	10	16
16	2	Oui	56	36
18	2	Oui	42	19
23	2	Oui	34	19
25	2	Oui	21	12
26	2	Oui	26	9
27	2	Oui	23	11
		Moyenne	34	18
		Écart-type	13	10

B) Valeur de l'échantillon

Nous voulons principalement savoir si le prototype 1 est utile aux mesureurs débutants (Q2). À cet effet, nous allons tenir compte du groupe et vérifier l'additivité (H_0) entre les deux

groupes. En rapport avec cet échantillon, il faut se demander si l'utilisation ou la non-utilisation du prototype a le même effet pour les deux groupes. Une analyse de variance (ANOVA) avec deux facteurs a été réalisée séparément pour les bonnes réponses et pour les mauvaises réponses (voir Annexe J : *Univariate Analysis of Variance* BONS et MAUVAIS) pour l'ensemble des mesureurs débutants.

Pour les bonnes réponses, la valeur de F pour la relation entre les groupes et l'utilisation ou pas du prototype est de $F(1,21) = 5,900$ avec $p = 0,024$, soit inférieure à 0,05. La p-valeur se trouvant en dessous de 0,05, le résultat confirme qu'il y a un effet de prototype entre les deux groupes (EXP) de l'expérimentation même s'il n'est pas significatif sur la moyenne des deux groupes.

Les deux groupes de l'expérimentation étant différents, nous allons faire une analyse de ces deux groupes pris séparément en utilisant une analyse de contraste (ANOVA – CONTRAST). Nous avons fait quatre catégories.

Les deux premières catégories sont les suivantes :

- comparaison, pour le groupe 1, des deux sous-groupes (avec et sans prototype) ;
- comparaison, pour le groupe 2, des deux sous-groupes (avec et sans prototype) ;
- Les résultats sont les suivants :
 - pour le groupe 1, la valeur statistique pour le contraste 1 est de $t = -3,68$ avec $p = 0,001$, soit inférieure à 0,05. Le résultat confirme que les deux sous groupes (avec et sans prototype) du groupe 1 sont différents ;
 - pour le groupe 2, la valeur statistique pour le contraste 2 est de $t = 0,01$ avec $p = 0,99$, soit supérieur à 0,05. Le résultat confirme que les deux sous groupes (avec et sans prototype) du groupe 2 ne sont pas différents.

Les deux autres catégories comparent les non et les oui entre chaque groupe :

- comparaison entre l'expérience 1 et 2 des deux sous-groupes sans prototype ;
- comparaison entre l'expérience 1 et 2 des deux sous-groupes avec prototype.

Les résultats sont les suivants :

- pour les deux sous groupes sans prototype (non), la valeur statistique pour le contraste 3 est de $t = -1,32$ avec $p = 0,20$, soit supérieur à $0,05$. Le résultat confirme que les deux sous groupes sans prototype des groupes 1 et 2 ne sont pas différents;
- pour les deux sous groupes avec prototype (oui), la valeur statistique pour le contraste 4 est de $t = 2,317$ avec $p = 0,04$, soit inférieure à $0,05$. Le résultat confirme que les deux sous groupes avec prototype des groupes 1 et 2 sont différents.

Pour les mauvaises réponses, la valeur de F pour l'utilisation ou pas du prototype est de $F(1,21) = 10,233$ avec $p = 0,004$. La p-valeur se trouvant au dessus de $0,05$, le résultat confirme que les deux groupes ne sont pas différents et sont additifs (H_0) (annexe J).

Les résultats semblent indiquer qu'il n'y a pas de corrélation entre les bonnes et les mauvaises réponses puisque selon que l'on analyse les bonnes ou les mauvaises réponses, les deux expériences peuvent être additives (bonnes réponses) ou non additives (mauvaises réponses). Notre question est alors la suivante : est-ce qu'il y a une corrélation entre les bonnes et les mauvaises réponses de l'ensemble de l'échantillon ?

C) Corrélation entre les bonnes et les mauvaises réponses

Pour répondre à cette question nous utilisons les tests de corrélations de Pearson et de Spearman (Annexe J: Correlations).

Un premier test de corrélation (Pearson) indique qu'il n'y a pas de corrélation significative entre les bonnes et les mauvaises réponses : $r = -0,297$, $p = 0,189$ de l'ensemble de l'échantillon. Un deuxième test de corrélation (corrélation de Spearman qui tient compte du rang de la réponse) indique aussi qu'il n'y a pas de corrélation significative entre les bonnes et les mauvaises réponses : $r_s = -0,143$, $p = 0,475$ de l'ensemble de l'échantillon.

Nous avons réalisé une analyse de sensibilité en enlevant le cas dont la mauvaise réponse est la plus haute (72) dans l'échantillon afin de vérifier si la corrélation entre les bonnes et les mauvaises réponses est toujours non significative. À cet effet, nous avons refait les tests de Pearson et de Spearman. Les résultats sont les suivants : $r = -0,102$, $p = 0,475$ et $r_s = -0,040$, $p = 0,847$. Ces deux tests confirment à nouveau qu'il n'y a pas de corrélation significative.

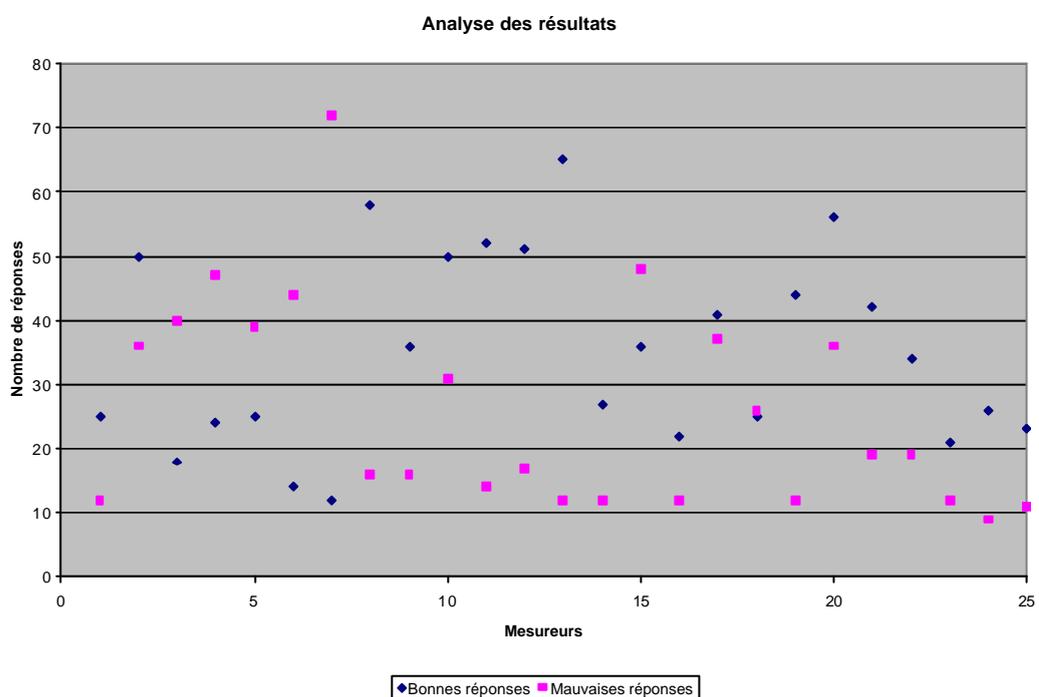


Figure 5-2 Répartition des bonnes et des mauvaises réponses par mesureur pour l'ensemble de l'échantillon

Nous présentons à la Figure 52 la corrélation entre les bonnes réponses et les mauvaises réponses par mesureur pour l'ensemble de l'échantillon.

Tout ces résultats confirment qu'il n'y a pas de corrélation entre les bonnes et les mauvaises réponses, même en enlevant de l'échantillon le mesureur ayant le plus grand nombre de mauvaises réponses (72).

D) Présentation des résultats des expérimentations

La Figure 5-3 présente les moyennes des résultats pour les bonnes réponses des deux groupes.

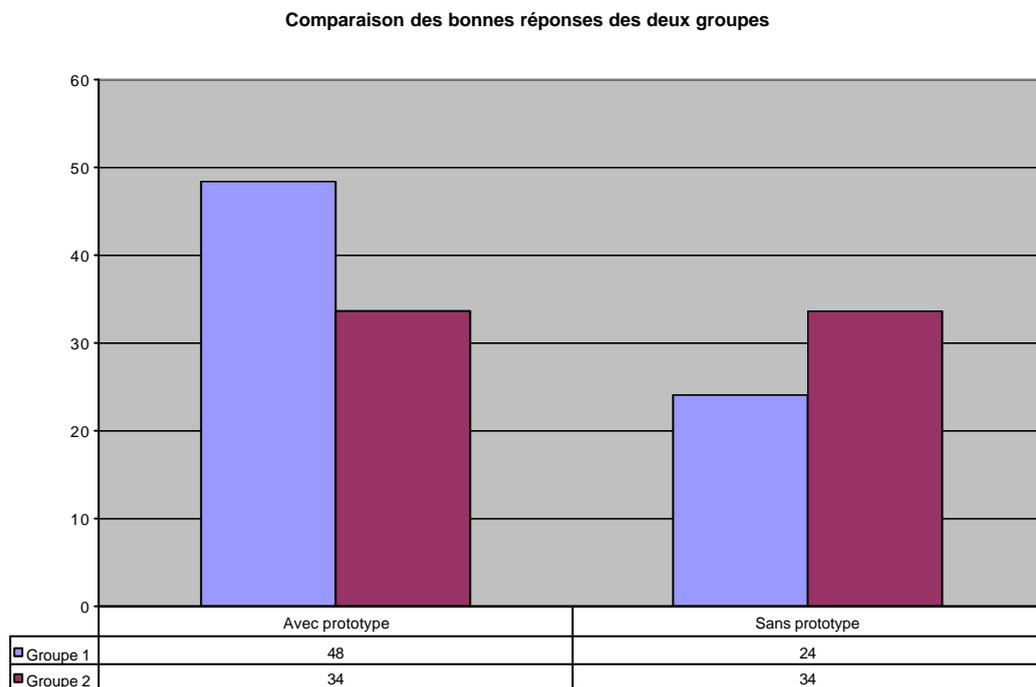


Figure 5-3 : Résultats par groupe du nombre moyen de bonnes réponses avec ou sans prototype

Les mesureurs débutants du groupe 1 qui ont utilisé le prototype 1 ont eu un plus grand nombre moyen de bonnes réponses que les mesureurs débutants du groupe 2 qui ont utilisé le prototype 1 (48 et 34 respectivement). Les mesureurs débutants du groupe 1 qui n'ont pas utilisé le prototype 1 ont eu un moins grand nombre moyen de bonnes réponses que les mesureurs débutants du groupe 2 qui n'ont pas utilisé le prototype 1 (24 et 34 respectivement).

On peut aussi constater que, pour le groupe 1, les mesureurs débutants qui ont utilisé le prototype 1 ont eu un plus grand nombre moyen de bonnes réponses que les mesureurs débutants qui n'ont pas utilisé le prototype 1 (48 et 24 respectivement). Pour le groupe 2, il n'y a pas de différence dans la moyenne entre les deux sous-groupes (avec et sans prototype), avec un écart-type plus grand pour ceux qui ont utilisé le prototype 1.

Pour les bonnes réponses, les deux groupes sont différents, ce que semble confirmer notre analyse de variance : $F(1,21) = 5,900$ avec $p = 0,024$, soit inférieure à 0,05.

Suite à l'analyse des bonnes réponses des deux groupes nous pouvons conclure que :

- le premier groupe confirme l'hypothèse A à l'effet que l'utilisation du prototype 1 permet au mesureur débutant d'obtenir de meilleurs résultats en obtenant un nombre moyen de bonnes réponses plus élevé.
- le deuxième groupe est différent et ne permet pas de confirmer ou d'infirmer aucune des quatre (4) hypothèses formulées plus haut.

La Figure 5-4 présente les résultats moyens pour les mauvaises réponses des deux groupes :

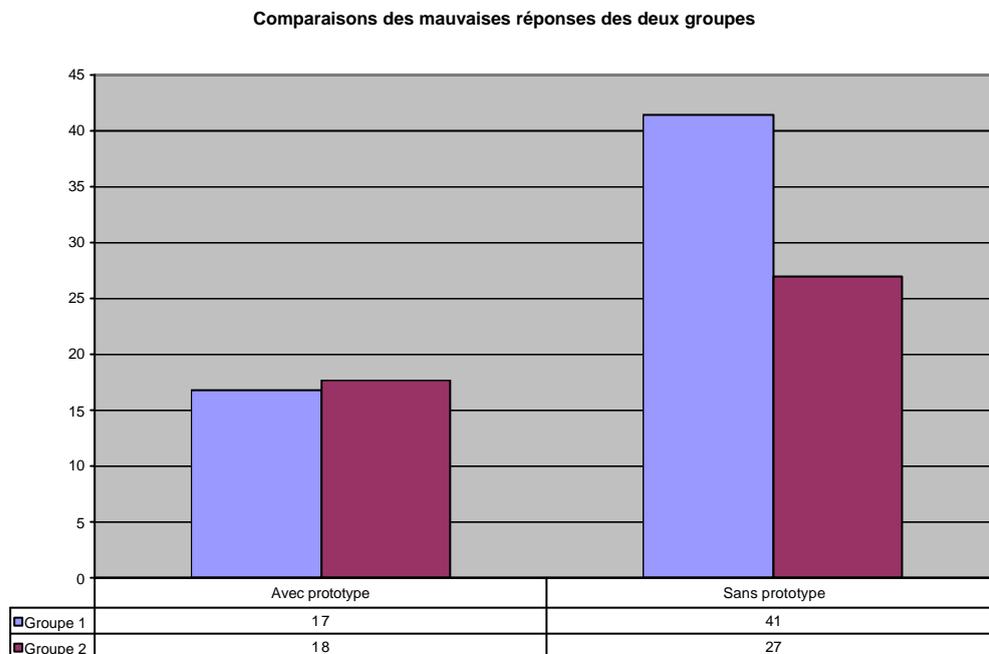


Figure 5-4 : Résultats par groupe du nombre moyen de mauvaises réponses avec ou sans prototype

Les mesureurs débutants du groupe 1 qui ont utilisé le prototype 1 ont eu à peu près le même nombre moyen de mauvaises réponses que les mesureurs débutants du groupe 2 qui ont utilisé le prototype 1 (17 et 18 respectivement). Les mesureurs débutants du groupe 1 qui n'ont pas utilisé le prototype 1 ont eu aussi un plus grand nombre moyen de mauvaises réponses que les mesureurs débutants du groupe 2 qui n'ont pas utilisé le prototype 1 (41 et 27 respectivement).

Pour les mauvaises réponses, si l'on considère uniquement l'effet "prototype", on peut constater que les mesureurs débutants qui ont utilisé le prototype 1 ont eu moins de mauvaises réponses que les mesureurs débutants qui n'ont pas utilisé le prototype 1.

Les résultats des deux groupes ne sont pas différents, ce qui est confirmé par l'analyse statistique : $F(1,204)=10,233$ avec $p=0,004$.

La Figure 5-4 et nos analyses statistiques nous permettent de conclure sur les mauvaises réponses uniquement.

Suite à l'analyse des mauvaises réponses pour les deux groupes, l'hypothèse B est confirmée, à l'effet que l'utilisation du prototype 1 permet au mesureur débutant d'obtenir moins de mauvaises réponses en obtenant un nombre moyen de mauvaises réponses plus faible.

En conclusion, l'hypothèse A est confirmée pour le premier groupe suite à nos analyses des bonnes et des mauvaises réponses. L'hypothèse B est confirmée pour les deux groupes à l'effet que le mesureur débutant qui utilise le prototype 1 obtient un nombre moyen de mauvaises réponses plus faible.

Les hypothèses C et D ne sont pas confirmées par les résultats.

E) Pourquoi les groupes 1 et 2 sont-ils différents dans le cas des bonnes réponses ?

Les résultats statistiques indiquent que la relation entre les bonnes réponses et l'utilisation ou pas du prototype 1 est significative pour le groupe 1, alors qu'elle ne l'est pas pour le groupe 2. Par contre, la relation entre les mauvaises réponses et l'utilisation ou pas du prototype 1 est significative pour les deux groupes.

La seule cause identifiée est relative au support technique entre les deux groupes. Il est possible que ce changement ait suffisamment influencé la nature de l'expérimentation et ainsi modifié les résultats. Des informations sur le nombre d'interventions techniques à chaque jour auraient pu nous aider, informations qui ne sont pas disponibles.

5.2.3 Observations sur la vérification avec les experts et les expérimentations avec les mesureurs débutants

La vérification avec les experts indique que les experts de la méthode de mesure COSMIC-FFP sont généralement d'accord avec les recommandations (Q1).

Est-ce que les expérimentations avec les mesureurs débutants, pour vérifier si le prototype 1 leur a été utile (Q2), peuvent être généralisées ?

Généralisation des résultats pour les mesureurs débutants

Est-ce que le prototype a été utile aux mesureurs débutants pour :

- identifier
- et vérifier si le concept identifié est bien un concept COSMIC-FFP ? (Q2)

Nos analyses statistiques indiquent que la relation d'interaction confirme que les deux sous-groupes de mesureurs débutants (avec et sans prototype) sont différents pour les bonnes réponses du groupe 1 { $F(1,21)=12,789$ avec $p=0,0014$ }, mais ne peut être confirmée pour les bonnes réponses du groupe 2. Pour les mauvaises réponses, la relation d'interaction (mesureurs débutants avec et sans prototype) est confirmée pour les deux groupes { $F(1,204)=10,233$ avec $p=0,004$ }.

Les résultats de nos expérimentations sont encourageants, mais la portée des résultats de cette expérimentation est limitée et ne peut être généralisée. Nous aurions souhaité que les deux groupes de l'expérimentation arrivent à des résultats significatifs tant pour les bonnes que pour les mauvaises réponses, ce qui n'a pas été le cas pour les bonnes réponses du groupe 2. Il serait donc souhaitable que d'autres expérimentations similaires soient réalisées avec un plus grand nombre de mesureurs débutants pour analyser la qualité statistique des expérimentations.

Généralisation pour les autres mesureurs

Nos résultats ne nous permettent pas de généraliser les conclusions avec des conditions expérimentales différentes, par exemple, avec des mesureurs ayant une meilleure expertise de la méthode de mesure COSMIC-FFP, c'est-à-dire ayant déjà mesuré quelques applications avec la méthode de mesure COSMIC FFP.

D'autre part, l'expérimentation n'a pas été aussi rigoureuse que nous l'aurions souhaité pour les deux groupes. Nous avons sous-estimé l'importance d'un soutien technique égal pour les deux groupes, ce qui, selon une de nos hypothèses, pourrait être la cause de résultats "mitigés" lorsque l'on compare les deux groupes pour les bonnes réponses. Nous croyons cependant que le processus proposé pourrait être repris d'une façon plus rigoureuse dans d'autres recherches et ainsi confirmer ou infirmer nos premiers résultats.

Expérimentation et domaine du génie logiciel

En informatique, l'expérimentation dans un environnement contrôlé, est plutôt restreinte.

Dans un article publié en 1998 sur l'expérimentation en informatique [74], Zelkowitz et Dolores écrivaient :

..computer science has not developed a concise taxonomy of methods for demonstrating the validity of new techniques [74, p. 23].

Les auteurs ont réalisé une étude des méthodes utilisées par les auteurs des articles dans plus de 600 publications dans la revue *IEEE* en 1985, 1990 et 1995. Les résultats indiquent que la majorité des auteurs n'utilisent pas de méthodes d'expérimentation : pas d'expérimentation ou assertion pour 60 % à 70 % des publications. La majorité des articles ne font pas état de méthodes d'expérimentation et se retrouvent dans la catégorie leçons apprises et études de cas (environ 10 % chacune). Les analyses statistiques sont utilisées dans moins de 2 % des cas.

Pour le domaine du génie logiciel (mesure fonctionnelle), nous avons mentionné quelques expérimentations à la section 2.1.9 (Rudolf, Low et Kemerer). Le nombre d'individus

impliqués dans notre expérimentation se compare avec le nombre d'individus des expérimentations réalisées par ces auteurs.

6. PRÉSENTATION ET VÉRIFICATION DU PROTOTYPE 2

La principale motivation qui nous a incité à réaliser le prototype 2 est de faciliter la vérification de la qualité des références et de réduire les redondances dans la documentation. Le prototype 2 est une conséquence des questions et des commentaires sur le prototype 1.

Le prototype 2 répond aux mêmes exigences que le prototype 1 du point de vue du mesureur, alors que les spécifications de maintenance de la base de connaissances par l'expert et l'administrateur n'ont pas été intégrées. Ainsi, l'interface du mesureur du prototype 2 ressemble à l'interface du mesureur du prototype 1 avec des améliorations mineures que nous expliquons dans la section 6.1.

Tel que mentionné, le prototype 2 ne possède pas d'interface pour l'expert et pour l'administrateur. Son entretien demande de la part de l'expert et de l'administrateur la connaissance d'un éditeur XML. Dans le cadre de cette recherche, c'est un inconvénient très mineur puisque la base de connaissances est maintenue par une seule personne.

Vous trouverez à l'Annexe C des informations sur les techniques de développement du prototype 2.

La base de connaissances du prototype 1 est composée de 950 fichiers de références en format RTF, ce qui rend difficile sa vérification et sa maintenance. Il nous faut alors penser à une réingénierie de la base de connaissances. En termes de réingénierie, il est possible de bâtir un :

- document de références permettant une visualisation des références à partir d'un seul fichier ;
- document de références permettant une visualisation des références à partir de plusieurs fichiers.

Nous expliquons l'approche de réingénierie à la sous-section 6.2.

6.1 Interface du mesureur

L'interface du mesureur a été modifiée afin de faciliter le choix des concepts topologiques, des cas problèmes et des thèmes après que le mesureur ait choisi un mot-clé. Le cheminement du mesureur est seulement facilité, il ne change pas. La Figure 6-1 montre qu'en entrant dans le prototype le mesureur peut faire l'entrée d'un mot-clé et commencer la recherche. Le menu apparaissant dans le haut de l'écran est le menu de l'expert. Le glossaire, les études de cas et la structure des connaissances (Xpert Tree) sont uniquement accessibles à l'expert. Par contre, le mesureur peut accéder à l'aide, tout comme l'expert, et envoyer des suggestions à l'expert. Pour sortir du prototype il suffit de cliquer sur *log out*.

À la Figure 6-1, on peut aussi lire que le mesureur a choisi le mot-clé *functional process* et a obtenu deux concepts topologiques avec des pourcentages de probabilité différents.



Figure 6-1 : Interface du mesureur (concept topologique)

L'interface du mesureur a été modifiée avec des "boutons radio" afin de permettre le choix d'un concept topologique avant que n'apparaissent les cas problèmes et les thèmes. Lorsque le mesureur choisit *triggering event* par exemple, les cas problèmes liés à *triggered events* apparaissent.

Help - Suggestion - Glossary - Case Studies - Xpert Tree - Logout

Keyword : Functional Process Search Definition ?

Topological Concepts		%
Functional Process		90
Triggering Event		50

Case problems		Case Study	%
How to identify a Triggering Event		Generic	95
Clock_Cycle signal		Valve Control System	90
Clock signal		Rice Cooker	90
User press a switch		Rice Cooker	90
Creating a report		Report Generator	90
User Trigger		Warehouse software portfolio	90

Themes		Facts	%
The trigger initiates one or more functional processes		No	90
The trigger occurs outside the boundary of the software measured		No	80
The trigger is instantaneous		No	80

4. Answer themes

Figure 6-2 : Interface du mesureur (cas problèmes et thèmes)

Dans les cas problèmes, le mesureur a choisi le bouton *user press a switch* (Figure 6-2), ce qui fait apparaître les thèmes (Figure 6-3). À droite des thèmes apparaissent les faits (réponses).

Themes		Facts	%
The trigger initiates one or more functional processes		Yes	90
The trigger occurs outside the boundary of the software measured		Yes	80
The trigger is instantaneous		Yes	80

OK

Answer		%
You did answer positively to all themes.		98

[See recommendation](#)

Figure 6-3 : Interface du mesureur (recommandation)

Lorsque le mesureur a donné une réponse (fait) à tous les thèmes, il clique sur OK et il obtient une recommandation avec un pourcentage.

De plus, ce prototype a été construit pour fonctionner sur le WEB, ce qui permet à un plus grand nombre de personnes d'accéder au prototype sans avoir à installer le logiciel.

6.2 Approche de réingénierie

Il est possible d'atteindre notre objectif de vérification des fichiers sans nous occuper de la redondance, mais le principal inconvénient est que le nombre de fichiers reste le même. Réduire le nombre de fichiers, si la réduction est significative, devrait faciliter la maintenance de la base de connaissances.

Comment la vérification des fichiers sera-t-elle facilitée ? En créant différents types de fichiers avec une structure définie à l'avance. C'est ce que permet entre autres le format XML défini par le *World Wide Web Consortium*²¹ (W3C). Cette section décrira en détail les différents types de fichiers retenus et leur structuration.

Comment pourra-t-on éviter la redondance des fichiers ? Nous allons créer un document de référence (une vue) indépendant des différents fichiers de format XML. Il y a une redondance dans les fichiers pour deux raisons :

- d'un fichier à l'autre, les mêmes types d'informations reviennent (ex. : les définitions sont utilisées dans tous les fichiers) ;
- d'un fichier à l'autre, les mêmes informations reviennent (ex. : pour une étude de cas, l'information sur le contexte du cas revient dans plus d'une étude de cas).

Pour atteindre ces objectifs nous avons procédé en différentes étapes :

- choisir un langage formel (6.3) ;
- définir les types de fichiers et leurs relations (6.4) ;
- définir les types de données dans la base de connaissances (6.5) ;

²¹ <http://www.w3.org/>

- réaliser la transformation des fichiers (6.6) ;
- détecter les redondances (6.7) ;
- définir une nouvelle organisation de l'information – sans redondance (6.8) ;
- définir les vues (6.9) ;
- traduire les documents (6.10).

6.3 Choix du langage formel

Le langage formel choisi est le XML. En plus de répondre à nos besoins de structuration des fichiers, ce langage est une norme reconnue internationalement grâce au W3C. Plus spécifiquement, nous avons choisi une série de normes définies par le W3C, normes qui nous ont permis d'utiliser différents types de fichiers pour passer des fichiers de type RTF aux fichiers de type XHTML.

6.4 Définition des types de fichiers et leurs relations

La Figure 6-4 présente les différents types de fichiers et les relations qu'ils ont entre eux.

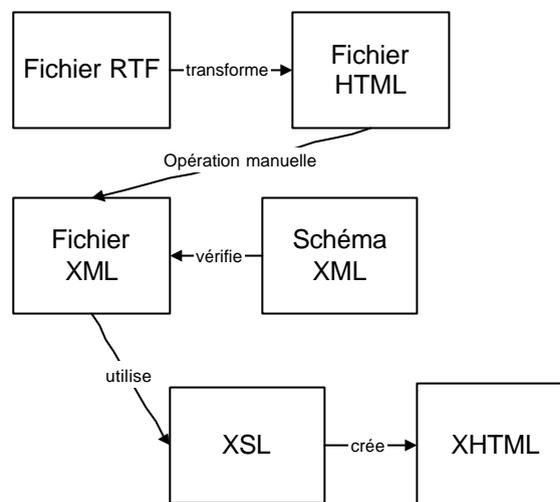


Figure 6-4 : Présentation des types de fichiers

6.4.1 RTF et HTML

Le format RTF est une méthode d'encodage de textes et de graphiques reconnu par les systèmes d'opérations tels MS/DOS et Windows de Microsoft, OS/2 d'IBM et Apple MacIntosh. Le RTF permet de transférer du texte ASCII d'une plate-forme à une autre, mais sans format précis.

Le HTML est un langage spécialisé pour le WEB. En tant que langage, il a une structure précise. En fait, il a été conçu à l'origine pour représenter la structure d'un document et non le document lui-même. Ce sont les manufacturiers des navigateurs qui ont ajouté des éléments de présentation dans le langage, ce qui le rend très dépendant d'un navigateur particulier.

6.4.2 XML

Le format XML a été défini à l'origine pour la publication à large échelle de documents électroniques. XML est un méta-langage facilitant la création de langages. Chaque schéma XML définit un nouveau langage basé sur le méta-langage XML.

Différentes normes accompagnent le format XML avec des objectifs différents. Trois normes sont intéressantes dans le cas de vérification de la base de connaissances : XML schéma, XSL et XHTML.

6.4.3 XML schéma

Cette norme permet de décrire la structure, le contenu et la sémantique d'un document XML. Avec cette norme, la vérification de la structure des documents composant la base de connaissances peut être automatisée.

Le schéma XML est une norme toujours en développement, mais on trouve sur le marché des outils qui permettent de l'utiliser.

6.4.4 XSL

XSL est un langage permettant de décrire comment doit être affiché un document XML d'un certain type. Il est important de pouvoir référencer les éléments dans un document XML. De cette façon, on peut avoir un seul fichier pour la définition des concepts et aller chercher le (ou les) concept(s) que l'on veut présenter dans le document utilisateur.

6.4.5 XHTML

XHTML est une reformulation de HTML qui respecte le méta-langage XML et qui peut-être interprétée par un navigateur Internet²². Afin de conserver la structure des textes contenus dans les documents, un sous-ensemble du langage XHTML est utilisé. Il permet de définir la structure en paragraphe et de définir l'emplacement des images à insérer. Un sous-ensemble du langage XHTML est utilisé dans tous les types de documents. Le langage de présentation XSL peut produire une vue en XHTML pouvant être utilisée dans la plupart des navigateurs.

Le fichier RTF ne possède pas de langage de structuration, c'est une méthode d'encodage. Il existe des convertisseurs du langage RTF au langage HTML. Nous avons utilisé le convertisseur de Microsoft (CZ Document Converter V1.5) de façon à ce que le navigateur Internet Explorer puisse lire le fichier HTML. Il est ensuite possible de créer un fichier XML dont la structure sera vérifiée par le schéma XML correspondant au type de fichier souhaité. Enfin, une procédure écrite en XSL permet de créer une vue du fichier XML en le transformant en XHTML tel qu'illustré à la Figure suivante.

6.5 Description des types de données de la base de connaissances

La base de connaissances est constituée de différents types de documents qui sont en fait les données de la base de connaissances et que nous appelons "types de données". Les types de données sont, entre autres, les mots-clés, les concepts topologiques, Les cas problèmes, Les thèmes et Les recommandations compris dans le modèle du prototype 2 (6-5). Pour chacun de ces concepts, nous avons défini un type de données compris dans la base de connaissances. Enfin, pour chacun de ces concepts, le mesureur a accès à sa définition, également selon un format spécifique. Il y a donc six normes de structuration ou six types de données dont cinq

²² XHTML n'est pas encore supporté par la plupart des navigateurs Internet mais les documents en format XHTML peuvent être quand même visualisés par ceux-ci grâce à certains trucs expliqués sur le site W3Schools.

se retrouvent dans le modèle du prototype 2 (Figure 6-5). Le "fait" étant, pour le prototype, seulement un oui ou un non, nous l'avons exclu du modèle. Les études de cas ne font pas partie de l'ontologie COSMIC-FFP. Ils représentent les logiciels à mesurer.

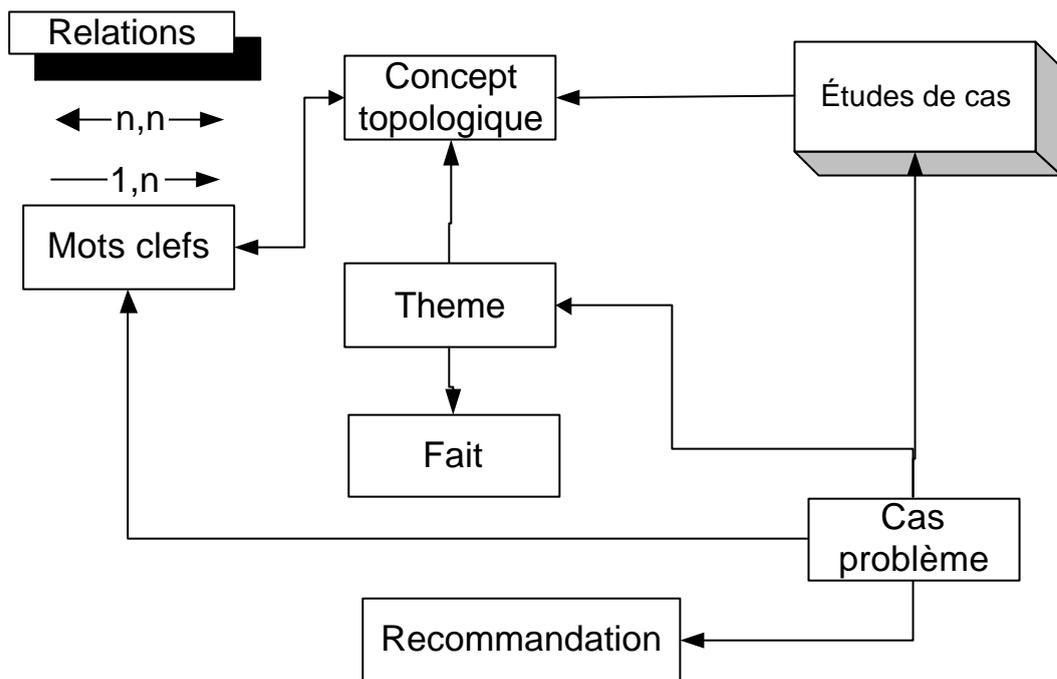


Figure 6-5 : Modèle du prototype 2

Les étapes de la première possibilité de description des documents sont :

- de définir un sous-ensemble XHTML (TextHTML²³, Inside P) et les concepts du prototype : mot-clé, concept topologique, cas problème, thème et recommandation et

²³ TextHTML est un type utilisé dans les schémas XML des concepts du prototype. Il est constitué d'un sous-ensemble du langage XHTML. Les schémas des concepts du prototype sont construits à partir du type TextHTML et d'autres éléments construits spécifiquement pour le prototype.

- de traduire les fichiers de type RTF en XML.

Dans cette sous-section, nous traitons du premier point. Afin de retrouver les paragraphes, les listes et les images des documents RTF originaux dans la "vue" créée par le XSL, il faut pouvoir représenter ces entités en format XML. Pour cela, nous avons défini un type de données²⁴ appelé **TextHTML** basé sur XHTML. Le type **TextHTML** est utilisé dans la description de tous les types de données. Les types de données²⁵ font référence aux informations de la base de connaissances définies dans TextHTML, soit, par exemple, le mot-clé, le concept topologique et insideP. Voici une représentation graphique (Figure 6-6) du type TextHTML générée par l'outil XMLSpy [75] :

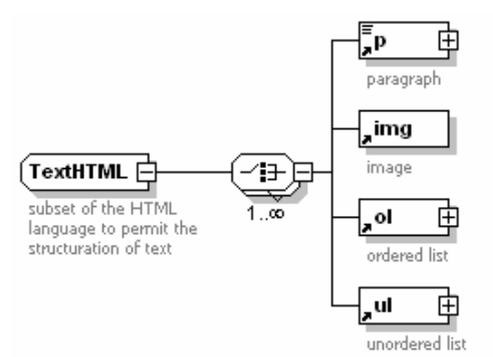


Figure 6-6 : Type TextHTML

Chaque rectangle dans la Figure 6-6 représente un élément XML pouvant être utilisé pour décrire une partie de document dans le langage. Chaque élément peut contenir un type de données.

24 Les types de fichiers sont, par exemple, RTF, HTML et XML.

25 On pourrait objecter ici que le terme types de documents serait plus approprié car dans un document XML, car « type de données » représente un entier ou une chaîne de caractères dans un élément. Voir première phrase de 4.7.5.

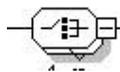


Figure 6-7 : Choix

La Figure 6-7 représente un choix. Ce mécanisme, dans un schéma, signifie que l'élément ou le type à gauche peut contenir n'importe quel élément se trouvant à droite. Une cardinalité peut être associée à ce symbole et représente le nombre minimal et maximal d'élément de droite contenu dans l'élément de gauche.

TextHTML est un type complexe de données, car il peut contenir d'autres éléments ayant un type de données.

Le type de données **TextHTML** peut donc contenir un à plusieurs éléments. Voici, à titre d'exemple (Tableau 6-1), les éléments de **TextHTML** avec une description, le type de données et les attributs pour chaque élément (lorsque nécessaire).

Tableau 6-1 : Éléments de textHTML

Élément	Description	Type de donnée	Attributs	
p	Délimite un paragraphe	InsideP	align	Détermine la position horizontale du paragraphe (<i>left, center, right</i>)
img	Détermine l'emplacement d'une image à insérer	Vide	src	URL de l'image à insérer
			height	Hauteur de l'image
			width	Largeur de l'image
ol	Liste ordonnée			
ul	Liste non ordonnée			

Ces informations se répètent pour tous les autres types de données en plus de **InsideP** : mot-clé, concept topologique, thème, recommandation.

6.5.1 InsideP

InsideP est un type complexe dit mixte car il peut contenir du texte et divers éléments.

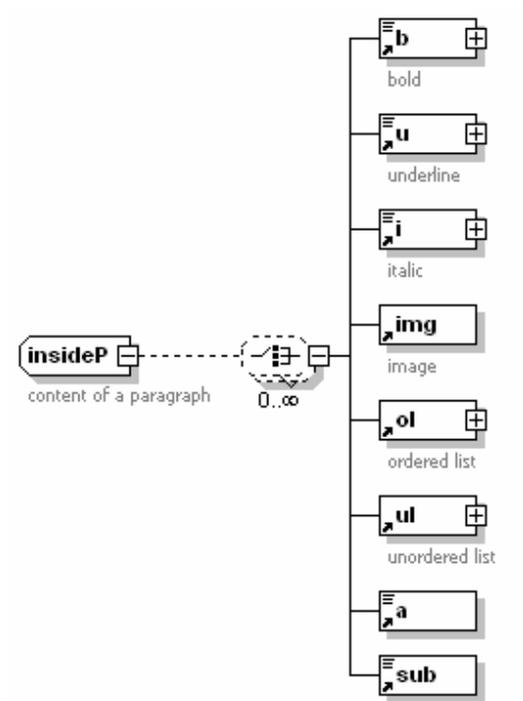


Figure 6-8 : Type de données InsideP

Le type de données InsideP de la Figure 6-8 décrit le contenu d'un paragraphe. Dans un paragraphe, un mot peut être en gras, en souligné ou en italique. On peut aussi avoir une image et une liste ordonnée ou non ordonnée d'éléments. Voici un exemple de liste ordonnée à la Figure 6-9. Dans l'élément liste ordonnée, on a la liste ordonnée, une séquence et des éléments ou des items dans la liste de 1 à n.

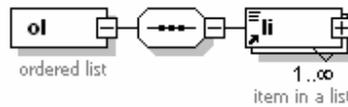


Figure 6-9 : Liste ordonnée

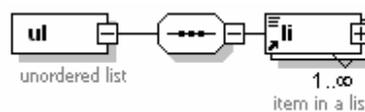


Figure 6-10 : Liste non ordonnée

Dans l'élément liste non ordonnée (Figure 6-10), on a : la liste non ordonnée, une séquence et des éléments ou des items dans la liste de 1 à n. Voici un exemple à la Figure 6-11 :

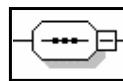


Figure 6-11 : Séquence

La Figure 6-11 représente une séquence. Ce mécanisme, dans un schéma, signifie que l'élément de gauche peut contenir une suite ordonnée d'éléments définis à droite du symbole.

L'exemple du Tableau 6-2 est le texte de la définition du mot-clé *Persistence of data group*. La définition est composée de deux paragraphes dont le dernier contient une liste. Ce texte est une instance du type TextHTML défini par le schéma représenté à la Figure 6-6.

Tableau 6-2 : Persistence of Data Group

<p><p></p> <p><i>The persistence of a data group is a quality describing the how long the data group is retained in the context of the Functional User Requirements. Three types of persistence are defined : transient (exists only for the life of a functional process), short (beyond the life of a functional process for as long as the software is operational) and indefinite (beyond the duration of operation of the software).</i></p> <p></p></p> <p><p></p> <p><i>There are three sets of data groups to be identified and classified :</i></p> <p></p> <p><i>Data Groups representing stored (persistent) data</i> </p> <p><i>Data Groups representing data crossing the boundary inbound</i> </p> <p><i>Data Groups representing data crossing the boundary outbound</i> </p> <p></p> <p></p></p>
--

Voici les schémas des autres éléments avec une brève explication.

6.5.2 Mot-clé

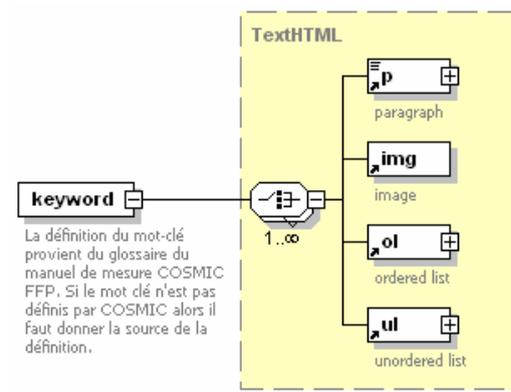


Figure 6-12 : Type de données Mot-clé

Le type de données Mot-clé de la Figure 6-12 est relativement simple. Il donne une définition provenant de la méthode de mesure COSMIC-FFP ou d'une autre source avec sa référence. Il comprend un paragraphe, une possibilité d'image et une liste ordonnée ou non ordonnée d'items.

6.5.3 Concept topologique

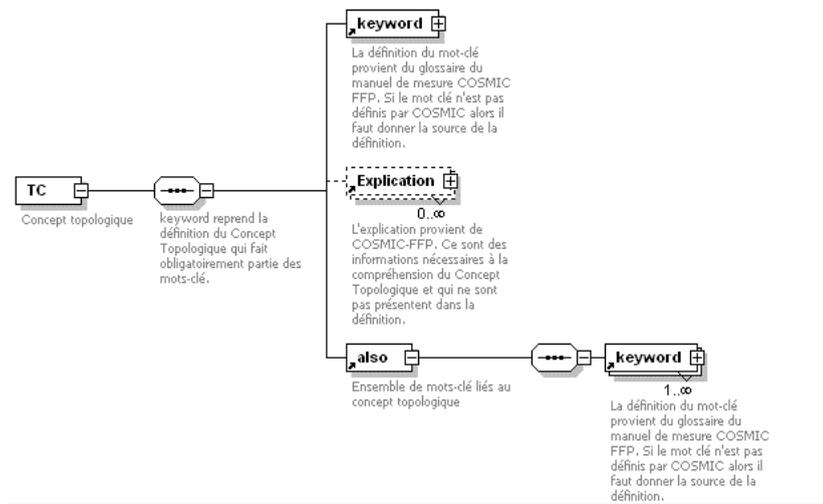


Figure 6-13 : Type de données Concept topologique

À la Figure 6-13, le type de données Concept topologique comprend le nom du concept topologique, le nom du mot-clé, la référence de la définition, le titre de l'explication et la possibilité d'avoir d'autres mots-clés.

6.5.4 Cas problème

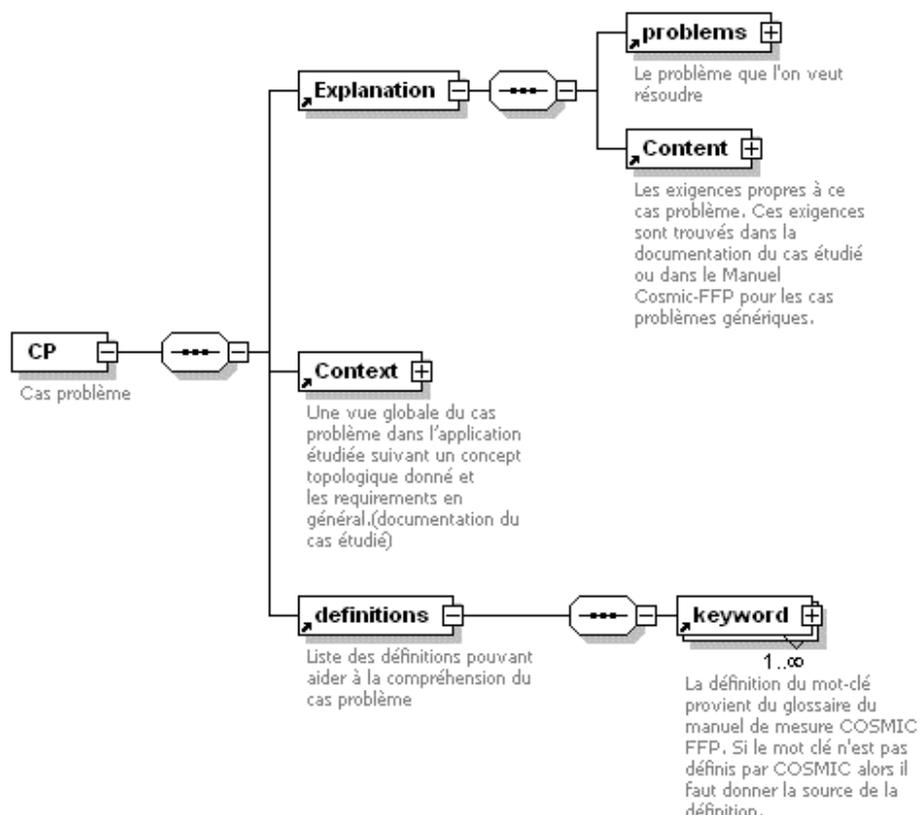


Figure 6-14 : Type de données Cas problème

Le type de données Cas problème de la Figure 6-14 comprend le nom du cas problème avec :

- son explication ;
- la description du problème que l'on veut résoudre et son contenu.

Il comprend aussi le contexte (vue globale du cas) et des définitions correspondant aux mots-clés.

6.5.5 Thème

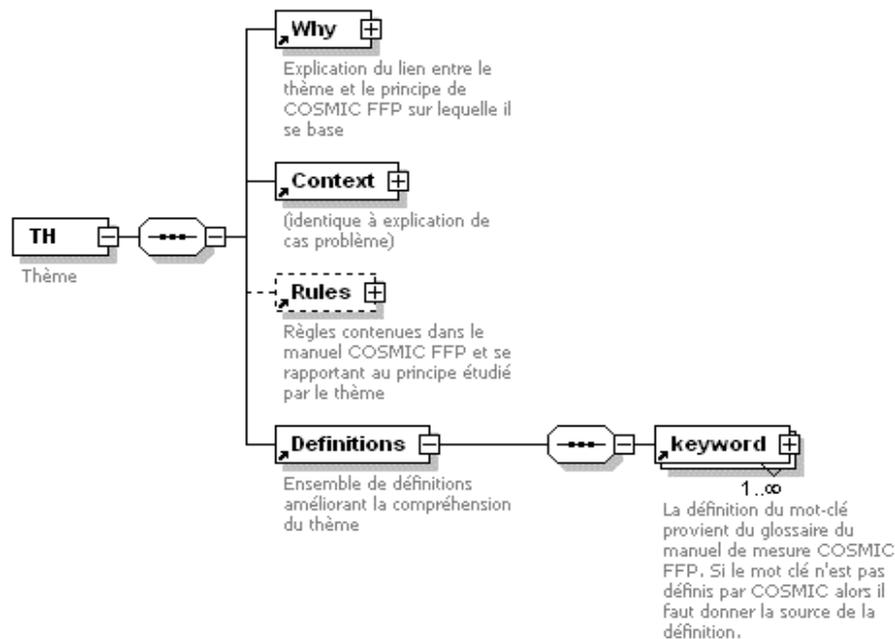


Figure 6-15 : Type de données Thème

Le type de données Thème de la Figure 6-15 explique d'abord le lien entre le thème et le principe{ XE "principe" } de la méthode de mesure COSMIC-FFP{ XE "COSMIC-FFP" } sur lequel se base le thème. Dans le thème, on trouve aussi le contexte (voir cas{ XE "cas" } problème), les règles{ XE "règles" } se rapportant au principe lié au thème et enfin les définitions (voir mot-clé).

6.5.6 Recommandation

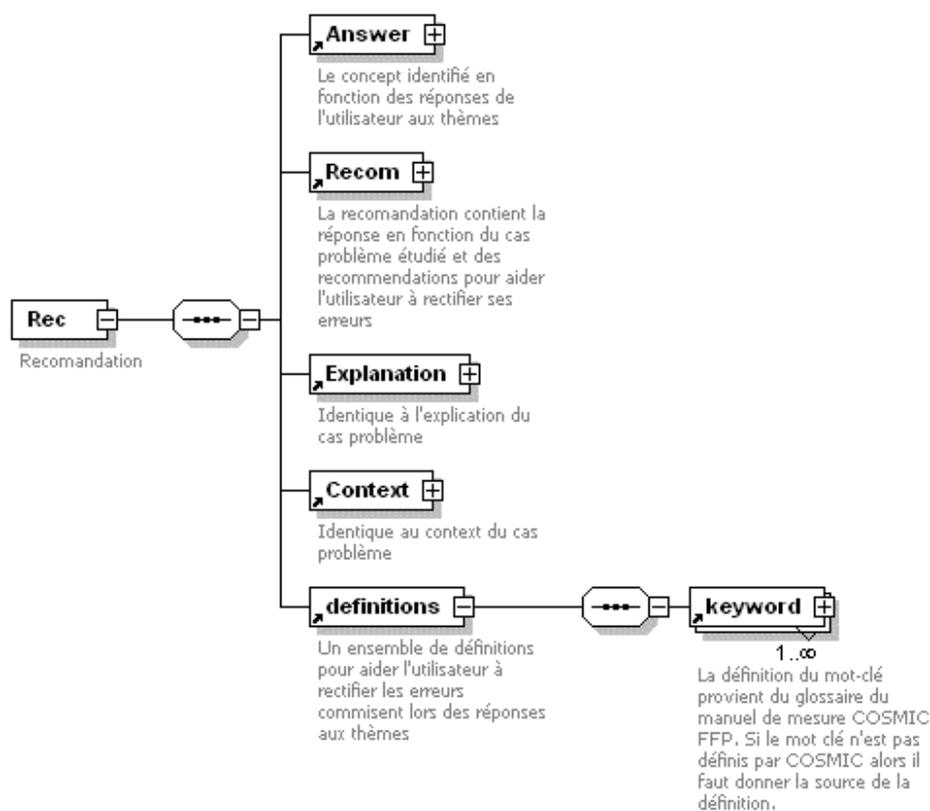


Figure 6-16 : Type de données Recommandation

Le type de données Recommandation de la Figure 6-16 fournit une réponse en rapport avec les faits fournis par le mesureur à chacun des thèmes associés au cas problème étudié. Il y a ensuite la recommandation, une explication du pourquoi de cette recommandation, un contexte et des définitions (voir mots-clés).

6.6 Procédure de transformation

Suite à la définition formelle de tous les types de données (schémas XML), il est possible de traduire les fichiers RTF dans les langages définis par les schémas des types de données.

Cette étape est difficilement automatisable lorsque les fichiers RTF ne suivent pas un modèle ayant des styles prévisibles d'un fichier à un autre. C'est le cas avec les 950 fichiers de référence du premier prototype. La transformation s'est réalisée en deux étapes :

- les fichiers RTF ont été traduits automatiquement du RTF au HTML de façon à ne conserver que les éléments autorisés par le type TextHTML ;
- les fichiers HTML ont ensuite été transformés manuellement pour respecter les différents schémas XML en fonction du type de données.

6.6.1 1° Cycle

Input :

- 950 fichiers RTF sans structure rigoureuse mais avec une similitude assez marquée (difficile à vérifier) ;
- base de données ACCESS reprenant les liens entre les fichiers RTF ;
- modèle de données COSMICXpert (Figure 6-5).

Processus :

- vérification des définitions et corrections des erreurs (inspection) ;
- construction d'un gabarit (Norme de Structuration) (inspection) ;
- restructuration des fichiers RTF suivant le gabarit construit (inspection).

Output :

- Un ensemble de fichiers RTF respectant une charte de structuration construite à partir du modèle de données COSMICXpert (Figure 6-5).

6.6.2 2° Cycle

Input :

- 950 fichiers RTF avec une structure suivant une norme créée définissant la structure et le contenu des fichiers RTF ;
- gabarit à respecter permettant une vérification ;
- modèle de données COSMICXpert (Figure 6-5) ;
- base de données reprenant les liens entre les fichiers RTF.

Processus :

- représentation²⁶ des gabarits en schéma XML respectant le modèle de données COSMICXpert (Figure 6-5) ;
- intégration des liens relatifs à la base de données dans les fichiers XML permettant de visualiser la structure des liens entre les différents concepts de COSMICXpert ;
- transformation des fichiers RTF en fichiers HTML permettant de respecter la mise en page des fichiers RTF existants (Gras – souligné – ...). Cette transformation se fait automatiquement ;

²⁶ Les gabarits sont une vue graphique des schémas XML. C'est pourquoi on écrit représentation et non transformation.

- transformation des fichiers HTML en fichier XML respectant le schéma (inspection) ;
- vérification de la structure syntaxique des fichiers XML et correction des erreurs de syntaxe ;
- vérification de la structure des fichiers XML avec le schéma XSD correspondant et correction des erreurs liées à la structuration XML ne respectant pas la structure du fichier XSD (par exemple un tag inconnu, mal formulé, ...).

Output :

- un ensemble de fichiers XML respectant la structure des données et le modèle de données COSMICXpert (Figure 6-5).

6.6.3 3° Cycle**Input :**

- 950 fichiers XML ;
- raffinement d'un gabarit permettant d'éviter la redondance, une structuration, une vérification automatique des concepts liés au manuel COSMIC-FFP.
- Modèle de données de COSMICXpert (Figure 6-5) plus raffiné.

Processus :

- raffinement du modèle de données COSMICXpert (Figure 6-5) ;
- transformation des schémas XML ;
- automatisation des liens permettant d'éviter la redondance et d'effectuer une meilleure vérification des concepts et une meilleure maintenance du système.

Output :

- un ensemble de fichiers XML respectant la structure des données et le modèle de données COSMICXpert (Figure 6-5).

6.7 Détecter les redondances

La redondance peut provenir de deux causes. La première cause se trouve dans la description des types de données. On retrouve un élément commun entre différents types de données. C'est le cas pour l'élément "définition" qui se trouve dans tous les types de données. La deuxième cause de redondance se retrouve dans le contenu des différentes références pour les cas problèmes de même nature et appartenant à la même étude de cas.

La Figure suivante met en évidence la redondance dans les types de données par rapport aux éléments. Une des techniques que l'on peut utiliser pour détecter la redondance est de construire un diagramme en reprenant les différents types de données et en les reliant entre eux.

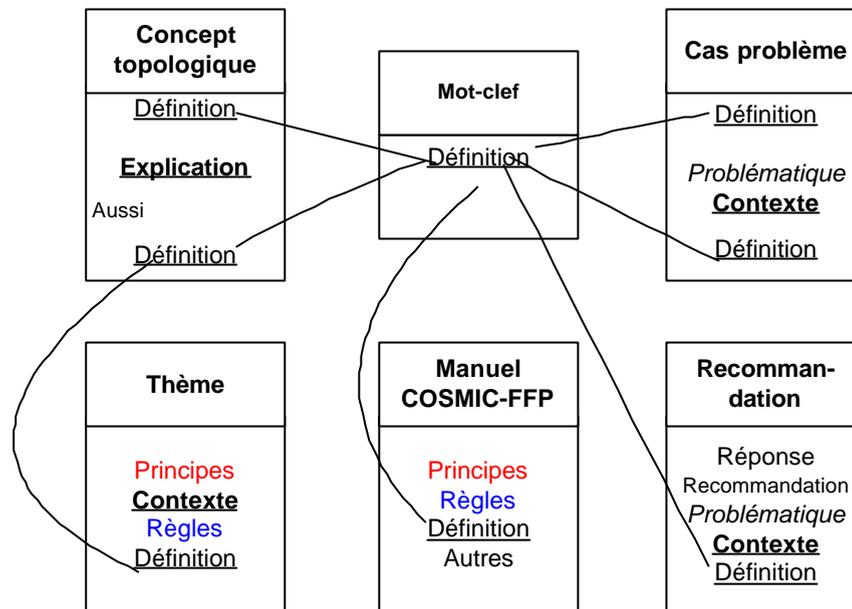


Figure 6-17 : Éléments redondants des types de données

La Figure 6-17 fait ressortir les éléments redondants des types de données. Le concept “définition” est utilisé huit fois, ce qui veut dire qu’il est possible de créer un seul fichier avec le mot “définition” plutôt que d’avoir à le transcrire huit fois, ce que nous faisons dans le premier prototype. Dans le prototype 2, nous n’avons pas, pour le moment, de manuel COSMIC-FFP. Il a été placé dans la Figure 6-17 afin de montrer la redondance potentielle entre le Manuel COSMIC-FFP et les autres types de données. Les autres éléments redondants sont : contexte (3), principe (2), règles (3) et problématique (2).

La deuxième cause de redondance se retrouve dans le contenu des différentes références pour les cas problèmes de même nature et appartenant à la même étude de cas. Par exemple, dans une étude de cas, le contexte est toujours le même dans la description du cas problème pour un concept topologique spécifique. Nous avons constaté la même redondance dans les thèmes. Les thèmes visant à identifier le même concept topologique étaient presque tous identiques. En associant les thèmes aux concepts topologiques plutôt que de les associer à chaque cas problème, nous réduisons le nombre de fichiers.

6.8 Définir une nouvelle organisation de l'information

La deuxième façon de définir les types de documents décrits dans la charte de structuration consiste à voir ces types de documents comme des vues sur une nouvelle organisation de l'information. Il faut donc définir cette nouvelle organisation de l'information.

Pour le prototype 1, c'est la base de donnée (ACCESS) associée au mode de référence du prototype 1 qui se chargeait des liens entre les fichiers RTF. Les informations contenues dans cette base de données ont été reprises dans le prototype 2.

La base de données comprend les pourcentages (intervalle de confiance) qui établissent les liens entre les différents concepts et les informations qui permettent, en fonction des faits choisis (*fact*), de déterminer la recommandation à proposer à l'utilisateur.

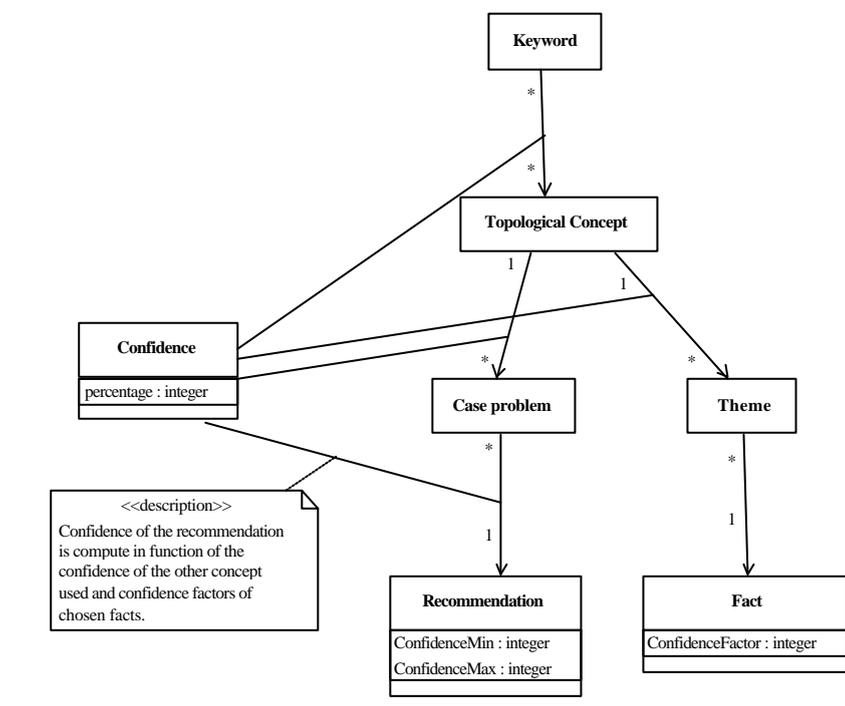


Figure 6-18 : Liens contenus dans la base de données

6.8.1 Unités de stockage

Les unités de stockage (Figure 6-18) après réingénierie restent identiques aux unités de stockage avant réingénierie suite à la procédure de transformation décrite plus haut. Par exemple, dans la base de connaissances composée de fichiers RTF, il y a un fichier par mot-clé, un par concept topologique, un par cas problème, un par thème et un par recommandation. C'est toujours vrai après la transformation; il y a un fichier par vue utilisateur.

Avec la possibilité de dissocier les vues et les types de données, il est possible de réduire considérablement le nombre de fichiers tout en conservant le nombre de vues en enlevant les redondances détectées ci-dessus. Le nombre de fichiers XML (ou RTF pour le prototype 1) passe de 950 s à moins de 150 pour une centaine de cas.

En rassemblant les concepts apparentés dans un même type de fichier XML, il est possible de créer un schéma XML pour définir une nouvelle structure et ainsi créer un nouveau type de documents.

Pour permettre la réduction du nombre de fichiers, nous avons identifié cinq types de documents :

- glossaire
- concept topologique
- cas problème
- étude de cas
- expert.

Les différents types de documents contiennent des liens vers les autres types de documents. Ces liens sont souvent représentés par une URL. La Figure 6-19 reprend les liens entre les types de documents.

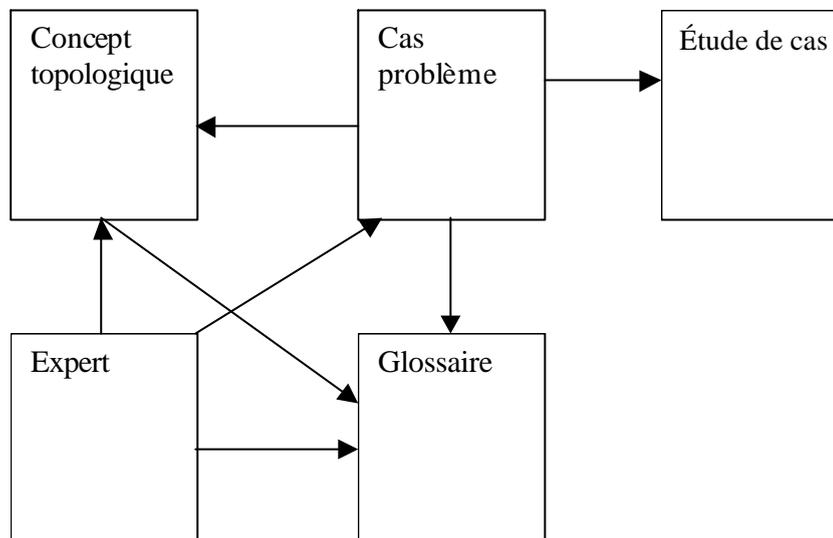


Figure 6-19 : Liens entre les types de fichiers

Le type de document **Glossaire** reprend tous les mots-clés que l'on rencontre dans la base de connaissances avec leur définition.

Le type de document **Concept topologique** comprend toutes les informations qui concernent le concept topologique, ses thèmes et ces faits associés.

Le type de document **Cas problème** comprend les informations sur le cas problème et les recommandations associées à ce cas problème.

Le type de document **Étude de cas** comprend plusieurs formulations des spécifications de l'étude de cas classées par concept topologique à identifier. Ce type de document peut contenir également des informations générales sur l'étude de cas.

6.8.2 Définition des nouveaux types de documents

Glossaire

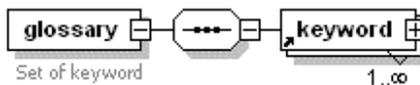


Figure 6-20 : Type de document Glossaire

Voici la description du glossaire avec XMLSPY (Tableau 6-3) :

Tableau 6-3 : Documentation du type de document Glossaire

Élément	Description	Type de données	Attributs	
glossary	Délimite le document glossaire			
keyword	Délimite un mot-clé et sa définition	TextHTML	id	Identifiant et permettant de retrouver le mot-clé dans le glossaire
			name	Nom du mot-clé
			ref	Référence de la définition

Dans cette documentation on retrouve le nom de l'élément (ici *glossary* et *keyword*), leur description, le type de données, les attributs de l'élément et une description de l'attribut. Tous les types de documents ont ces informations dans XMLSPY.

6.8.3 Concept topologique

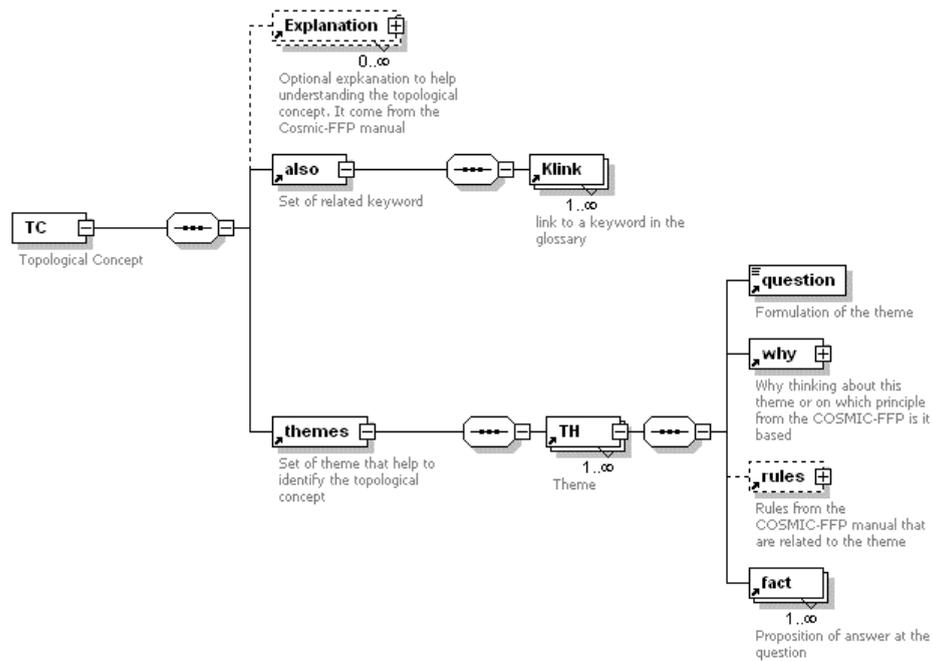


Figure 6-21 : Type de document Concept topologique

Tableau 6-4 : Documentation du type de document Concept topologique

Élément	Description	Type de données	Attributs	
TC	Délimite le document Concept topologique		name	Nom du concept topologique
			defID	Identifiant le mot-clé définissant le concept topologique
Explanation	Délimite une éventuelle explication de concept topologique	TextHTML	name	Titre de l'explication
also	Délimite une série de mots-clés relative au concept topologique			
Klink	Représente un lien vers	vide	idref	Identifiant le mot-clé lié dans le

Élément	Description	Type de données	Attributs	
	un mot-clé du glossaire			glossaire
Themes	Délimite les thèmes associés au concept topologique			
TH	Délimite un thème		id	Identifiant du thème
			cf	Pourcentage du lien entre le thème et son concept topologique
question	Délimite la formulation du thème en question	Text		
why	Délimite l'explication de la présence du thème	TextHTML		
rules	Délimite d'éventuelles règles apparentées au thème et provenant du manuel de mesure de COSMIC-FFP	TextHTML		
fact	Délimite un fait	vide	name	Nom du fait
			CF	Facteur permettant le calcul pour déterminer la recommandation à proposer

6.8.4 Cas problème

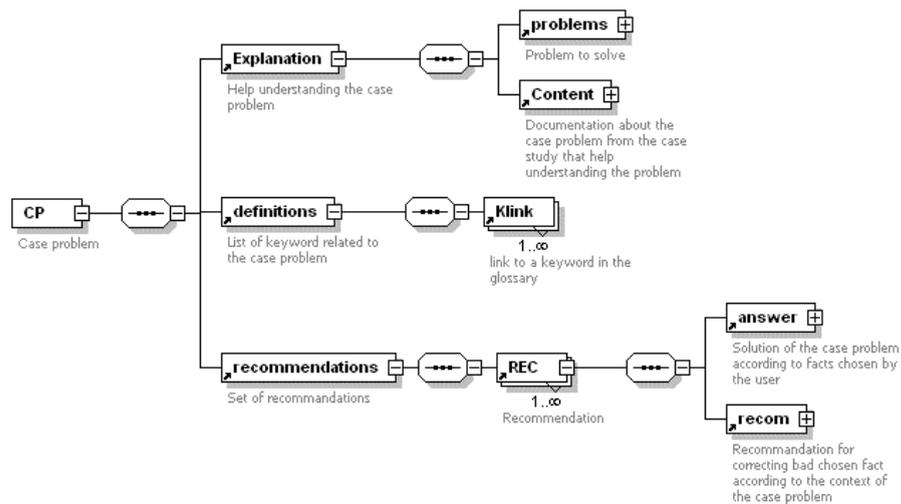


Figure 6-22 : Type de document Cas problème

Tableau 6-5 : Documentation du type de document Cas problème

Élément	Description	Type de données	Attributs	
CP	Délimite le document Cas problème		name	Nom du cas problème
			tc	URL vers le fichier du concept topologique à identifier
			cs	URL vers le fichier décrivant l'étude de cas d'où est extrait le cas problème
Explanation	Délimite l'explication du cas problème			
problems	Délimite le problème à résoudre	TextHTML		
Content	Délimite le contenu de l'explication	TextHTML		
definitions	Délimite une série de mots-clés			

Élément	Description	Type de données	Attributs	
Klink	Idem que dans le concept topologique			
recommendations	Délimite une série de recommandations			
REC	Délimite une recommandation	TextHTML	id	Numéro de la recommandation
			mincf	Borne inférieure de validité
			maxcf	Borne supérieure de validité
answer	Délimite la réponse	TextHTML		
recom	Délimite le texte de la recommandation	TextHTML		

6.8.5 Étude de cas

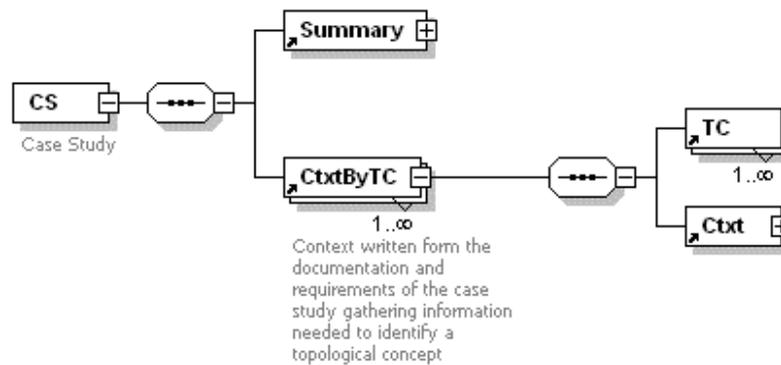


Figure 6-23 : Type de document Étude de cas

Tableau 6-6 : Documentation du type de document Étude de cas

Élément	Description	Type de données	Attributs	
CS	Délimite le document étude de cas		name	Nom du cas problème
			ref	URL vers un éventuel document explicatif du cas problème
Summary	Délimite un résumé du cas d'étude	TextHTML		
CtxtbyTC	Délimite les contextes			
TC	Délimite le contenu de l'explication	vide	TCref	URL vers le fichier du concept topologique pour lequel le contexte a été rédigé
Ctxt	Délimite un contexte	TextHTML		

6.8.6 Expert

Nous avons réparti les informations contenues dans la base de données ACCESS dans les différents types de fichiers. Cependant, certaines informations ne sont pas liées directement au type de fichiers mais plutôt à l'association entre eux, c'est pourquoi nous avons créé le type de fichier Expert. Ce dernier reprend les liens entre les fichiers et les informations qui sont associées à ces liens.

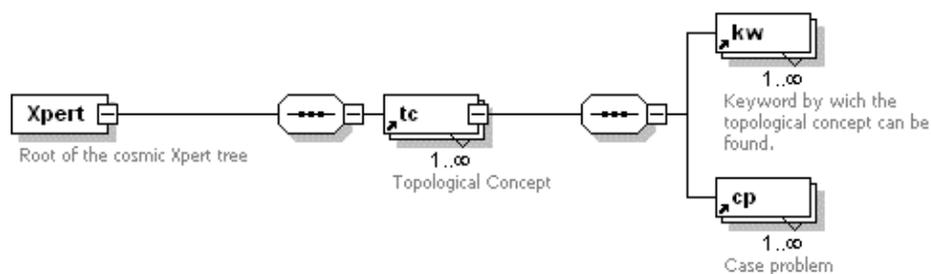
**Figure 6-24** : Type de document Expert

Tableau 6-7 : Documentation du type de document Expert

Élément	Description	Type de données	Attributs	Élément
Xpert	Délimite le document expert			
tc	Délimite un concept topologique		href	URL vers le fichier décrivant le concept topologique
kw	Représente un mot-clé par lequel on peut retrouver le concept topologique associé	vide	idRef	Identifiant du mot-clé dans le glossaire
			cf	Pourcentage associé au lien entre le mot-clé et le concept topologique
cp	Représente un cas problème associé au concept topologique	vide	href	URL vers le fichier décrivant le cas problème
			cf	Pourcentage associé au lien entre le cas problème et le concept topologique

6.9 Définir les vues

À chaque nouveau type de documents, plusieurs vues peuvent être définies afin de fournir les informations désirées à l'utilisateur.

Le langage XSL a servi à décrire les vues. Ainsi, des paramètres peuvent être associés aux vues. Voici une description des vues pour chacun des types de documents.

6.9.1 Glossaire

L'objectif du glossaire est de fournir la définition d'un mot-clé. Une seule vue est associée à ce type de document. Elle reçoit en paramètre l'identifiant du mot-clé à afficher.

Les informations affichées correspondent à la description d'un document Mot-clé dans la charte de structuration.

6.9.2 Concept topologique

Le type de document Concept topologique reprend deux vues ; celle décrite par les documents concept topologique et les thèmes de la charte.

La vue du document Concept topologique utilise le glossaire pour afficher les définitions relatives au concept topologique décrit.

La seconde vue, celle des thèmes reçoit comme paramètre l'identifiant du thème à expliquer ainsi que l'identifiant (le nom de fichier) du cas problème. Le fichier de type Cas problème fournit le contexte pour lequel l'utilisateur veut appliquer le thème.

6.9.3 Cas problème

Au type de document Cas problème sont associées deux vues. La première correspond au type de données Cas problème décrit dans la charte de structuration. La seconde correspond au type de données Recommandation.

Les deux vues prennent des paramètres. La vue du document Cas problème doit aller chercher le type de données Concept topologique afin de retrouver le contexte général décrivant l'étude de cas pour un concept topologique particulier. Les contextes se trouvent dans le type de données Étude de cas.

La vue des recommandations reçoit en paramètre le pourcentage calculé en fonction des faits choisis par l'utilisateur. Elle affiche la recommandation associée à ce pourcentage.

6.9.4 Étude de cas

Le document Étude de cas est utilisé dans plusieurs vues. La première donne le lien vers le fichier décrivant le cas ainsi qu'un bref résumé. La seconde vue qui utilise ce document est la vue des cas problèmes décrite ci-dessus.

6.9.5 Expert

Le document Expert, document central du système, est ce qui relie tous les documents entre eux. Une seule vue lui a été associée ; elle est réservée aux utilisateurs experts. Elle permet à ces derniers d'accéder à différents outils de vérification de la base de connaissances, comme par exemple un générateur de tests.

6.10 Traduire les documents

En définissant les types de données, nous avons créé une première base de connaissances dans le format XML. Nous avons pu utiliser cette première version de la base de connaissances pour en créer une nouvelle.

Le langage XSL a été utilisé pour décrire les vues. Il est aussi utilisé pour transformer le fichier XML source en un fichier HTML pouvant être visualisé par un navigateur Internet. Il est aussi possible d'utiliser le langage XSL pour traduire les fichiers XML de la première base de connaissances pour créer la base de connaissances finale du prototype 2.

La réalisation de la base de connaissances finale a pu être automatisée pour transférer la plupart des données de la première base de connaissances. Nous avons cependant ajouté manuellement les informations provenant de la base de données ACCESS.

6.11 Vérifications avec le prototype 2

Pour le prototype 2 notre principal but est de vérifier la base de connaissances tant dans son contenu que dans les recommandations. Un but secondaire, et ne faisant pas partie de notre recherche doctorale, est de faciliter l'accès au prototype pour les mesureurs, via le *World Wide Web*.

6.11.1 Déroulement de la vérification

La vérification consiste à déterminer si le produit a été construit correctement c'est-à-dire, dans ce cas-ci, s'il a suivi les normes de construction de la base de connaissances. Les normes de construction de la base de connaissances sont décrites de façon détaillée dans la présentation du prototype 2. Voici un résumé de ces normes :

- la base de connaissances est constituée de mots-clés, de concepts topologiques, de cas problèmes, de thèmes et de recommandations ;
- pour chacun de ces concepts, nous avons défini un format ou une norme de structuration du fichier ;
- pour chacun de ces concepts, le mesureur a accès à sa définition selon un format spécifique ;
- il y a six normes de structuration pour les fichiers dont cinq se retrouvent dans le modèle du prototype.

Note : dans le prototype 2, le "fait" étant seulement un oui ou un non, nous l'avons exclu du modèle. De plus, les études de cas ne font pas partie de l'ontologie COSMIC-FFP. Elles représentent les logiciels à mesurer.

Nous avons donc vérifié :

- si tous les fichiers respectaient les normes de structuration du schéma XML ;

- si pour chaque concept nous respectons la définition fournie dans l'ontologie COSMIC-FFP ;
- la syntaxe et la grammaire de tous les documents.

Avant la vérification, les fichiers (près de 1 000) en format RTF fournissaient toutes les références d'une centaine de cas problèmes. Pour chaque cas problème, nous produisons en moyenne une dizaine de références. Pour faciliter la vérification, nous avons les possibilités suivantes :

- automatiser le plus possible la vérification des fichiers ;
- réduire leur nombre ;
- faire les deux.

Nous avons décidé de faire les deux en utilisant une structure de fichiers de type XML. Il nous fallait donc transformer les fichiers RTF (dont la structure peut difficilement être vérifiée automatiquement) en format XML. Avec le format XML, nous pouvons non seulement définir la structure des documents de l'utilisateur (virtuel), mais aussi la contrôler par un programme (script) XSL. Le déroulement de la transformation est expliqué en détails à la sous-section 6.6.2.

La Figure 6-25 présente les liens entre les différents concepts et les réutilisations de ces concepts. Voici une explication des documents utilisateurs à partir des différents types de fichiers XML.

La structuration des fichiers XML de type "mot-clé" comprend uniquement une définition du mot-clé, définition qui se retrouve dans presque tous les autres documents utilisateurs.

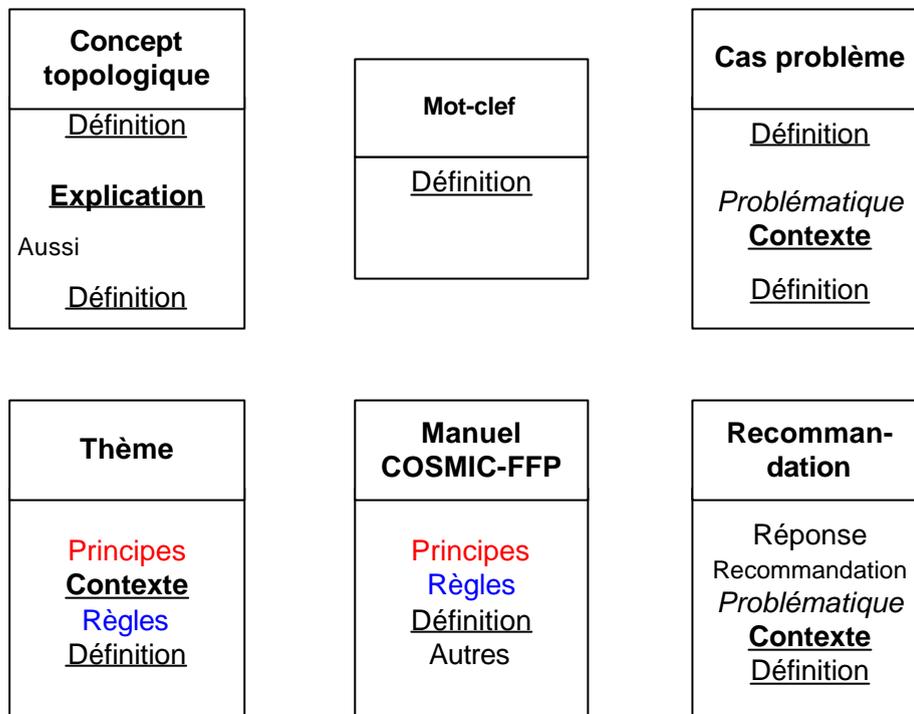


Figure 6-25 : Liens entre les différents types de documents utilisateurs

Les documents utilisateurs de type "concept topologique" comprennent une définition (tirée du Manuel COSMIC-FFP) et une explication de ce que veut dire le concept topologique utilisé (ex. : processus fonctionnel). Dans ce type de document, il y a aussi des définitions qui sont liées à la définition du concept utilisé (ex. : l'événement déclencheur est lié au processus fonctionnel).

Les documents utilisateurs de type "cas problème" comprennent les définitions, la problématique, le contenu du cas problème analysé, le contexte du cas problème provenant de l'étude de cas et aussi des définitions liées à ce cas problème en particulier. La problématique et le contexte se retrouvent aussi dans les documents utilisateurs de type "thème" et de type "recommandation". Le contexte se retrouve aussi dans les documents utilisateurs de type "thème". Chaque fois que le même cas problème est utilisé, la problématique et le contexte sont réutilisés.

Les documents utilisateurs de type "thème" comprennent des principes, un contexte (déjà vu dans les documents utilisateurs de type "cas problème"), des règles et des définitions. On peut constater que les principes et les règles sont tirés du Manuel COSMIC-FFP. À chaque principe est associé une réponse (oui/non) qui amène un calcul pour la recommandation.

Les documents utilisateurs de type "recommandation" ont une réponse générique (oui/non) résultant de la compilation des principes associés au cas problème analysé. Une recommandation est associée à la réponse. Le choix de la recommandation sera expliqué dans la sous-section vérification des recommandations. Dans les documents utilisateurs de type "recommandation" nous avons aussi à identifier la provenance des documents utilisateurs de type "cas problèmes", son contexte tiré des documents utilisateurs de type "thème" et enfin les définitions tirées des documents utilisateurs de type "mot-clé".

6.11.2 Résultats pour la vérification

Il y a plusieurs résultats pour la vérification correspondant à chaque type de vérification :

- le respect des normes de structuration. Nous avons des statistiques sur le nombre de fichiers/documents utilisateurs qui ne respectaient pas les normes de structuration avant la transformation. Les "documents utilisateurs" doivent nécessairement respecter la structuration puisque contrôlés par un "programme" de type XSL ;
- le respect des définitions de l'ontologie de COSMIC-FFP à chaque fois qu'une définition était utilisé dans un document utilisateur. Les nouveaux documents utilisateurs doivent nécessairement respecter les définitions de l'ontologie COSMIC-FFP puisqu'il n'y a qu'une seule référence ;
- la syntaxe et la grammaire de tous les documents utilisateurs.

Il y aussi un résultat lié au nouveau design de la base de connaissances :

- la réduction du nombre de fichiers dupliqués. Ce nombre est obtenu en comparant le nombre de fichiers en HTML (avant la transformation) au nombre de fichiers XML (après la transformation).

6.11.2.1 Respect des normes de structuration

Il y a cinq types de fichiers XML. Nous avons des statistiques pour chacun de ces fichiers :

- mot-clé : suite à cette vérification nous avons enlevé un mot-clé et avons ajouté deux mots-clés. Il y a eu sept modifications au contenu de la définition et sept modifications ou ajouts pour les références ;
- il n'y a pas eu de modifications des définitions et une seule modification à l'explication des concepts topologiques (ajout d'un exemple). Par contre, il y a eu cinq ajouts de référence ;
- les relations entre les concepts topologiques et les thèmes en termes de pourcentage ont été revus (12 % ont été modifiés) ;
- il y a neuf thèmes qui ont été modifiés au niveau du texte (modifications cosmétiques pour la plupart) ;
- il y a eu ajout d'une à quatre définitions pour une trentaine de thèmes. Ces ajouts auraient été assez difficiles à faire dans l'ancienne structure des fichiers (RTF) ;
- nous avons vérifié plus de 500 recommandations. Nous n'avons trouvé qu'une seule erreur dans l'explication des cas problèmes. Nous avons cependant ajouté de une à quatre définitions pour 288 recommandations.

La structuration des fichiers en format XML nous a permis de retracer beaucoup d'erreurs qu'il aurait été difficile de corriger en format RTF. Elle nous a surtout permis d'ajouter un bon nombre de définitions à l'intérieur des documents utilisateurs sans nous préoccuper de vérifier à chaque fois les définitions de l'ontologie COSMIC-FFP puisque ces définitions sont maintenant toutes dans un seul fichier.

6.11.2.2 Respect des définitions de l'ontologie

Nous avons vérifié toutes les définitions via les mots-clés :

- nous avons modifié le nom de deux définitions, enlevé une définition (non utilisée par aucun autre concept) et ajouté deux définitions ;
- nous avons modifié le contenu de six définitions (modifications cosmétiques) ;
- nous avons ajouté ou précisé les références pour huit définitions.

La structuration des fichiers en format XML nous a permis de retracer un mot-clé non utilisé par les autres concepts. Elle nous a aussi permis d'améliorer le contenu et les références d'un certain nombre de définitions sans nous préoccuper des conséquences dans les différents documents utilisateurs.

6.11.2.3 Syntaxe et grammaire de tous les documents utilisateurs

La syntaxe et la grammaire ont été réalisées avec le correcteur de WORD. Nous n'avons pas établi de statistiques à cet effet. Notre approche permet cependant de ne pas avoir à nous préoccuper des conséquences de ces corrections sur les différents documents utilisateurs mais seulement sur les fichiers. En effet, si une "définition" est utilisée dans une centaine de documents utilisateurs et qu'il y a une faute de syntaxe dans cette définition, il suffit de corriger la faute dans le fichier de type "glossaire" et la correction apparaît dans tous les documents utilisateurs possédant cette définition.

6.11.3 Déroulement de la vérification des recommandations

6.11.3.1 Définition et déroulement de la vérification des recommandations

Pourquoi ne pas avoir validé ? La démonstration de la validation est subjective, tel que mentionné en 3.1.11. La preuve de la validation vient du client (obtenir le bon produit) à travers sa propre subjectivité. Pour commencer, il nous faudra définir qui est le client de notre démarche et plus particulièrement dans le contexte du prototype 2. Sont-ce les experts ? Les

mesureurs sur le terrain ? Les apprentis mesureurs ? Les éducateurs ? Nous voulons, idéalement, fournir un "bon produit" (*the right product*) à toutes ces personnes.

Dans cette perspective, nous devrions rejoindre toutes ces catégories de clients et leur demander si l'on a construit le bon produit. Les expérimentations avec le prototype 1 auprès du mesureur, indiquent par exemple que, pour le mesureur débutant, le prototype 1 lui a permis d'obtenir de meilleures réponses (du moins pour un groupe) et de fournir un nombre de mauvaises réponses plus faible.

D'autres questions se posent :

- est-ce que les experts pourront arriver à mieux définir et plus rapidement les solutions à différents problèmes ?
- est-ce que les mesureurs habitués à la mesure pourront obtenir de meilleurs résultats en utilisant le prototype 2 ?
- est-ce que l'apprentissage de la norme COSMIC-FFP sera plus rapide en utilisant le prototype 2 ?
- est-ce que l'utilisation du prototype 2 peut permettre aux développeurs de construire un outil d'automatisation de la méthode de mesure COSMIC-FFP et de mieux tenir compte des contraintes des normes ?

Toutes ces questions, et d'autres, exigent que soient conduites plusieurs expérimentations auprès de ces différents types d'utilisateurs afin d'obtenir des résultats significatifs. Ce type d'expérimentation dépasse largement le contexte de notre thèse en raison de l'ampleur des travaux.

Que veut alors dire "vérification des recommandations" du prototype 2 dans le contexte de notre recherche ?

Première définition : lorsque le mesureur a complété les réponses (les faits) pour un cas problème, il obtient une recommandation. Si la recommandation respecte les règles de la méthode de mesure COSMIC-FFP il n'y a pas d'erreurs. Une autre façon de vérifier leur

exactitude, est de comparer les réponses des 105 cas problèmes de la base de connaissances avec les cas utilisés pour les préparer.

Deuxième définition : est-ce que le contenu des recommandations est bien formulé, comprend des erreurs de syntaxes, etc. ?

Étendue et limites de la vérification des recommandations

Idéalement, le système à base de connaissances devrait fournir une recommandation différente pour toutes les possibilités de réponses (les faits) aux thèmes et ce, pour chaque cas problème. Avec 105 cas problèmes, nous avons calculé qu'il pouvait y avoir plus de 1 480 recommandations. Notre vérification a porté sur 545 recommandations. Pourquoi cette différence ?

Pour la plupart des cas problèmes, un seul fait négatif (ou réponse négative) pour un thème conduit à une recommandation négative. Par exemple, si le nombre de thèmes est de quatre, et que l'on répond négativement à un seul de ces thèmes, la recommandation sera négative. Si deux, trois ou quatre faits sont négatifs, la recommandation sera toujours négative. Il y aurait alors une recommandation positive pour 105 cas problèmes et une recommandation négative pour 105 cas problèmes. Pourquoi avons-nous vérifié 545 recommandations et non 210 (2 fois 105) ?

À nos recommandations, lorsqu'il y avait un seul fait négatif, nous avons fourni une recommandation spécifique en rapport avec ce fait négatif et ce, pour les 105 cas problèmes. Il y a aussi quelques cas où deux réponses négatives pouvaient conduire à une recommandation spécifique. C'est ce qui explique qu'au total nous avons vérifié 545 recommandations.

6.11.4 Résultats de la vérification avec le prototype 2

Les résultats de vérification nous ont permis de trouver des erreurs de contenu (deuxième définition) mais pas d'erreur dans les recommandations (première définition). Nous avons

trouvé 191 erreurs sur les 545 existantes dans le prototype 2 (soit 35 % d'erreurs), la plupart de nature grammaticale (i.e. reformulation de la recommandation ou déplacement de texte). Toutes ces erreurs ont été corrigées rapidement dans les différents textes.

Le fait de ne pas trouver d'erreur dans les recommandations n'est pas surprenant puisque les cas utilisés pour créer les 105 cas problèmes donnaient déjà les réponses.

7. SOMMAIRE DES RÉSULTATS

7.1 Problématique, objectifs, méthodologie et résultats

Dans notre problématique, nous avons établi qu'il n'existe pas de méthodes systématiques et reconnues pour réaliser la mise en correspondance des règles formelles avec les différents environnements de développement (monde empirique). De plus, les connaissances d'experts en mesure, connaissances acquises après plusieurs années de pratique, reflètent deux ensembles d'habiletés ou de connaissances nécessaires au mesureur :

- la compréhension des artefacts du logiciel pour situer le problème (ou cas problème) à résoudre (comprendre et interpréter) ;
- la compréhension des règles de la méthode de mesure pour les appliquer aux cas problèmes à résoudre (utiliser et solutionner).

Nos objectifs étaient : 1) d'encapsuler et de structurer la démarche des experts, en plus 2) de fournir aux mesureurs des règles et des recommandations qui permettent, en tenant compte de l'expérience du mesureur et de la qualité des informations obtenues, d'améliorer les résultats des mesures.

Nous avons développé une méthodologie afin d'atteindre nos objectifs de recherche. Cette méthodologie comprenait plusieurs étapes illustrées à la Figure 3-1. La première partie de la méthodologie (partie gauche de la Figure 3-1) consistait à trouver une solution théorique à notre objectif d'encapsuler et de structurer la démarche des experts en mesure fonctionnelle (objectif 1). La deuxième partie de la méthodologie (partie droite de la Figure 3-1) consistait à fournir aux mesureurs des règles et des recommandations qui leur permettront d'améliorer les résultats des mesures, ce qui se traduit par la création de deux prototypes (objectif 2).

Les résultats des expériences avec le premier prototype auprès des experts de la mesure fonctionnelle pour la méthode de mesure COSMIC-FFP ont montré que ces derniers ont

favorablement accueilli les résultats produits par le premier prototype. Ce prototype a été construit à partir du design du cheminement du mesureur et d'une approche cognitive de type diagnostic. Les résultats des expériences avec le premier prototype, auprès de deux groupes de mesureurs débutants, ont montré que les utilisateurs du prototype 1 ont mieux réussi à identifier les concepts de la méthode de mesure COSMIC-FFP que ceux qui n'avaient pas accès au prototype 1.

Nous avons aussi vérifié les recommandations de la base de connaissances du système à base de connaissances{ XE "base de connaissances" }. Cet exercice nous a permis de constater :

- qu'il y avait des possibilités d'utiliser ce système à base de connaissances avec un plus grand nombre de cas problèmes (d'une trentaine de cas à une centaine) sans pour autant sacrifier sur la qualité du contenu des références de la base de connaissances ;
- qu'il y avait des possibilités de rendre disponible la base de connaissances à un plus large public de mesureurs.

Ces nouvelles possibilités seront à la base de nouvelles recherches.

7.2 Réalisations

Voici la liste des réalisations dans le cadre de notre recherche :

- nous avons décrit le cheminement du mesureur en quatre étapes : comprendre, interpréter, utiliser et solutionner (Chapitre 4). Les résultats de cette recherche ont été publiés dans un article intitulé "Applying a Functional Measurement Method: Cognitive Issues" [11] ;
- nous avons fait un arrimage du cheminement du mesureur avec deux autres modèles de mesure : la méthode de mesure COSMIC-FFP [2] et les activités de la phase 2 des phases du processus de Jacquet et Abran [29]. Une partie des résultats de cette recherche ont été publiés dans l'article déjà cité [11] et repris plus en profondeur dans cette thèse (Chapitre 4) ;

- nous avons modélisé l'approche de façon plus détaillée (chapitre 4). Les résultats de cette recherche ont été publiés dans un article intitulé "Knowledge Modeling for the Design of a KBS in the Functional Size Measurement Domain" [76] ;
- nous avons présenté les étapes et les sous-étapes de la démarche du mesureur en fournissant quelques exemples de cas de diagnostic (Chapitre 4). Les résultats de cette recherche ont été publiés dans un article intitulé "A Knowledge-Based System in Functional Size Measurement" [77] ;
- nous avons choisi une méthode de calcul pour le traitement de la connaissance. Cette méthode de calcul est basée sur la théorie de la certitude et elle est décrite dans cette thèse (Chapitre 4) ;
- nous avons créé un premier prototype (COSMICXpert) pour vérifier la justesse de notre démarche (Annexe B). Les résultats de cette recherche ont été publiés dans un article intitulé *Design of a Diagnostic Tool to Improve the Quality of Functional Measurement* [78].
- Nous avons créé un deuxième prototype (COSMICXpert) pour permettre la vérification de la base de connaissances, tout en permettant un accès plus facile au système de base de connaissances COSMICXpert.

Dans un deuxième temps, nous avons procédé à des expérimentations pour vérifier les résultats des étapes précédentes. Voici la démarche suivie pour les expérimentations :

- nous avons défini et réalisé une vérification d'une base de connaissances (35 cas problèmes avec près de 200 références) auprès de quelques experts de la mesure fonctionnelle. Les résultats de cette vérification sont dans cette thèse (Chapitre 5) ;
- nous avons défini et réalisé une expérimentation auprès de deux groupes de mesureurs débutants afin de vérifier l'hypothèse suivante : les mesureurs débutants ayant accès à un prototype de diagnostic peuvent obtenir de meilleurs résultats d'application de la méthode de mesure COSMIC-FFP (pour mesurer un logiciel) que ceux qui n'ont pas accès au prototype mais seulement au manuel de mesure. Les résultats de cette expérimentation sont dans cette thèse (Chapitre 5) ;

- nous avons inscrit plus d'une centaine de cas problèmes dans la base de connaissances, créé près d'un millier de références et un deuxième prototype (COSMICXpert) (Annexe C). Les recommandations des cas problèmes et les références ont été vérifiées. Les résultats de cette vérification sont dans cette thèse (Chapitre 6).

En résumé, les principaux résultats de notre recherche :

- le cheminement du mesureur ;
- le prototype 1 de diagnostic ;
- la modélisation d'un système à vase de connaissances faisant le pont entre le domaine de la mesure et l'informatique cognitive ;
- les résultats de la vérification auprès des experts ;
- les résultats des expérimentations auprès des experts ;
- les résultats de l'expérimentation auprès de mesureurs novices ;
- le prototype 2 de diagnostic ;
- la disponibilité du prototype 2 sur le WEB. Adresse URL :
<http://142.137.16.116/COSMICXpert/login.jsp> ;
- les résultats de la vérification de la base de connaissances et les corrections apportées à la base de connaissances ;
- les résultats de la vérification de la base de connaissances et les corrections apportées à la base de connaissances ;
- la rédaction du rapport de thèse.

7.3 Recherches à venir

Les recherches à venir se situent surtout au niveau des expérimentations avec le deuxième prototype. Ce prototype étant plus facilement disponible via le WEB, il nous sera possible :

- de vérifier le prototype 2 avec un plus grand nombre d'experts de la méthode de mesure COSMIC-FFP ;
- d'expérimenter le prototype 2 avec des mesureurs débutants dans l'industrie ;
- de déterminer quels sont les principaux concepts utilisés par les mesureurs débutants à partir du fichier "suivi du mesureur" de prototype 2 ;
- d'ajouter dans la base de connaissances les "règles locales" d'une organisation et d'expérimenter le prototype avec des mesureurs de cette organisation ;
- de définir des concepts liés à la notion de qualité de la documentation et des expérimentations en vue d'établir la qualité des mesures en rapport avec la qualité de la documentation ;
- d'explorer plus à fonds le domaine cognitif afin de voir si d'autres approches que les techniques de représentation à base de cas et de règles peuvent être utilisées ;
- de créer une interface "intelligente" pour le prototype 2 afin de rendre plus facile l'entrée de nouveaux cas dans la base de connaissance, tout en vérifiant les erreurs potentielles ;
- de créer une ontologie du cycle du logiciel.

BIBLIOGRAPHIE

- [1] KOLODNER, J. L., *Case-Based Reasoning*. San Francisco: Morgan Kaufmann, 1993, 668 p.
- [2] ABRAN, A., J.-M. DESHARNAIS, S. OLIGNY, D. ST-PIERRE, et C. SYMONS, "Measurement Manual 2.2," UQÀM May 2002.
- [3] ISO, "ISO/IEC 19761:2002 Software Engineering - COSMIC-FFP Functional size measurement method," 2002.
- [4] ISO, "ISO/IEC 14143-3: 1997 - Information Technology - Software Measurement - Functional Size Measurement - Definition of Concepts." Sydney, December, 1996, pp. 11.
- [5] RÉDACTION, "Les dépenses technologiques devraient croître de 3,4% en 2002," vol. 2003, 2002.
- [6] ALBRECHT, A. J., "Measuring Application Development Productivity," document présenté à IBM Applications Development Symposium, Monterey, CA, 1979, pp. 83-92.
- [7] SYMONS, C. R., "Mark 2 Function Points for Productivity Measurement & Estimating," in *International Conference on Project Management, Planning & Estimation*. London: Nolan, Norton & Co., 1990, pp. 15.
- [8] MORRIS, P. et J.-M. DESHARNAIS, "Measuring ALL the Software not just what the Business Uses," document présenté à International Function Point Users Group, 1998, 18 p.
- [9] ISO, "ISO/IEC 14143-1: 1997 - Information Technology - Software Measurement - Functional Size Measurement - Definition of Concepts." Sydney, December, 1996, pp. 11.
- [10] ABRAN, A. et C. SYMONS, "COSMIC, The 2nd generation of Functional size measurement methods," document présenté à ESCOM, London, 2001, 12 p.
- [11] DESHARNAIS, J.-M. et A. ABRAN, "Applying a Functional Measurement Method: Cognitive Issues," document présenté à International Workshop on Software Measurement, Montreal, August, 2001, 26-50 p.
- [12] BIELAWSKI, L. et R. LEWAND, *Expert Systems Development: Building PC-based applications*: Wellesley, 1988, 302 p.

- [13] MÜLLERBERG, M., A. ABRAN, J.-M. DESHARNAIS, D. MEYERHOFF, et D. ST-PIERRE, "The Function Point Assistant: a Joint German-Quebec Initiative," in *Metrics in Software Evolution, October*. Bonn: GMD, 1995, pp. 247-273.
- [14] ISO, *Vocabulaire International des termes fondamentaux et généraux de métrologie*. Genève: ISO, 1993, 59 p.
- [15] FENTON, N. E. et S. L. PFLEEGER, *Software Metrics - A rigorous & Practical Approach*, Second ed. London: International Thomson Computer Press, 1997, 638 p.
- [16] FENTON, N. E., *Software Metrics: a Rigorous Approach*: Chamman & Hall, 1991, 337 p.
- [17] ZUSE, H., *A Framework of Software Measurement*. Berlin: Walter de Gruyter, 1998, 755 p.
- [18] HALSTEAD, M. H., *Elements of Software Science*. New York, NY: Elsevier Computer Science Library, 1977, 127 p.
- [19] BOEHM, B. W., *Software engineering economics*. Englewood Cliff, New Jersey: Prentice-Hall Inc., 1981, 767 p.
- [20] CONTE, S. D., H. E. DUNSMORE, et V. Y. SHEN, *Software engineering metrics and models*. Menlo Park: The Benjamin/Cummings Publishing Company, Inc., 1986, 396 p.
- [21] ABRAN, A., "Analyse du processus de mesure des points de fonction," in *École polytechnique*. Montreal: Université de Montréal, 1994, pp. 324.
- [22] DEMARCO, T., *Controlling software projects*. Englewood Cliffs, New Jersey: Yourdon Press, a Prentice Hall company, 1982, 284 p.
- [23] ALBRECHT, A. J., "Productivity Measurement and Estimate Validation," IBM CIS&A Guideline 313 AD/M, 1984.
- [24] IFPUG, "Function Point Counting Practices Manual, Release 4.1," International Function Point Users Group (IFPUG), Mequon, Wisconsin Release 4.1, January 1999.
- [25] DESHARNAIS, J.-M. et P. MORRIS, "Comparison between FPA and FFP: a field experience," document présenté à in 8th International Workshop on Software Measurement, Magdeburg, Germany, 1998, 17 p.
- [26] SYMONS, C. R., "Function Point Analysis: Difficulties and Improvements," *IEEE Transactions on Software Engineering [ISO]*, vol. 14 (1), pp. 2-11, 1988.

- [27] DESHARNAIS, J.-M., "Analyse statistiques de la productivité des projets de développement en informatique à partir de la technique des points de fonction," in *Informatique*. Montréal: Université du Québec à Montréal, 1988, pp. 56.
- [28] ABRAN, A., J. W. MOORE, P. BOURQUE, R. DUPUIS, et L. TRIPP, "Guide for the Software Engineering Body of Knowledge." Montréal: A project of the IEEE Computer Society, 2001, pp. 213.
- [29] JACQUET, J.-P. et A. ABRAN, " Metrics Validation Proposals: A Structured Analysis, 8th International Workshop on Software Measurement, Magdeburg, Germany, 1998.," document présenté, 1998, 15 p.
- [30] MORRIS, P. M. et J.-M. DESHARNAIS, "Validation of Function Points - An Experimental Perspective," document présenté à IFPUG - Complete the Rings of Measurement... Go for the Gold Using Function Points, Atlanta, Georgia, 1996, 167-184 p.
- [31] NISHIYAMA, S. et T. FURUYAMA, "The validity and applicability of function point analysis - as related to specification quality and ergonomics -," document présenté à EOQ-SC'94, Basel, Switzerland, 1994, 479-490 p.
- [32] OLF, "Le grand dictionnaire terminologique," 2003.
- [33] JACKSON, P., *Introduction to Expert Systems*, Third ed. New York: Addison-Wesley, 1998, 542 p.
- [34] LEE, Y. J. et C. Y. SHIH, "The Riesz representation theorem on infinite dimensional spaces and its applications," *Infinite Dimensional Analysis Quantum Probability and Related Topics*, vol. 5 (1), pp. 41-59, 2002.
- [35] DESHARNAIS, J.-M., "Validation Process for Industry Benchmarking Data," document présenté à IEEE Conference on Software Maintenance, Montreal, 1993, 6 p.
- [36] RUDOLPH, E. E., "Precision of Function Point Counts," document présenté à IFPUG Spring Conference, San Diego, California, 1989, 14 p.
- [37] LOW, G. C. et D. R. JEFFERY, "Function points in the estimation and evaluation of the software process," *IEEE Transactions on Software Engineering*, vol. 16 (1), pp. 64-71, 1990.
- [38] KEMERER, C. F., "Function Point Measurement Reliability: a Field Experiment," MIT Sloan School of Management, 1990, pp. 16.
- [39] NISHIYAMA, S., "On Precision on Function Point Analysis," Research and Development Center, Nippon Telegraph and Telephone, Japan 1999.

- [40] NDAGIJIMANA, S. M., "Étude du processus de mesures des points de fonction et analyse de la documentation fonctionnelle," in *Institut d'informatique*. Namur: Notre-Dame de la Paix, 2002, pp. 80.
- [41] PRINCE, V., *Vers une informatique cognitive dans les entreprises: Le rôle central du langage*. Paris: Masson, 1996, 190 p.
- [42] SCHREIBER, A. T., J. M. AKKERMANS, A. A. ANJEWIERDEN, R. DE HOOG, N. R. SHADBOLT, W. VAN DE VELDE, et B. J. WIELINGA, *Knowledge Engineering and Management: The CommonKADS Methodology*. Boston: MIT Press, 2000, 510 p.
- [43] KARKAN, J.-M. et G. TJOEN, *Systèmes experts: un nouvel outil pour l'aide à la décision*. Bruxelles: Masson, 1993, 153 p.
- [44] WATSON, I., *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco, California: Morgan Kaufmann Publishers, Inc., 1997, 289 p.
- [45] ALLAHVERDI, N. et S. YALDÝZ, "EXPERT SYSTEM APPLICATIONS IN MEDICINE AND EXAMPLE OF DESIGN OF A PRE-DIAGNOSIS EXPERT SYSTEM," vol. 2003. Konya, Turkey: Selçuk University.
- [46] PAQUETTE, G., "L'intelligence artificielle, comprendre et prolonger l'intelligence humaine," in *Télé-université*. Montréal: Textes écrits et recueillis par Gilbert Paquette avec la collaboration de Anne Bergeron, 1989, pp. 450.
- [47] LAURIÈRE, J.-L., *Intelligence artificielle, résolution de problèmes par l'homme et la machine*. Paris: Eyrolles, 1987, 502 p.
- [48] BUDGEN, D., *Software Design*: Addison-Wesley, 1994, 370 p.
- [49] POLYA, G., *How to solve it*. Princetown: University Press, 1957, 224 p.
- [50] DURKIN, J., *Expert system: Design and Development*. New York: Prentice Hall, 1994, 800 p.
- [51] LUKASIEWICZ, J., "Curriculum Vitae of Jan Lukasiewicz," *Metalogicon* 7, vol. 2, pp. 133-137, 1994.
- [52] MARLING, C., M. SQALLI, E. RISSLAND, H. MUNOZ-AVILA, et D. AHA, "Case-Based Reasoning Integrations," *AI Magazine*, vol. 23 (1), pp. 69-85, 2002.
- [53] RISSLAND, E. L. et D. B. SKALAK, "CABARET: Rule Interpretation in a Hybrid Architecture," *International Journal of Man-Machine Studies*, vol. 34, pp. 839-887, 1991.

- [54] GOLDING, A. et P. ROSENBLOOM, "Improving Accuracy by Combining Rule - Based and Case-Based Reasoning," *Artificial Intelligence*, vol. 87 (1-2), pp. 215-254, 1996.
- [55] AHA, D. W. et J. J. DANIELS, "CBR Integrations," document présenté à Workshop (TR WS-98-15), Menlo Park, 1998, 177 p.
- [56] PRENTZAS, J. et I. HATZILYEROU, "Integrating Hybrid Rule-Based with Case-Based Reasoning," document présenté à LNAI 2416 ECCBR 2002, Aberdeen, 2002, 336-349 p.
- [57] KOLODNER, J. L., *Case-Based Learning*. Boston/Dordrecht/London: Kluwer Academic Publishers, 1993, 363 p.
- [58] STUDER, R., R. BENJAMINS, et D. FENSEL, "Knowledge Engineering: Principles and Methods," in *Data & Knowledge Engineering*, vol. 25, 1998, pp. 161-197.
- [59] ERIKSSON, H., R. W. FERGERSON, Y. SHAHAR, et M. A. MUSEN, "Automatic Generation of Ontology Editors," document présenté à Twelfth Banff Knowledge Acquisition for Knowledge-based systems Workshop, Banff, Alberta, Canada., 1999, 20 p.
- [60] VAN HEIJST, G., A. T. SCHREIBER, et A. WIELINGA, "Using Explicit Ontologies in KBS Development," vol. 2003. Amsterdam: University of Amsterdam, Department of Social Science Informatics, 1997.
- [61] ROY, R., *Industrial Knowledge Management: a micro-level approach*. Bedford (UK), 2001, 565 p.
- [62] MIZOGOUCI, R. et Y. KITAMURA, "Foundation of Knowledge Systematisation: Role of Ontological Engineering," in *Industrial Knowledge Management: a micro-level approach*, C. University, Ed. Bedford (UK), 2001, pp. 17-36.
- [63] USCHOLD, M. et R. JASPER, "Ontologies for Knowledge Management," in *Industrial Knowledge Management: a micro-level approach*, Bedford (UK), 2001, pp. 549-563.
- [64] BUCHANAN, B., D. BARSTOW, R. BECHTEL, J. BENNETT, W. CLANCEY, C. KULIKOWSKI, T. MITCHELL, et D. A. WATERMAN, "Constructing an expert system," in *Building Expert Systems*, D. A. W. F. Hayes-Roth, and D.B. Lenat, Ed. Reading: Addison-Wesley, 1983, pp. 127-167.
- [65] WIELINGA, B., G. SCHREIBER, et J. BREUKER, *KADS: A Principled Approach to Knowledge-Based System Development*. New York: Academic Press, 1993, 305 p.

- [66] HAYES-ROTH, F., "Artificial Intelligence, Whats Works and What Doesn't?," *AI Magazine* (Summer 1997), pp. 99-113, 1997.
- [67] PREECE, A., "Evaluating Verification and Validation Methods in Knowledge Engineering," in *Micro-Level Knowledge Management*, I. R. R. (ed), Ed., 2001, pp. 123-145.
- [68] BOEHM, B., "Verifying and validating software requirements and design specifications," *IEEE Software*, vol. 1 (1), 1984.
- [69] COENEN, F., "Verification and validation issues in expert and database systems: the expert systems perspective," document présenté à Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on, 1998, 16-21 p.
- [70] WILLIE, C., A. ABRAN, J.-M. DESHARNAIS, et R. R. DUMKE, "The quality concepts and sub concepts in SWEBOK: An ontology challenge," document présenté à 13th International Workshop on Software Measurement, Montreal, 2003, 113- 130 p.
- [71] SPECIFICATIONS, U. M. L., vol. 2003, 1999.
- [72] DEHN, N. et R. SCHANK, *Artificial and Human Intelligence*: Robert J. Stenberg, 1982.
- [73] RUSSO, A., "Modal labelled deductive systems," in *Technology and Medecine*. London: Imperial College of Science, Technology and Medecine, Department of Computing, 1995, pp. 266.
- [74] ZELKOWITZ, M. V. et R. W. DOLORES, "Experimental Models for Validating Technology," *IEEE Software* (May), pp. 23-31, 1998.
- [75] "XMLSpy Enterprise Edition," 5e ed: Altova, 2002.
- [76] DESHARNAIS, J.-M., A. ABRAN, A. MAYERS, L. BUGLIONE, et V. BEVO, "Knowledge Modeling for the Design of a KBS in the Functional Size Measurement Domain," document présenté à KES 2002, Crema, Italy, 2002, 7 p.
- [77] DESHARNAIS, J.-M., A. ABRAN, A. MAYERS, L. BUGLIONE, et V. BEVO, "A Knowledge-Based System in Functional Size Measurement," UQÀM, Montréal, Departemental document 2002.
- [78] DESHARNAIS, J.-M., A. ABRAN, A. MAYERS, et T. KÜSSING, "Design of a diagnostic tool to improve the quality of functional measurement," document présenté à Proceedings of the 12th International Workshop on Software Measurement, Magdeburg, Germany, 2002, 52-60 p.

Présentation des annexes

Annexe A : Description de Help!CPR

Help!CPR est une coquille de système de connaissance de type diagnostic. Nous avons utilisé Help!CPR pour vérifier nos hypothèses concernant l'utilisation d'un système de connaissance pour fins de diagnostic pour aider le mesureur. Nous avons emprunté, pour la construction de notre premier prototype, plusieurs idées contenues dans ce système de connaissance. Dans cette annexe nos décrivons brièvement en quoi consiste ce système de connaissance de type diagnostic.

Annexe B : Description du prototype 1

Cette annexe décrit les exigences que nous avons fournies à un étudiant stagiaire à la maîtrise en informatique (Tim Küssing de l'université de Magdeburg en Allemagne) pour réaliser le premier prototype (diagramme de cas, scénarios des agents, exigences détaillées). Il fournit aussi le Guide de l'utilisateur du prototype 1.

Annexe C : Description de l'environnement de développement du prototype 2

Cette annexe décrit l'environnement de développement du prototype 2. Le logiciel a été développé par deux stagiaires dans le cadre de leur maîtrise en informatique (Julien Vilz et François Gruselin de l'université Notre Dame de la Paix en Belgique et supervisé par le Dr Habra). Nous ne présentons pas les exigences car elles sont les même que ceux du premier prototype. En ce qui concerne la base de connaissance, sa structure est déjà décrite dans le texte principal.

Annexe D : Procédure de vérification de la base de connaissance du prototype 1

Cette annexe est la procédure de vérification de la base de connaissance du prototype 1 qui a été fournie aux experts.

Annexe E : Curriculum des experts pour le prototype 1

Cette annexe comprend sur une page le curriculum des experts qui ont contribué à la vérification de la base de connaissance.

Annexe F : Liste des associations internationales: mesure fonctionnelle

L'annexe F fournit la liste des associations internationales pour la mesure fonctionnelle. Cette liste est probablement incomplète, mais elle montre l'intérêt que porte la communauté informatique internationale à la mesure fonctionnelle.

Annexe G : Exemples de règles "locales"

Dans notre recherche il est souvent questions de règles locales ou règles industrielles. L'annexe G fournit un exemple de ces règles locales. Nous remercions ici particulièrement M. Denis St-Pierre du CIM (Centre d'intérêts sur les métriques) pour avoir fourni gracieusement ces règles à l'automne 2001. Nous n'avons pas modifié la liste, ni le vocabulaire utilisé même si parfois non conforme avec le vocabulaire de COSMIC-FFP version 2.2 et notre thèse. Nous avons laissé les règles locales en anglais puisque c'est la langue d'échange entre les experts de la mesure fonctionnelle.

Annexe H : Études de cas

Nous avons utilisé 4 études de cas pour réaliser la base de connaissance du prototype 2. Pour trois de ces études de cas nous fournissons les références URL soit Data Warehouse, Rice Cooker et Valve Control. Pour la dernière (Report Generator), nous fournissons le texte de l'étude de cas.

Annexe I : Formules du prototype 1

Dans le prototype 1, codé en Visual Basic de Microsoft (version 6.0), on retrouve un ensemble de formules qui sont résumées en pseudo code.

Annexe J : Analyses statistiques

Résultats des analyses de variance et de corrélation (SPSS version 8.0)

Annexe K : Résultats des bonnes réponses

Le résultat détaillé des bonnes réponses de chaque mesureur débutant.

Annexe L : Exemples de mauvaises réponses

Exemples de mauvaises réponses de quelques mesureurs débutants (5)

Annexe M : Spécifications présentées aux mesureurs débutants

Cette annexe présente un exemple de documentation fournie aux mesureurs débutants. Il s'agit des deux premières étapes de la première spécification. Cette documentation contient des termes anglais que nous avons conservés tels quels. Nous avons aussi noté de nombreuses fautes de français que nous n'avons pas corrigées.

Annexe A: description de Help!CPR

A.1 Description fonctionnelle de Help!CPR (utilisateur)

Help!CPR a un menu permettant d'ouvrir et de sauvegarder plusieurs bases de données en format « Microsoft Access ». Help!CPR organise l'information sur les cas en trois objets distincts: les problèmes, les questions et les actions. Un quatrième objet pourrait être le « query » ou les mots clefs. Les cas sont créés en interconnectant ces objets ensembles. Il n'est pas nécessaire d'avoir l'objet « action » pour la résolution de cas, par contre les autres « objets » sont essentiels.

Il est possible d'assigner (via l'interface de l'expert¹) une information au niveau des « problèmes » (objet « problems ») ou des actions (objet « actions »), mais pas au niveau des questions (objet « questions »). Cette assignation est l'équivalent d'une référence hypertexte.

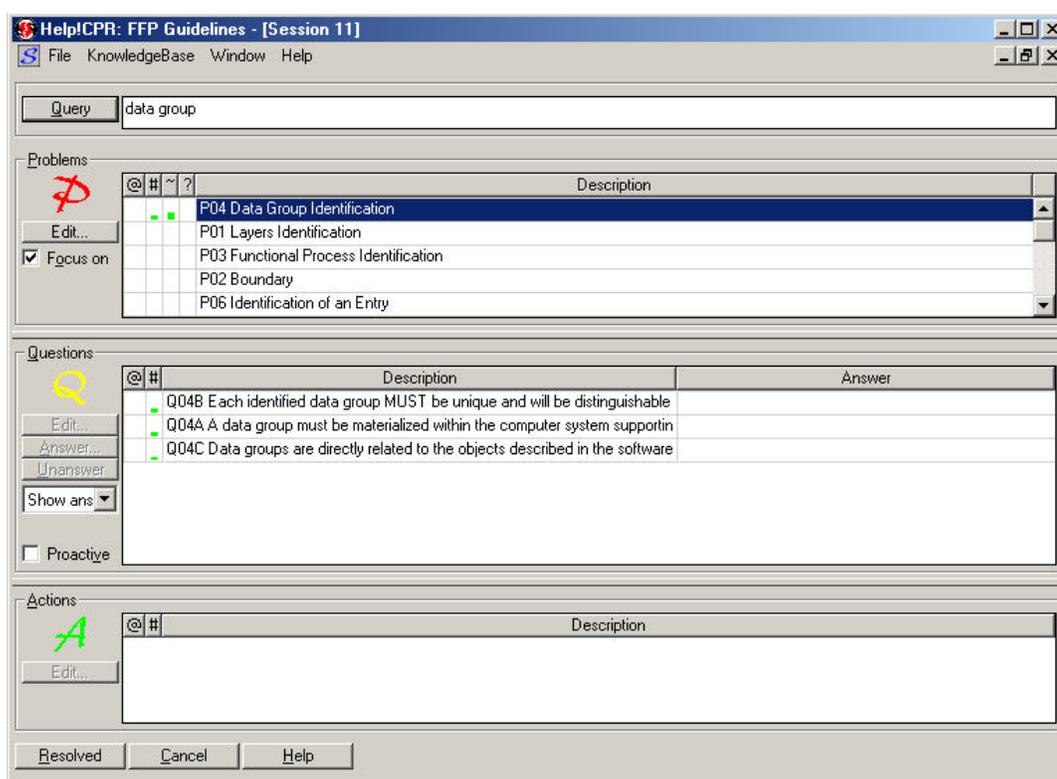


Table 1 Interface utilisateur de Help!CPR

Il est possible de choisir un problème particulier en utilisant la fonction ou l'objet « Query » (mot clef). Help!CPR met en relation le problème identifié et les cas se trouvant dans la base de cas. Il suggère alors

¹ Nous ne présentons pas les interfaces de l'expert de Help!CPR, ce qui demanderait une trop longue explication non nécessaire pour les fins de ce document.

un ou des problèmes à solutionner. Par exemple, en entrant le terme « data group » dans l'objet « Query » on retrouve dans l'objet « problems », le problème relatif au « data group ».

Automatiquement, dans l'objet « questions », les questions relatives à ce problème apparaissent. Le novice doit alors entrer une réponse. Ces réponses donnent une solution possible au problème. La couleur verte à gauche indique une réponse positive, alors que la couleur rouge indique une réponse négative. Il est aussi possible de suggérer une action selon la nature des réponses aux questions. Ici l'action pourrait être d'aller vers d'autres « mots clefs » pour compléter la connaissance. À noter aussi le numéro de session. Chaque session est numérotée et suivie.

A.2 Description fonctionnelle de Help!CPR (expert)

Ce logiciel a inspiré notre démarche cognitive au départ.

A 2.1 Les problèmes

La Figure 15 montre l'interface de l'expert qui permet de poser des questions et aussi l'interface des questions. Le problème peut apparaître sous forme d'énoncés ou de questions. À chaque problème est associé un ensemble de questions (voir questions plus bas). Il est aussi possible d'introduire des commentaires à chaque problème en éditant un problème particulier (voir EDIT).

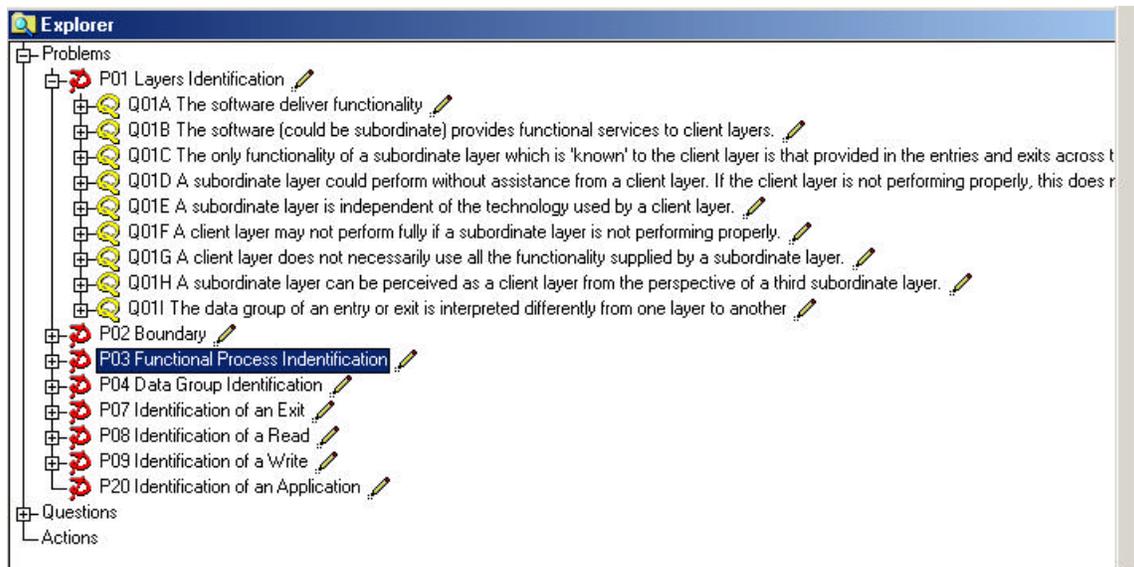


Table 2 Interface expert

La Figure 16 montre les questions liées au problème d'identification du processus unique.

Il est aussi possible d'entrer des commentaires.

A.2.2 Les questions

La Figure 17 montre la section « questions » d'une session. À une réponse à une question est associé un pourcentage. Ici, à la question : «Do we have batch and on line function? », la réponse «Yes » est associée à un pourcentage d'influence de la réponse à la résolution du problème selon l'expert. Ces influences sont ensuite additionnées. Il est aussi possible d'introduire des commentaires à chaque question en éditant une question particulière (voir EDIT).

Questions		@ #	Description	Answer
			Does the transaction maintain a group of data?	
			Is it a subset process?	
			Does the single transaction can operate in multiple ways?	
			Is the same function is available through a number of proces	
			Is the transaction at the the level of calculating an individua	
			Is the transaction exist for technical reason?	
			Is the transaction achieves a business goal?	

Actions		@ #	Description

Table 3 Questions

La Figure 18 montre comment est traitée une question par l'expert.

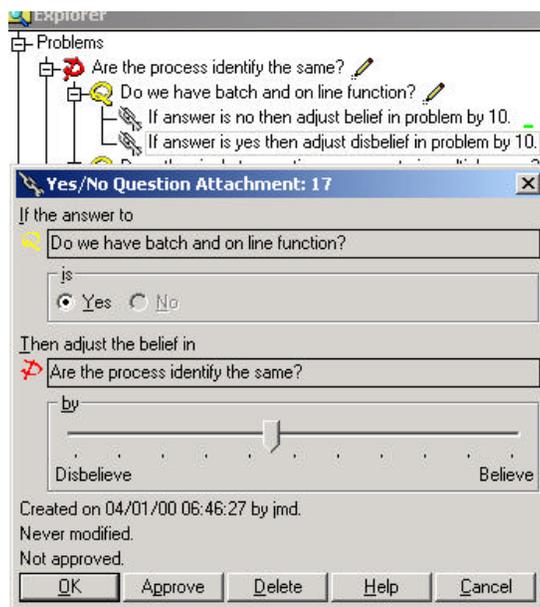


Table 4 Traitement d'une question par l'expert

Il est possible à l'expert de noter un pourcentage sur la valeur de la réponse, ici appelée croyance positive ou négative. Help!CPR permet des réponses aux questions avec des choix multiples et avec des nombres. Ces options ne sont pas montrées à l'écran. Dans le contexte de COSMIC-FFP ces options ne sont pas utilisées, sur la base des expériences que nous avons réalisées à date. Nous ne rejetons pas cependant définitivement ces options, mais il nous faudra des expérimentations plus poussées pour les valider.

A.2.3 Les actions

La section action est reliée aux problèmes. Dans le modèle de diagnostic, nous utiliserons l'action pour inciter un novice à regarder d'autres problèmes apparaissant normalement avec le type de problème posé. Les exemples d'action de Help!CPR ne sont pas de cette nature : lors du diagnostic d'une imprimante, on suggérera comme action de vérifier la connexion par exemple. Notre intention est de pouvoir associer d'autres problèmes au problème du novice (les mots clefs permettent les réponses aux questions à d'autres problèmes). Ce serait une forme d'apprentissage guidé en tenant compte du contexte des réponses aux questions.

A.2.4 Les mots

Il est aussi possible, pour un utilisateur, de voir la liste des mots clés qu'il peut utiliser pour résoudre un problème. Il y a trois catégories de mots (Figure 19) : les mots clefs, les mots qu'il faut ignorer et les mots non classifiés. La classification des mots devrait se faire par l'expert dans le mode sélection. Il suffit de sélectionner un mot dans la liste et d'indiquer s'il s'agit :

- d'un mot clef ou
- d'un mot à ignorer

Word	Category	Count	Mean	Deviation
identifier	I	9	1.	0.
p	I	9	1.	0.
of	I	5	1.	0.
an	I	3	1.	0.
a	I	2	1.	0.
functional	K	2	1.	0.
process	K	2	1.	0.
application	K	1	1.	0.
boundary	K	1	1.	0.
data	K	1	1.	0.
entry	K	1	1.	0.
exit	K	1	1.	0.
group	K	1	1.	0.
layers	K	1	1.	0.
read	K	1	1.	0.
write	K	1	1.	0.

Table 5 Les mots

Les mots peuvent apparaître par ordre alphabétique ou selon leur nombre d'occurrences. Il est aussi possible de choisir s'il faut montrer une (ou des) catégorie(s) de mots.

La somme des mots indique combien de fois un mot apparaît au moins une fois dans la description d'un problème ou dans la documentation. La moyenne fait référence au nombre moyen de fois qu'un mot apparaît au moins une fois dans la chaîne de caractères de la description d'un problème ou dans la documentation. La déviation réfère à la déviation à partir de la moyenne.

La table des mots (Figure 20) est extraite des descriptions des problèmes et de leur documentation. Il est possible de réindexer cette table à tout moment.

Word	Category	Count	Mean	Deviation
functional	K	2	1.	0.
process	K	2	1.	0.
application	K	1	1.	0.
boundary	K	1	1.	0.
data	K	1	1.	0.
entry	K	1	1.	0.
exit	K	1	1.	0.
group	K	1	1.	0.
layers	K	1	1.	0.
read	K	1	1.	0.
write	K	1	1.	0.

Table 6 Les mots (suite)

Annexe B: description du prototype 1

Annexe B, partie 1: Exigences détaillées (English)

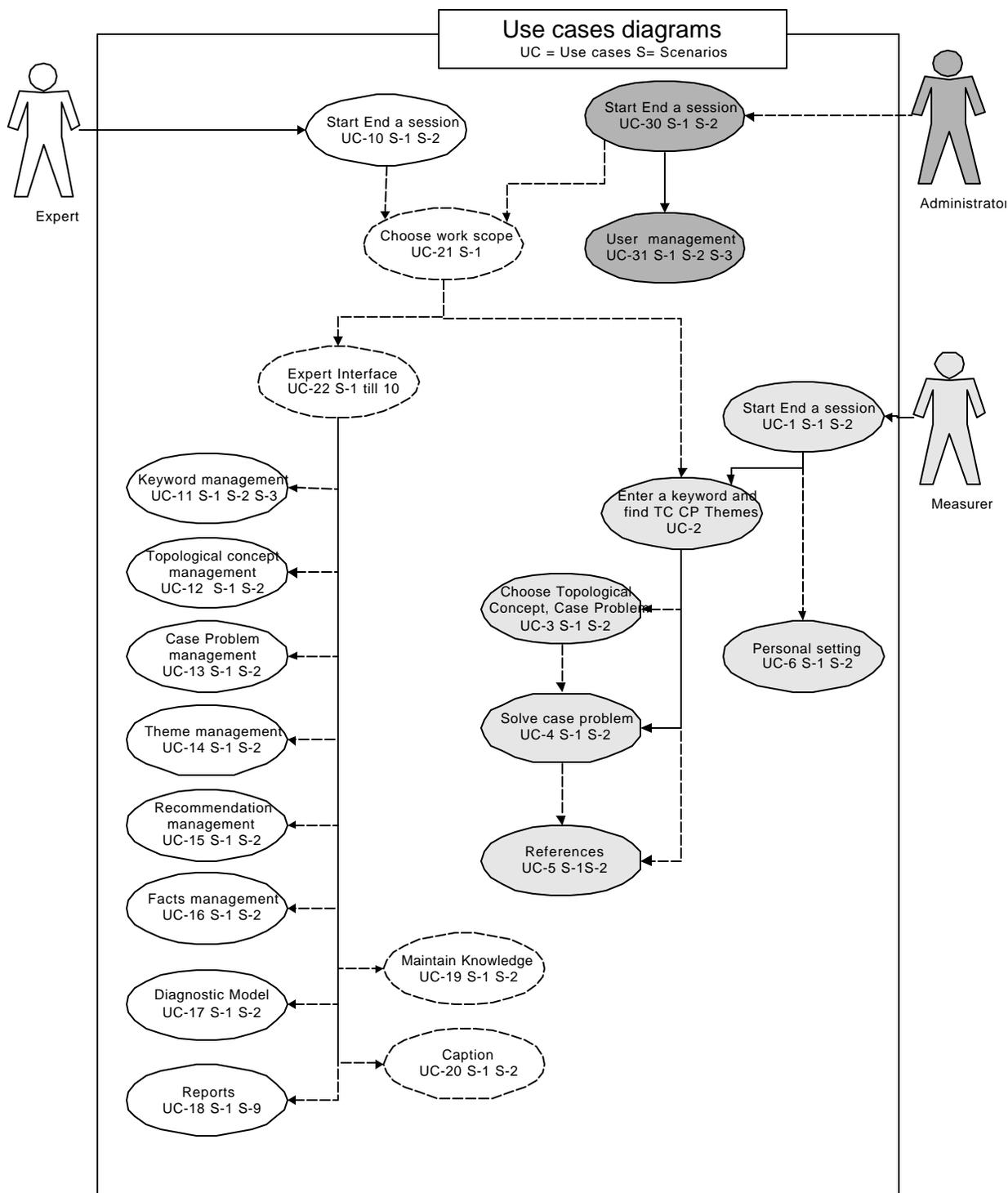


Figure 1 Diagramme de cas

Voici un tableau avec les acteurs, événements et entrée/sortie (E) et (S) du prototype pour le mesureur.

Acteur	Événement	Entrée/Sortie
Mesureur	Entrer un mot clef	(E)
Mesureur	Rechercher les concepts topologiques et cas problèmes	(E)
Mesureur	Sélectionner les concepts topologiques et cas problèmes désirés	(E)
Mesureur	Sélectionner les faits sur la base des informations disponibles via la documentation	(E)
Mesureur	Entrer une information sur la qualité de la documentation en regard avec un thème particulier (Note: a été enlevé dans le deuxième prototype)	(E)
Mesureur	Recommandation appropriée	(S)
Mesureur	Réponse ou valeur sur la qualité de la documentation	(E)
Mesureur	Demander comment compacter la base de données	(E) et (S)
Mesureur	Demander une explication sur un mot clef, un concept topologique, un cas problème, un thème ou une recommandation spécifique.	(E) et (S)
Mesureur	Demander une description de ce qu'ent une ontologie, un mot clef, un concept topologique, un cas problème, un thème ou une recommandation.	(E) et (S)
Mesureur	Requête sur ses paramètres utilisateur	(S)
Mesureur	Modifier le nom de l'utilisateur, le mot de passe et le langage	(E)

Tableau 9 Scénarios du mesureur

Voici un tableau avec les acteurs, événements et entrée/sortie (E) et (S) du prototype pour l'expert et l'administrateur.

Acteur	Événement	Entrée/Sortie
Expert	Ajouter, modifier et détruire un cas spécifique de connaissance	(E)
Expert	Modifier les boutons, en-têtes, titres et autres des interfaces	(E)
Expert	Demander des rapports tels le suivi du mesureur, la session, ou encore les relations entre les connaissances d'expertise.	(S)
Expert	Montrer une description sommaire d'un noeud dans un arbre lorsque sélectionné	(S)
Expert	Copier, couper et coller des noeuds dans l'arbre	(E)
Expert	Demander les propriétés des cas spécifiques des connaissances	(S)
Expert	Sélectionner l'outil de diagnostique désiré	(E)
Expert	Maintenir des cas spécifiques de connaissance dans différents langages (Note: pas dans le deuxième prototype)	(E)
Expert	Détruire le rapport de suivi du mesureur ou la session	(E)

Acteur	Événement	Entrée/Sortie
Expert	Aller vers les fonctionnalités du mesureur et résoudre un problème	(E)
Admin	Ajouter, modifier ou détruire un utilisateur	(E)
Admin	Utiliser les fonctionnalités de l'expert	(E) et (S)
Admin	Utiliser les fonctionnalités du mesureur	(E) et (S)

Tableau 1 Scénarios de l'expert et de l'administrateur

Use case 1: Start/End a session Scenario 1: Start a session
Description: Enter username, password and click 'OK' button. Primary actor: Measurer Secondary actor: None Pre conditions User registered to the system with the status "MEASURER" Exception: If user does not exist or expired, or password not valid, display error message. Post condition: see Figure 42 Classes implied: user, session, error message Interface: see Figure 40
Calculations: No

Use case 1: Start/End a session Scenario 2: End a session
Description: When the Measurer clicks the 'Exit' button or the close button of a form, the application will be terminated. Primary actor: Measurer Secondary actor: None Pre conditions: UC-1, S-1 Exception: None Post condition: Measurer Follow Up if Themes have been answered; Program ends. Classes implied: Session, Keyword, Topology, Problem, Theme, Facts, Recommendation, Subsession, User
Interface: see Figure 42
Calculations: No

Use case 2: Enter a keyword and find TCs2, CPs3, Themes and R4 Scenario 1: Enter a keyword
Description: Measurer enters a keyword into a text box and starts search Primary actor: Measurer Secondary actor: None Pre conditions: Existence of KW AND relationships between KW5 and TCs, TC and CPs, CP and Themes Exception: If keyword does not exist and error message will be displayed Post condition: All related items to the chosen keyword will be displayed in Figure 42 Classes implied: keyword, topology, case problem, theme, recommendation, measurer follow-up, session, answer, results, parameters, adjustment Interface: see Figure 42
Calculations: No

2 Topological Concepts

3 Case problems

4 Recommendations

5 Keyword

Use case 3: Choose Topological Concepts, case problems Scenario 1: Choose Topological Concepts
Description: From the list of displayed TC(s) related to the keyword, the measurer can make a choice which he wants to keep. Primary actor: Measurer Secondary actor: None Pre conditions: Use case 2, Scenario 1. Exception: If there is no TC displayed, the user does not have the ability to choose. Post condition: Only the CP and Themes related to the chosen TC(s) will be displayed. Classes implied: topology, case problem, Theme, recommendation Interface: see Figure 42
Calculations: No

Use case 3: Choose Topological Concepts, case problems Scenario 2: Choose case problem
Description: From the list of displayed CP(s) related to the keyword, the measurer can make a choice which he wants to keep. Primary actor: Measurer Secondary actor: None Pre conditions: Use case 2, Scenario 1. Exception: If there is no CP displayed, the user does not have the ability to choose. Post condition: Only the Themes related to the chosen CP(s) will be displayed. Classes implied: : topology, case problem, Theme, Recommendation Interface: see Figure 42
Calculations : No

Use case 4: Solve Case Problem Scenario 1: Select facts
Description: The user chooses one fact from the selection of possible facts belonging to each Theme Primary actor: Measurer Secondary actor: None Pre conditions: UC 2, S-1 OR UC 3, S-1. Exception: No Post condition: The choosen fact will be displayed. Classes implied: Theme, fact Interface: see Figure 46
Calculations: Yes

Use case 4: Solve Case Problem Scenario 2: Enter quality of documentation
Description: Provide system with the quality of your documentation to select a fact for this Theme. Primary actor: Measurer

Secondary actor: None Pre conditions: : UC 2, S-1 OR UC 3, S-1 OR UC 4, S-1 Exception: Default value is set to 100%. Post condition: The entered percentage will be displayed Classes implied: Adjustment, Fact, Measurer follow-Up Interface: see Figure 47
Calculations: Yes

Use case 5: References Scenario 1: Ask for reference
Description: the user asks to see a reference (rich text file format) for a specific KW, TC, CP, Theme, Recommendation Primary actor: Measurer Secondary actor: None Pre conditions: Existence of a rtf file AND UC 2 Exception: If no file exist with that name, error message Post condition: Loaded rich text file Classes implied: Reference, Keyword, Topology, Problem, Theme, Recommendation, Fact Interface: see Figure 42 and 44
Calculations: No

Use case 5: References Scenario 2: Ask for definitions
Description: the user has the possibility to get a definition of what is a KW, TC, CP, Theme, Recommendation and Ontology Primary actor: Measurer Secondary actor: None Pre conditions: Existence of a rtf file AND UC 1, S-1 Exception: If no file exist with that name, error message Post condition: Loaded rich text file Classes implied: None Interface: see Figure 42 and 44
Calculations: No

Use case 6: Personal setting Scenario 1: Query personal settings
Description: User wants to see his personal settings. Primary actor: Measurer, Expert, Administrator Secondary actor: None Pre conditions: UC 1, S-1 Exception: None Post condition: The personal setting are displayed in Figure 45

Classes implied: user
Interface: see Figure 45
Calculations: No

Use case 6: Personal setting
Scenario 2: Change personal settings
Description: User is able to change the language of program, his username or password.
Primary actor: Measurer
Secondary actor: None
Pre conditions: Use case 6, scenario 1.
Exception: Error message if user didn't confirmed his new password or if the confirmed password differs from the new password.
Post condition: Changes are updated in the database.
Classes implied: User
Interface: see Figure 45
Calculations: No

EXPERT

Use case 10: Start/End a session
Scenario 1: Start a session
Description: Enter username, password and click 'OK' button.
Primary actor: Expert
Secondary actor: None
Pre conditions: User registered to the system with the status "EXPERT"
Exception: If password is incorrect, user does not exist or his time is expired, display error message.
Post condition: Use case 21, session started
Classes implied: Session, User
Interface: see Figure 40
Calculations: No

Use case 10: Start/End a session
Scenario 2: End a session
Description: Exit the program
Primary actor: Expert
Secondary actor: None
Pre conditions: Use case 10, scenario 1.
Exception: No
Post condition: Program determined
Classes implied: Session
Interface: see Figure 49 or Figure 42

Calculations: No
Use case 11: Keyword management Scenario 1: Add a keyword
Description: Add a keyword to the system by entering its name, description and rtf file Primary actor: Expert Secondary actor: None Pre conditions: Use case 22, Scenario 1 Exception: Keyword and keyword description are required. If one or both are not provided display error message. Post condition: Ready to enter another keyword (use case 11, scenario 1) Classes implied: Keyword, Reference Interface: see Figure 53
Calculations: No
Use case 11: Keyword management Scenario 2: Modify keyword
Description: User is able to modify the description of selected keyword as well as to make changes to his related Topological Concepts (name, description, rft file and relationship to the keyword) Primary actor: Expert Secondary actor: None Pre conditions: UC 11, S-1 AND UC 22, S-1 Exception: Name and Description have to be less than 256. Post condition: Ready to modify other keyword Classes implied: Keyword, Topology, Reference Interface: see Figure 54
Calculations: No
Use case 11: Keyword management Scenario 3: Delete keyword
Description: Delete a keyword by choosing it from a list Primary actor: Expert Secondary actor: None Pre conditions: (UC 11, S-1 OR S-2) AND UC 22, S-1 Exception: None Post condition: Ready to delete other keyword Classes implied: Keyword, Topology, Reference Interface: see Figure 60
Calculations: No
Use case 12: Topological Concept management Scenario 1: Add a Topological Concept
Description: Add a Topological Concept to a keyword Primary actor: Expert

Secondary actor: None Pre conditions: UC 11, S-1 Exception: None Post condition: UC 12, S-1. Classes implied: Keyword, Topology, Reference Interface: see Figure 58
Calculations: No

Use case 12: Topological Concept management Scenario 2: Modify a Topological Concept
Description: Modify the name of a TC, its description, rtf file and relationship to a keyword Primary actor: Expert Secondary actor: None Pre conditions: Existence of relationship between TC and at least one Keyword Exception: Name and Description have to be less than 256 characters and the relationship percentage has to be between 0% and 100% Post condition: UC12, Scenario 2 Classes implied: Keyword, Topology, Reference Interface: see Figure 54
Calculations: No

Use case 13: Case Problem management Scenario 1: Add a case problem
Description: Add a Case Problem to a Topological Concept Primary actor: Expert Secondary actor: None Pre conditions: UC 12, S-1 AND UC 22, S-3 Exception: Name of the CP, its description and relationship to the TC are required. If not provided show error message. Post condition: Ready to add another Case Problem or return to use case 22, scenario 3 Classes implied: Topology, Case Problem, Reference Interface: see Figure 58
Calculations: No

Use case 13: Case Problem management Scenario 2: Modify a case problem
Description: Modify the name of the case problem, its description, rtf file and the relationship to the TC Primary actor: Expert Secondary actor: None Pre conditions: Use case 22, scenario 3 AND UC 13, S-1 Exception: Name and Description have to be less than 256 characters and the relationship percentage has to be between 0% and 100% Post condition: Ready to modify another Case Problem related to the same TC

Classes implied: Topology, Case Problem, Reference
Interface: see Figure 56
Calculations: No

Use case 14: Theme management
Scenario 1: Add a Theme
Description: Add a Theme to a case problem
Primary actor: Expert
Secondary actor: None
Pre conditions: UC 13, S-1 AND UC 22, S-3
Exception: Required information in the Theme name, its description and the relationship to the case problem. If not entered an error message will appear.
Post condition: UC 14, S-1
Classes implied: Case Problem, Theme, Reference
Interface: see Figure 58
Calculations: No

Use case 14: Theme management
Scenario 2: Modify a Theme
Description: Modify the name of a Theme, the description, rtf file and the relationship to its case problem
Primary actor: Expert
Secondary actor: None
Pre conditions: UC 14, S-1 AND UC 22, S-3
Exception: Name and Description have to be less than 256 characters and the relationship percentage has to be between 0% and 100%
Post condition: UC 14, S-2
Classes implied: Case Problem, Theme, Reference
Interface: see Figure 59
Calculations: No

Use case 15: Recommendation management
Scenario 1: Add recommendation
Description: Add a recommendation to a case problem
Primary actor: Expert
Secondary actor: None
Pre conditions: UC 13, S-1 AND UC 22, S-3
Exception: Required information in the recommendation name, its description and the relationship to the case problem. If not entered an error message will appear.
Post condition: Add another recommendation or Use case 22, scenario 3
Classes implied: Case Problem, Recommendation, Reference
Interface: see Figure 58
Calculations: No

Use case 15: Recommendation management
--

Scenario 2: Modify recommendation
Description: Modify the name of recommendations, its description, rtf file and the relationship to its case problem. Primary actor: Expert Secondary actor: None Pre conditions: UC 15, S-1 AND UC 22, S-3 Exception: Name and Description have to be less than 256 characters and the relationship percentage has to be between -100% and +100% Post condition: UC15, S-2 Classes implied: Case Problem, Recommendation, Reference Interface: see Figure 59
Calculations: No

Use case 16: Fact management Scenario 1: Add fact
Description: Add a recommendation to a Theme Primary actor: Expert Secondary actor: None Pre conditions: UC 14, S-1, UC 22, S-3 Exception: Required information is the fact name, its description and the relationship to the Theme. If not entered an error message will appear. Post condition: UC16, S-1 Classes implied: Theme, Fact, Reference Interface: see Figure 58
Calculations: No

Use case 16: Fact management Scenario 2: Modify fact
Description: Modify the name of the fact, its description, rtf file and the relationship to its Theme. Primary actor: Expert Secondary actor: None Pre conditions: UC 16, S-1 AND UC 22, S-3 Exception: Name and Description have to be less than 256 characters and the relationship percentage has to be between -100% and +100% Post condition: UC 16, S-2 Classes implied: Theme, Fact, Reference Interface: see Figure 56
Calculations: No

Use case 17: Diagnostic model Scenario 1: Show models
Description: Display three different diagnostic models Primary actor: Expert Secondary actor: None

Pre conditions: User case 22, scenario 1 Exception: Default is diagnostic model 1. Post condition: Use case 17, scenario 2 Classes implied: Parameters Interface: see Figure 51
Calculations: No

Use case 17: Diagnostic model Scenario 2: Select a diagnostic model
Description: Select one of the three diagnostic models and hit button 'OK' Primary actor: Expert Secondary actor: None Pre conditions: Use case 17, scenario 1 Exception: No Post condition: Use case 22, scenario 1 Classes implied: Parameters Interface: see Figure 51
Calculations: No

Use case 18: Reports Scenario 1: Show measurer follow up
Description: Populate follow up data in an Excel sheet Primary actor: Expert Secondary actor: None Pre conditions: Use case 22, scenario 1 AND open Excel workbook Exception: No Post condition: Use case 22, scenario 1 Classes implied: Answers, Results Interface: Excel workbook
Calculations: No

Use case 18: Reports Scenario 2: Delete Measurer follow up information
Description: Delete the information to produce the Excel follow up file in the databae. Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S-1 Exception: none Post condition: UC 18 Classes implied: Measurer Follow Up Interface: see Figure 62
Calculations: No

Use case 18: Reports Scenario 3: Show session report

Description: Populate session parameters in an Excel sheet Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S-1 Exception: No Post condition: UC 22, S-1 AND open Excel workbook Classes implied: Session Interface: Excel workbook
Calculations: No

Use case 18: Reports Scenario 4: Delete session report
Description: Deletes the content of the database which is necessary to build th Session-report Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S-1 Exception: No Post condition: UC 18 Classes implied: Session Interface: see Figure 62
Calculations: No

Use case 18: Reports Scenario 5: Show relationship between KWs and TCs
Description: Populate how Keywords are related with Topological Concepts Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S-1 Exception: No Post condition: UC 22, S-1 Classes implied: Session Interface: see Figure 52
Calculations: No

Use case 18: Reports Scenario 6: : Show relationship between TCs and CPs
Description: Populate how Topological Concepts are related to Case Problems Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S-1 Exception: No Post condition: UC 22, S-1 Classes implied: Session Interface: see Figure 52

Calculations: No

Use case 18: Reports

Scenario 7: : Show relationship between CPs and Themes
--

Description: Populate how Case Problems are related to Themes

Primary actor: Expert

Secondary actor: None

Pre conditions: UC 22, S-1

Exception: No

Post condition: UC 22, S-1

Classes implied: Session

Interface: see Figure 52

Calculations: No

Use case 18: Reports

Scenario 8: Show relationship between CPs and Recommendations

Description: Populate how Case Problems are related to Recommendations
--

Primary actor: Expert

Secondary actor: None

Pre conditions: UC 22, S-1

Exception: No

Post condition: UC 22, S-1

Classes implied: Session

Interface: see Figure 52

Calculations: No

Use case 18: Reports

Scenario 9: Show relationship between Themes and Facts
--

Description: Populate how Themes are related to Facts

Primary actor: Expert

Secondary actor: None

Pre conditions: UC 22, S-1

Exception: No

Post condition: UC 22, S-1

Classes implied: Session

Interface: see Figure 52

Calculations: No

Use case 19: Maintain Knowledge

Scenario 1: Query case specific data in all languages

Description: The expert can choose from a drop down list what knowledge he wants to maintain, Keywords, TCS, etc.

Primary actor: Expert

Secondary actor: None

Pre conditions: UC 22, S-1

Exception: No Post condition: UC 22, S-1 Classes implied: Keyword, Topology, Case Problem, Theme, Recommendation, Fact Interface: see Figure 61
Calculations: No

Use case 19: Maintain Knowledge Scenario 2: Change case specific data in all languages
Description: The expert can the name and description in the different languages for all Keywords, TCs ... Primary actor: Expert Secondary actor: None Pre conditions: UC 19, S-1 Exception: No Post condition: UC 19, S-2 Classes implied: Keyword, Topology, Case Problem, Theme, Recommendation, Fact Interface: see Figure 61
Calculations: No

Use case 20: Caption Scenario 1: Query for captions of user interfaces
Description: Show captions of the complete program used for button, labels, form titles etc. in all languages Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S-1 Exception: No Post condition: UC 20, S-2 Classes implied: Language Interface: see Figure 50
Calculations: No

Use case 20: Caption Scenario 2: Adapt captions of user interfaces
Description: Change captions of buttons, labels, form titles etc. (all languages) Primary actor: Expert Secondary actor: None Pre conditions: UC 20, S-1 Exception: None Post condition: UC 22, S-1 Classes implied: Language Interface: see Figure 50
Calculations: No

Use case 21: Choose work scope

Scenario 1: Choose work scope
Description: Measurer selects if he wants to solve a problem or maintain the system Primary actor: Expert Secondary actor: None Pre conditions: UC 10, S-1 Exception: No Post condition: Use case 2 (to solve a problem) or Use case 22 (maintain the system) Classes implied: None Interface: see Figure 48
Calculations: No

Use case 22: Expert Interface Scenario 1: Query for structure of case specific knowledge
Description: Display the hierarchical structure of case specific knowledge in tree view Primary actor: Expert Secondary actor: None Pre conditions: Existence of expert knowledge Exception: none Post condition: Use case 22 Classes implied: Keyword, Topology, Case Problem, Theme, Fact, Recommendation Interface: see Figure 49
Calculations: No

Use case 22: Expert Interface Scenario 2: Description
Description: Description of the highlighted item in the tree view will appear in the text box. Primary actor: Expert Secondary actor: None Pre conditions: Existence of expert knowledge Exception: none Post condition: Use case 22 Classes implied: Topology, Case Problem, Theme, Fact, Recommendation Interface: see Figure 49
Calculations: No

Use case 22: Expert Interface Scenario 3: Properties
Description: When right-clicking on an item in the tree view a menu appear. Hitting “properties” will lead to a screen where modifications to related items can be made. Primary actor: Expert Secondary actor: None Pre conditions: Existence of expertise knowledge Exception: none Post condition: Use case 22

Classes implied: Topology, Case Problem, Theme, Fact, Recommendation
Interface: Figure 56 OR Figure 57
Calculations: No

Use case 22: Expert Interface Scenario 4: Cut
Description: Highlighted item can be cut into the temporary storage by employing the menu “Edit-cut” or using the shortcut “Strg+X” Primary actor: Expert Secondary actor: None Pre conditions: Existence of expert knowledge Exception: Post condition: Classes implied: Interface: see Figure 49
Calculations: No

Use case 22: Expert Interface Scenario 5: Paste
Description: Items in the temporary storage can be entered, employing the menu “Edit-paste” or using the shortcut “Strg+V” Primary actor: Expert Secondary actor: None Pre conditions: UC 22, S4 OR S-8 Exception: none Post condition: Item has been pasted to the new location Classes implied: Topology, Case Problem, Theme, Fact, Recommendation Interface: see Figure 49
Calculations: No

Use case 22: Expert Interface Scenario 6: Delete
Description: Highlighted item can be deleted by employing the menu “Edit-delete” or hitting the button “del” Primary actor: Expert Secondary actor: None Pre conditions: Existence of expert knowledge Exception: Post condition: Node is removed from the tree view and deleted in the database Classes implied: Topology, Case Problem, Theme, Fact, Recommendation Interface: see Figure 49
Calculations: No

Use case 22: Expert Interface Scenario 7: Drag&Drop
--

<p>Description: Nodes in the tree view can be drag&dropped like in the Windows Explorer.</p> <p>Primary actor: Expert</p> <p>Secondary actor: None</p> <p>Pre conditions: Existence of expert knowledge</p> <p>Exception: none</p> <p>Post condition: The dragged item has been copied to the new location</p> <p>Classes implied: Topology, Case Problem, Theme, Fact, Recommendation</p> <p>Interface: see Figure 49</p>
Calculations: No

<p>Use case 22: Expert Interface</p> <p>Scenario 8: Copy</p>
<p>Description: Highlighted item can be copied into the temporary storage by employing the menu "Edit-copy" or using the shortcut "Strg+C"</p> <p>Primary actor: Expert</p> <p>Secondary actor: None</p> <p>Pre conditions: Existence of expert knowledge</p> <p>Exception:</p> <p>Post condition:</p> <p>Classes implied:</p> <p>Interface: see Figure 49</p>
Calculations: No

ADMINISTRATOR

<p>Use case 30: Start/End a session</p> <p>Scenario 1: Start a session</p>
<p>Description: Enter username, password and click 'OK' button.</p> <p>Primary actor: Administrator</p> <p>Secondary actor: None</p> <p>Pre conditions User registered to the system with the status "ADMINISTRATOR"</p> <p>Exception: If user does not exist or the password is not correct, display error message.</p> <p>Post condition: see Figure 40</p> <p>Classes implied: user, session, error message</p> <p>Interface: see Figure 40</p>
Calculations: No

<p>Use case 30: Start/End a session</p> <p>Scenario 2: End a session</p>
<p>Description: Exit the program</p> <p>Primary actor: Administrator</p> <p>Secondary actor: None</p> <p>Pre conditions: UC 30, S-1</p>

Exception: None Post condition: Program terminated Classes implied: Session Interface: see Figure 42 OR Figure 49
Calculations: No

Use case 31: User management Scenario 1: Add a user
Description: Add a new user to the system Primary actor: Administrator Secondary actor: None Pre conditions: UC 30,S-1 Exception: User status has to be either “Measurer”, “Expert” or “Administrator”. Valid languages are English, French, German and Italian. Valid diagnostic-models values are 0, 1 or 2 Post condition: UC 31 Classes implied: User Interface: see Figure 63
Calculations: No

Use case 31 User management Scenario 2: Modify user
Description: Modify user parameter Primary actor: Administrator Secondary actor: None Pre conditions: UC 30,S-1 Exception: User status has to be either “Measurer”, “Expert” or “Administrator”. Valid languages are English, French, German and Italian. Valid diagnostic-models values are 0, 1 or 2 Post condition: UC 31 Classes implied: User Interface: see Figure 63
Calculations: No

Use case 31: User management Scenario 1: Delete a user
Description: Delete a user Primary actor: Administrator Secondary actor: None Pre conditions: UC 30,S-1 Exception: Post condition: UC 31 Classes implied: User Interface: see Figure 63
Calculations: No

Functional and Data Requirements

Functional Requirements.

Measurer

Requirement #: 1

Requirement Type: M_login

Use case #: 1

Description: Login to the system and open a new session by entering the *username* and *password***Rationale:** Only customers who paid for the product should have access**Source:** Jean-Marc Desharnais**Fit Criterion:** User who is not enlisted in the user-database or have an expired license may not have access to the program**Customer Satisfaction:** 5**Customer Dissatisfaction:** 5**Dependencies:****Requirement #: 2**

Requirement Type: M_LogIn

Use case #: 4

Description: Automatically open a new *subsession* everytime the measurer searches for a *keyword***Rationale:** For statistic reasons**Source:** Jean-Marc Desharnais , Tim Kuessing**Fit Criterion:** Cross check with session report, if sessions (and subsession) have been registered correct.**Customer Satisfaction:** 5**Customer Dissatisfaction:** 5**Dependencies:**

Requirement #: 3

Requirement Type: M_KW

Use case #: 2

Description: The measurer can browse and select a keyword from a list**Rationale:** Makes it easier for the measurer to find a matching keyword to his needs**Source:** Jean-Marc Desharnais**Fit Criterion:** Pop-up menu with keywords

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies: The keywords will be supplied from a database**Requirement #: 4**

Requirement Type: M_KW

Use case #: 5

Description: Displaying an rtf-reference for each keyword in the list on right-click (keyword has to be highlighted)**Rationale:** Additional information for the measurer.**Source:** Jean-Marc Desharnais, Tim Küssing**Fit Criterion:** Appearance of rtf file belonging to specific keyword

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Requirement #: 5

Requirement Type: M_MainWindow

Description: Query for the related Topological Concepts, problems and questions after choosing a keyword

Rationale: To enable the measurer to exclude topics or problems from whom he knows they are not relevant to solve his problem.

Source: Jean-Marc Desharnais

Fit Criterion: Display only those topological concepts and case problems (Themes and recommendations) the user selected

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies: The relationships will be supplied by a database

Requirement #: 6

Requirement Type: M_MainWindow

Use case #: 3

Description: Enable the measurer to exclude certain Topological Concepts from problem solving process

Rationale: The measurer should be able to eliminate possible of case problems from whom he knows are not relevant

Source: Jean-Marc Desharnais

Fit Criterion: Delete TCs and re-run query with the selected TCs

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 7

Requirement Type: M_MainWindow

Use case #: 3

Description: Enable the measurer to exclude certain case problems from problem solving process

Rationale: The measurer should be able to eliminate possible case problems from whom he thinks are not relevant

Source: Jean-Marc Desharnais

Fit Criterion: Delete Case Problems and re-run query with the selected Case Problems

Customer Satisfaction: 5

Customer Dissatisfaction: 5
Dependencies:

Requirement #: 8

Requirement Type: M_MainWindow

Use case #: 4

Description: The measurer will be able to select a fact from a variety of possible answers (yes/no or multiple choice) according to a Theme.

Rationale: The mesurer has to use pre-defined facts set up by the expert to perform the calculation

Source: Jean-Marc Desharnais

Fit Criterion: Select a fact belonging to a specific Theme

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies: The answers will be stored in a database

Requirement #: 9

Requirement Type: M_MainWindow

Use case #: 4

Description: Based on the raised Topological Concepts, case problems, Themes and selected facts, the diagnostic system has to automatically calculate one or more recommendations.

Rationale: Main task of a diagnostic tool, supply the user with problem solving strategies

Source: Jean-Marc Desharnais

Fit Criterion: As soon as a Theme has been answered one recommendation will appear.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 10

Requirement Type: M_MainWindow

Use case #: 5

Description: Displaying an explanation for a Topological Concept

Rationale: Tool has to be 'transperen' so the user can increase his own expertise.

Source: Jean-Marc Desharnais

Fit Criterion: Right click on a specific Topological Concept and choose *explanation* from the pop-up window, a rich text file will be displayed.

Customer Satisfaction: 4
 Customer Dissatisfaction: 4
 Dependencies:

Requirement #: 11

Requirement Type: M_MainWindow Use case #: 5
Description: Displaying an explanation for a Case Problem
Rationale: Tool has to be 'transparen' so the user can increase his own expertise.
Source: Jean-Marc Desharnais
 Fit Criterion: Right click on a specific Case Problem and choose *explanation* from the pop-up window, a rich text file will be displayed.
 Customer Satisfaction: 4
 Customer Dissatisfaction: 4
 Dependencies:

Requirement #: 12

Requirement Type: M_MainWindow Use case #: 5
Description: Displaying an explanation for a Theme
Rationale: Tool has to be 'transparen' so the user can increase his own expertise.
Source: Jean-Marc Desharnais
 Fit Criterion: Right click on a specific Theme and choose *explanation* from the pop-up window, a rich text file will be displayed.
 Customer Satisfaction: 4
 Customer Dissatisfaction: 4
 Dependencies:

Requirement #: 13

Requirement Type: M_MainWindow Use case #: 5
Description: Displaying an explanation for a Recommendation
Rationale: Tool has to be 'transparent' so the user can increase his own expertise.
Source: Jean-Marc Desharnais
 Fit Criterion: Right click on a specific Recommendation and choose *explanation* from the pop-up window, a rich text file will be displayed.
 Customer Satisfaction: 4
 Customer Dissatisfaction: 4
 Dependencies:

Requirement #: 14

Requirement Type: M_MainWindow

Use case #: 5

Description: The user can see an explanation of what is a Topological Concept.**Rationale:** Helps the user understand the system better.**Source:** Jean-Marc Desharnais

Fit Criterion: User hits a button and the explanation will come up

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Conflicts:

Supporting Materials:

Requirement #: 15

Requirement Type: M_MainWindow

Use case #: 5

Description: The user can see an explanation of what the Case Problem is.**Rationale:** Helps the user understand the system better.**Source:** Jean-Marc Desharnais

Fit Criterion: User hits a button and the explanation will come up

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Requirement #: 16

Requirement Type: M_MainWindow

Use case #: 5

Description: The user can see an explanation of what the Theme is.**Rationale:** Helps the user understand the system better.**Source:** Jean-Marc Desharnais

Fit Criterion: User hits a button and the explanation will come up

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Requirement #: 17

Requirement Type: M_MainWindow

Use case #: 5

Description: The user can see an explanation of what the Recommendation is.**Rationale:** Helps the user understand the system better.**Source:** Jean-Marc Desharnais

Fit Criterion: User hits a button and the explanation will come up

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Requirement #: 18

Requirement Type: M_Settings

Use case #: 6

Description: Changing the username and password of an user

Rationale: Standard

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: Changing the password inside the program.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 19

Requirement Type: M_Db

Use case #: 5

Description: Instructions on how to compact the database**Rationale:** MS Access databases don't release storage space after deleting a row. As a result of that the application will become slow after using it, compacting it would solve this issue.

Source : Tim Kuessing

Fit Criterion:

Customer Satisfaction: 1

Customer Dissatisfaction: 1

Dependencies:

Requirement #: 20

Requirement Type: M_language

Use case #: 6

Description: Change of the current language of the program during run time. This will involve the menu bar, all buttons and labels as well as the case specific content of the expert system.

Rationale: Due to different nationality of users, they should be able to switch between the languages of interfaces and the knowledge base.

Source: Jean-Marc Desharnais

Fit Criterion: After changing the language, all forms and their content have to be in the desired language without re-installing or restarting the program

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 21

Requirement Type: M_Error

Use case #: none

Description: Exception handling to ensure that the tool is running stable (defined state). Inform user about problems through error warnings.

Rationale: To ensure system and data integrity

Source: Jean-Marc Desharnais

Fit Criterion:

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 22

Requirement Type: M_MainWindow

Use case #: 4

Description: Enter the quality of the available documentation with a slider**Rationale:** Needed to verify uncertain information**Source:** Jean-Marc Desharnais

Fit Criterion: Enter percentage for the quality of documentation

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 23

Requirement Type: E_Report

Use case #: 4

Description: Measurer activities will be reported at the end of the problem solving process. Log information about answered facts (thereunto KW, TC, CP, Theme), calculated recommendation plus session infos.**Rationale:** For statistic reasons.**Source:** Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: Produce Excel report about measurer activities

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Expert part:

Requirement #: 24

Requirement Type: E_Manage

Use case #: 11

Description: Add a keyword

Rationale: Knowledge acquisition

Source: Jean-Marc Desharnais

Fit Criterion: Successfully add a keyword means to enter the name, its description and path of rtf-file containing its explanation. Furthermore a user shall see what keywords are already registered to the system.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 25

Requirement Type: E_Manage

Use case #: 11

Description: Delete a keyword

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: All related information (description, linked Topological Concepts) to this keyword has to be deleted, but if a related Topological Concepts is also related to other keywords, then don't delete it. On the other hand if a Topological Concept is only related to one particular keyword, then ask the user if he wants to delete it, too.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 26

Requirement Type: E_Manage

Use case #: 11

Description: Modify a keyword

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: The user has to be able to change the description of a keyword and to modify and delete related Topological Concepts.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 27

Requirement Type: E_Manage

Use case #: 12

Description: Add a Topological Concept

Rationale: Knowledge acquisition

Source: Jean-Marc Desharnais

Fit Criterion: Successfully add a Topological Concept with its description, its path to the reference file and a relationship percentage to a keyword

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 28

Requirement Type: E_Manage

Use case #:12

Description: Delete a Topological Concept

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Delete all the related information to this Topological Concept as well as related Case Problems.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 29

Requirement Type: E_Manage

Use case #: 12

Description: Modify a Topological Concept

Rationale: Knowledge acquisition

Source: Jean-Marc Desharnais

Fit Criterion: Modify its name, description and relationship to a keyword.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Conflicts:

Supporting Materials:

Requirement #: 30

Requirement Type: E_Manage

Use case #: 13

Description: Add a Case Problem

Rationale: Knowledge acquisition

Source: Jean-Marc Desharnais

Fit Criterion: Successfully add a Case Problem (name, description, relationship to Topological Concept, path for reference file)

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 31

Requirement Type: E_Manage

Use case #: 13

Description: Delete a Case Problem

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Delete a Case Problem with all its attributes and related Themes and Recommendations.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 32

Requirement Type: E_Manage

Use case #: 13

Description: Modify a Case Problem

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Modify its name, description and relationship to a Topological Concept.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 33

Requirement Type: E_Manage

Use case #: 14

Description: Add a Theme

Rationale: Knowledge acquisition

Source: Jean-Marc Desharnais

Fit Criterion: Successfully add a Theme (name, description, relationship to Case Problem, path for reference file)

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 34

Requirement Type: E_Manage

Use case #:14

Description: Delete a Theme

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Delete a Theme with all its attributes and if related Facts.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 35

Requirement Type: E_Manage

Use case #: 14

Description: Modify a Theme

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Modify its name, description and relationship to a Case Problem.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 36

Requirement Type: E_Manage

Use case #: 16

Description: Add a Fact

Rationale: Knowledge acquisition

Source: Jean-Marc Desharnais

Fit Criterion: Successfully add a Fact (name, description, relationship to The me, path to reference file)

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Conflicts:

Supporting Materials:

Requirement #: 37

Requirement Type: E_Manage

Use case #: 16

Description: Delete a Fact

Rationale: Knowledge maintenance
Source: Jean-Marc Desharnais
 Fit Criterion: Delete a Fact with all its attributes in all languages
 Customer Satisfaction: 5
 Customer Dissatisfaction: 5
 Dependencies:

Requirement #: 38

Requirement Type: E_Manage Use case #: 16
Description: Modify a Fact
 Rationale: Knowledge acquisition
Source: Jean-Marc Desharnais
 Fit Criterion: Modify its name, description and relationship to a Theme.
 Customer Satisfaction: 5
 Customer Dissatisfaction: 5
 Dependencies:

Requirement #: 39

Requirement Type: E_Manage Use case #: 15
Description: Add Recommendation
 Rationale: Knowledge acquisition
 Source: Jean-Marc Desharnais
 Fit Criterion: Successfully add a Recommendation (name, description, relationship to Case Problem, path to reference file)
 Customer Satisfaction: 5
 Customer Dissatisfaction: 5
 Dependencies:

Requirement #: 40

Requirement Type: E_Manage

Use case #: 15

Description: Delete Recommendation

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Delete a Recommendation with all its attributes in all languages.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 41

Requirement Type: E_Manage

Use case #: 15

Description: Modify a Recommendation

Rationale: Knowledge maintenance

Source: Jean-Marc Desharnais

Fit Criterion: Modify its name, description and relationship to a Case Problem.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Conflicts:

Supporting Materials:

Requirement #: 42

Requirement Type: E_Manage

Use case #: 17

Description: Select a diagnostic model**Rationale:** The expert has to be able to choose from three different diagnostic model

which stand for different ways of reasoning recommendations.

Source: Jean-Marc Desharnais

Fit Criterion: Choose a diagnostic model on a form.

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Requirement #: 43

Requirement Type: E_Manage

Use case #: 20

Description: Change the caption of the user interfaces (forms, controls)

Rationale: The expert shall be able to adjust the user-interface to his needs/domain

Source: Jean-Marc Desharnais

Fit Criterion: Edit caption of buttons, forms, menus and their items, labels, text boxes and datagrid headers.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 44

Requirement Type: E_Manage

Use case #: 18

Description: Produce an Excel sheet with “measurer follow up” information

Rationale: For statistic reasons.

Source: Jean-Marc Desharnais

Fit Criterion: Measurer Follow Up information in Excel format

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 45

Requirement Type: E_Manage

Use case #: 18

Description: Produce an Excel sheet with “Session” information

Rationale: For statistic reasons

Source: Jean-Marc Desharnais

Fit Criterion: Session information in Excel format.

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

Requirement #: 46

Requirement Type: E_Manage

Use case #: 18

Description: Delete Measurer Follow Up report either by days older than x or completely.

Rationale: Avoiding huge tables with information which is no longer necessary.

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: Deleted rows in class “measurer follow up”

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Conflicts:

Supporting Materials:

Requirement #: 47

Requirement Type: E_Manage

Use case #: 18

Description: Delete session report either by days older than x or completely.

Rationale: Avoiding huge tables with information which is no longer necessary

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: Deleted rows in class "session"

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 48

Requirement Type: E_Manage

Use case #: 21

Description: The expert has to have all the functionality of a measurer

Rationale:

Source: Jean-Marc Desharnais

Fit Criterion:

Customer Satisfaction:

Customer Dissatisfaction:

Dependencies:

Conflicts:

Supporting Materials:

Requirement #: 49

Requirement Type: E_MaintainKnowledge

Use case #: 11-16

Description: Modify a language-set⁶ of Keywords, TCs, CPs, Themes, Recommendations and Facts

Rationale: A keyword will be added in one language, that means an expert must have the opportunity to translate his new entered keyword into other languages. That will result in a multiple language knowledge (Case) base.

Source: Tim Kuessing

Fit Criterion: Modifying keywords in different languages

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Dependencies:

⁶ A "language-set" is one keyword (or TC, CP, Theme, Fact, Recommendation) presented in its four languages

Requirement #: 50

Requirement Type: E_MaintainKnowledge

Use case #: 22

Description: Drag and drop Case Problems, Themes, Facts or Recommendations

Rationale: The same Case Problems (or others) might be necessary within two different Topological Concepts and to save time a drag & drop function would be good.

Source: Jean-Marc Desharnais

Fit Criterion: Drag and drop the highlighted item and the hierarchel levels underneath.

Possible drop items have to marked different from impossible ones. For example a Fact can only be dropped on a Theme.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 51

Requirement Type: E_MaintainKnowledge

Use case #: 22

Description: Copy a Case Problem, Theme, Fact or Recommendation

Rationale: The same Case Problems (or others) might be necessary within two different Topological Concepts and to save time a copy function would be good.

Source: Jean-Marc Desharnais

Fit Criterion: When copying the hierarchical level underneath have to be copied as well

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 52

Requirement Type: E_MaintainKnowledge

Use case #: 22

Description: Cut a Case Problem, Theme, Fact or Recommendation

Rationale: A Case Problem (or other) can become obsolete for one Topological Concept, but might be interesting to consider in an other. For that case a cut function would be good.

Source: Jean-Marc Desharnais

Fit Criterion: Cut an item with its layers underneath

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 53

Requirement Type: E_MaintainKnowledge Use case #: 22

Description: Paste a Case Problem, Theme, Fact or Recommendation

Rationale: To complete the copy and cut process.

Source: Jean-Marc Desharnais

Fit Criterion: A copied or cutted item shall be pasted into its new position

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 54

Requirement Type: E_MaintainKnowledge Use case #: 11-16, 22

Description: Query Properties of Keyword, Topological Concept, CP, Theme, Fact, Recommendation

Rationale: Presentation of knowledge base

Source: Jean-Marc Desharnais

Fit Criterion: Displaying items and its properties and links to sub items

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 55

Requirement Type: E_MaintainKnowledge Use case #: 22

Description: Show description of item in tree view in text box

Rationale: Expert information

Source: Jean-Marc Desharnais

Fit Criterion: When selecting an item of the tree view, its description has to show up a text box.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Dependencies:

Requirement #: 56

Requirement Type: E_Report

Use case #: 18

Description: A list presenting the relationship between all Keywords and their Topological Concepts

Rationale: Gives the expert an overview of the existing relationships between keywords and Topological Concept. It'll help him to maintain the case specific data.

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: By clicking on a specific-button in the menu a list shall come up with the three columns: Keyword, Topological Concept and the relationship between both.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 57

Requirement Type: E_Report

Use case #: 18

Description: A list presenting the relationship between all TCs and their Case Problems

Rationale: Gives the expert an overview of the existing relationships between TCs and Case Problems. It'll help him to maintain the case specific data.

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: By clicking on a specific-button in the menu a list shall come up with the three columns: TC, Case Problem and the relationship between both.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 58

Requirement Type: E_Report

Use case #: 18

Description: A list presenting the relationship between all Case Problems and their Themes

Rationale: Gives the expert an overview of the existing relationships between CPs and Themes. It'll help him to maintain the case specific data.

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: By clicking on a specific-button in the menu a list shall come up with the three columns Case Problem, Theme and the relationship between both.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 59

Requirement Type: E_Report

Use case #: 18

Description: A list presenting the relationship between all CP and their Recommendations

Rationale: Gives the expert an overview of the existing relationships between CP and Recommendations. It'll help him to maintain the case specific data.

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: By clicking on a specific-button in the menu a list shall come up with the three columns: CP, Recommendations and the relationship between both.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 60

Requirement Type: E_Report

Use case #: 18

Description: A list presenting the relationship between all Themes and their Facts

Rationale: Gives the expert an overview of the existing relationships between Themes and Facts. It'll help him to maintain the case specific data.

Source: Jean-Marc Desharnais, Tim Kuessing

Fit Criterion: By clicking on a specific-button in the menu a list shall come up with the three columns: Theme, Fact and the relationship between both.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Administrator

Requirement #: 61

Requirement Type: E_Manage

Use case #: 31

Description: Manage⁷ the users

Rationale: Only the administrator is allowed to add, delete a user or grant certain access to the system.

Source: Jean-Marc Desharnais

Fit Criterion: The edit user interface must be hidden for experts and measurers

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

Requirement #: 62

Requirement Type: E_Manage

Use case #: 21/22

Description: An Administrator has to have all the functionality of an expert.

Rationale: Only the administrator is allowed to add, delete a user or grant certain access to the system.

Source: Jean-Marc Desharnais

Fit Criterion: The edit user interface must be hidden for experts and measurers

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Dependencies:

⁷ "Manage" means to add, modify and delete

Data requirements.

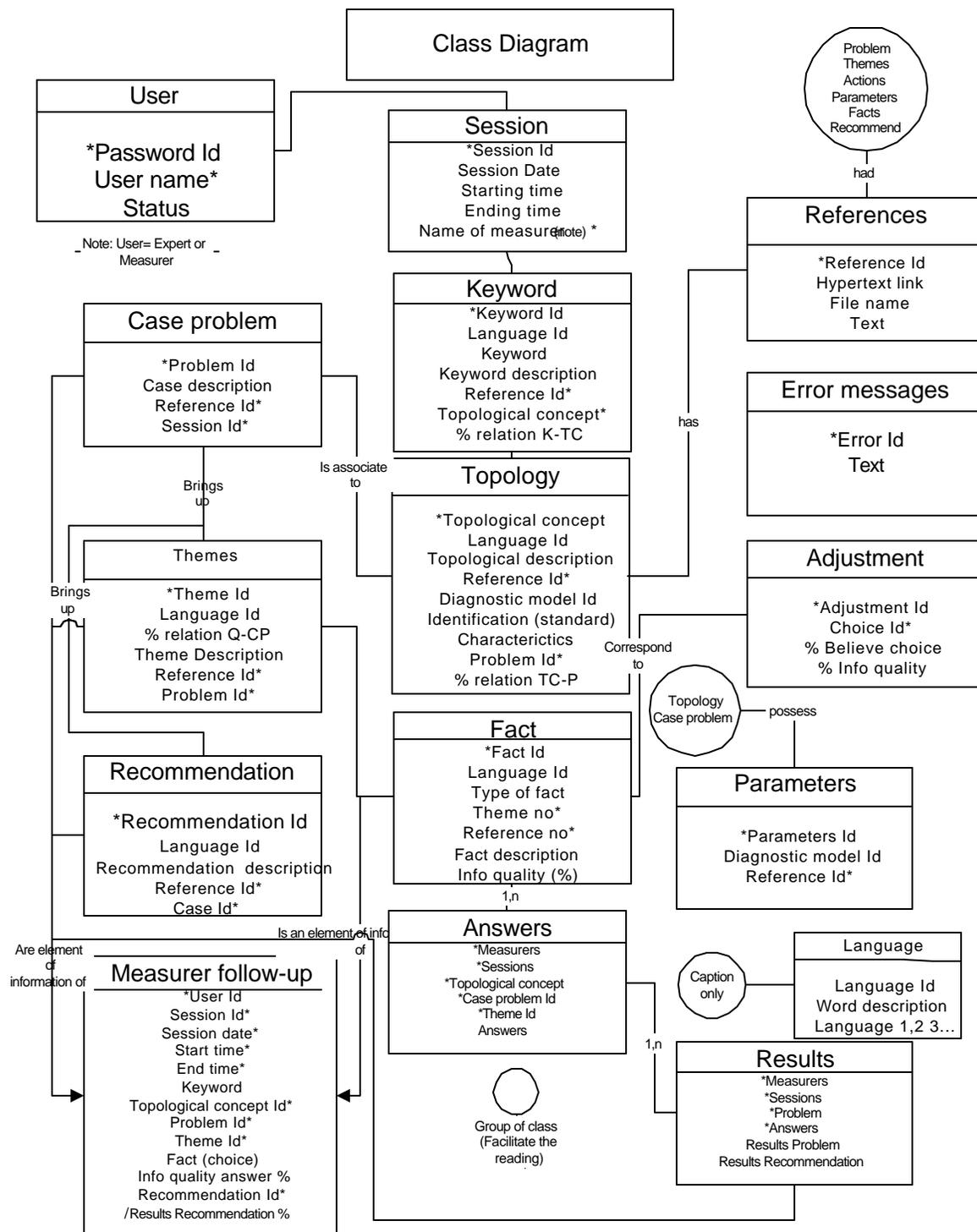


Figure 2: Data model

L'anglais du texte qui suit a été corrigé partout où il était possible de le faire. Cependant, pour les "copies" en format image des différents écrans du logiciel ce ne fut pas possible.

Annexe B: partie 2
 USER GUIDE FOR COSMICXPRT

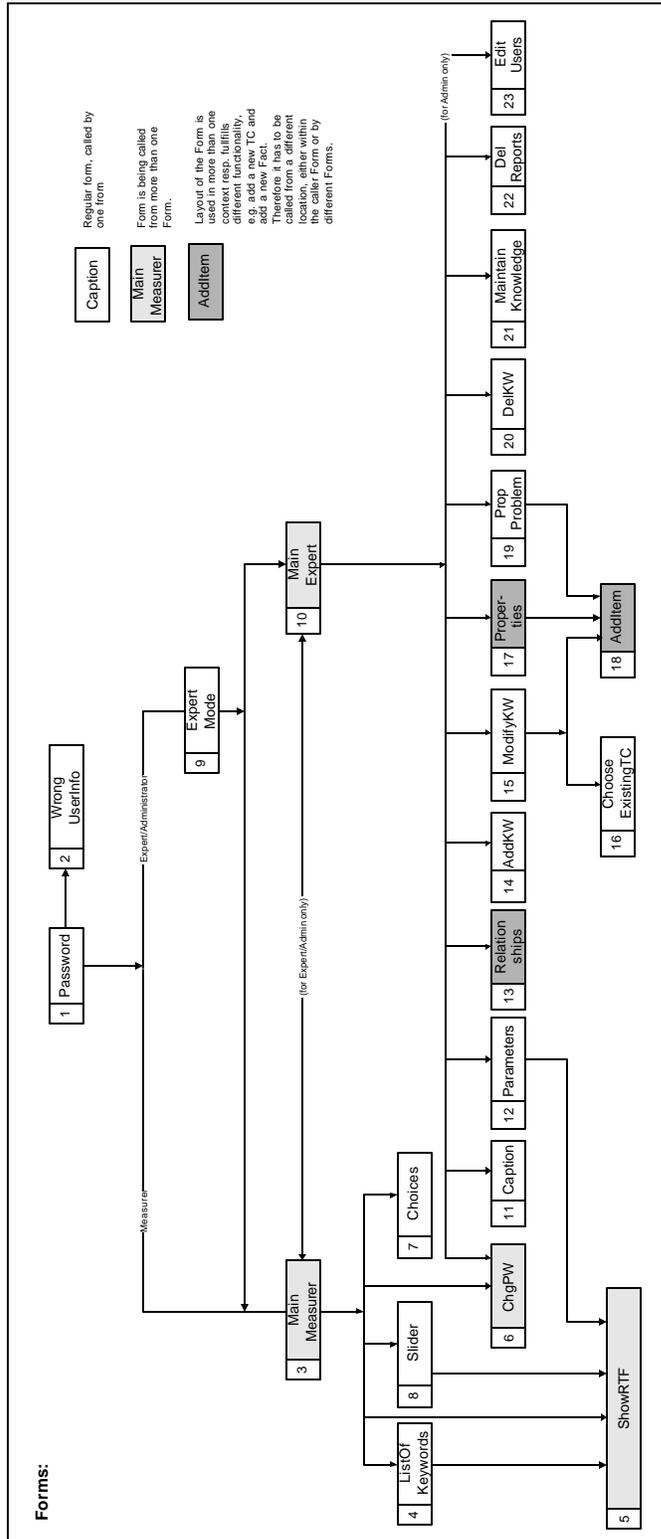


Figure 1: Menu

This user guide provides an overview of COSMICXPert, explaining its forms and their functions. The chapter is subdivided into sections subtitled Start Forms, Measurer Interface, Expert Mode and Administrator Mode. Note that the numbers in Figure 39 correspond to the numbers in the Table of Contents for forms on the next page.

List of Forms

Start Forms

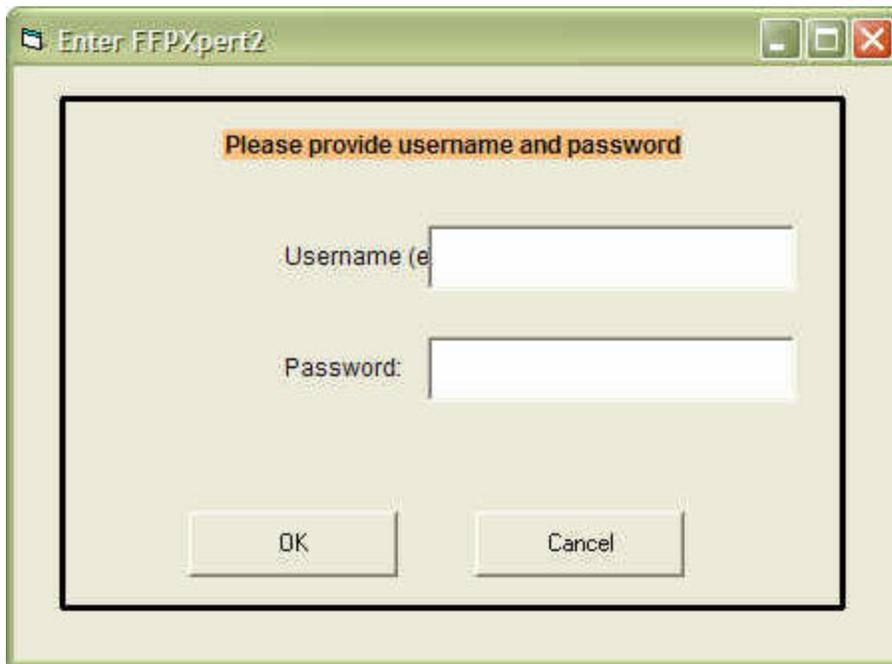
The image shows a standard Windows-style dialog box titled "Enter FFPXpert2". The dialog has a light beige background and a dark border. At the top, there is a title bar with a small icon on the left and three control buttons (minimize, maximize, and close) on the right. The main area of the dialog contains a red-bordered box with the text "Please provide username and password" in a bold, black font. Below this message, there are two text input fields. The first is labeled "Username (e" and the second is labeled "Password:". At the bottom of the dialog, there are two buttons: "OK" on the left and "Cancel" on the right.

Figure 2: Entering COSMICXPert

To enter the system, username and password are provided by the user. If they are valid, access is granted to the tools' functionality in accordance with the status of the user. A measurer, for example, sees the screen in Figure 2 and can solve a measurement problem, while an expert or administrator sees the screen in Figure 11, called Expert Mode, and can either maintain the system or solve a problem.

If the expiry date has passed, the user will be informed to contact the owner of the product.

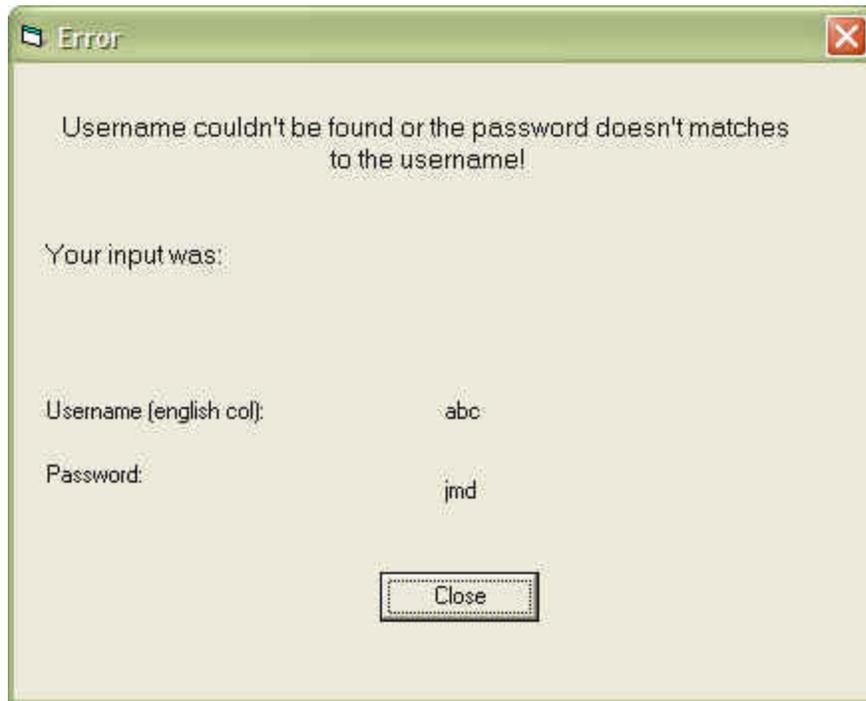
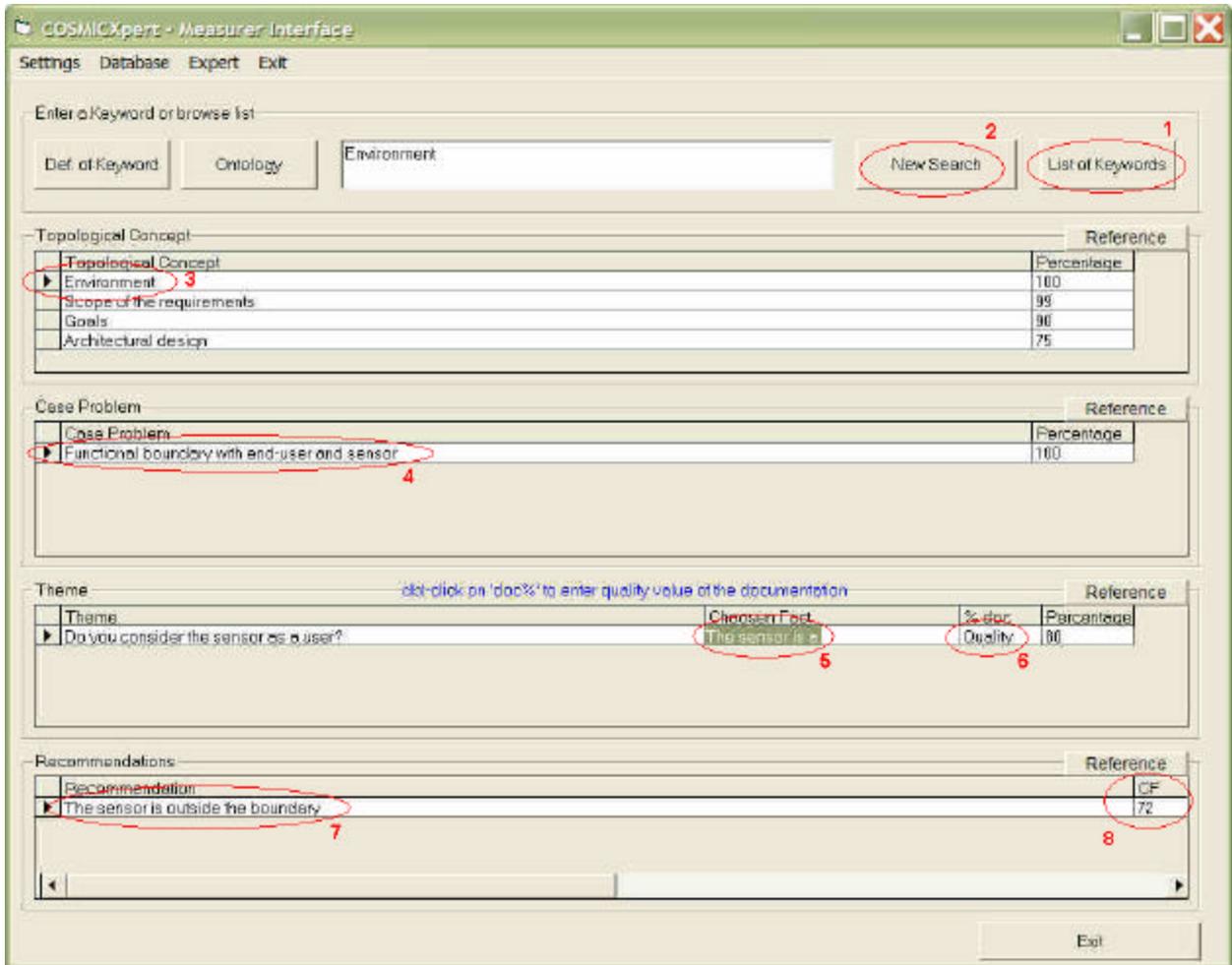


Figure 3: Wrong Password

This screen (Figure 3) comes up if the user entered the wrong password in Figure 2, and gives the username and password entered. The Close button brings the user back to the screen in Figure 2 so that the necessary correction(s) can be made.



Measurer Interface

Figure 4: Measurer Interface for solving a problem

This is the screen the measurer uses to solve a measurement problem. As described in Figure 4, the user can either enter a keyword or browse through a list of keywords by pressing the List of Keywords button, marked (1) in the Measurer Interface. After selecting a keyword and pressing the New Search button (2), the expert system displays

the affected TC, CP and Themes. The TC (3) or the CP (4) can now be deleted by marking a row on the data grid and pressing Delete on the user's keyboard. CPs and Themes are automatically updated in response to the new situation.

The next step is to answer a theme by choosing a Fact. To do this, the user double-clicks on the Chosen Fact field (5), at which point a window with multiple Facts appears. Double-clicking on the Quality field (6) provides the means for entering the measurer's confidence level in the selected Fact. This confidence level essentially depends on the quality of the measurer's documentation.

After answering a Theme or changing the quality rating of the documentation, recommendations are recalculated and displayed (7) on the Recommendations data grid. CF is the computed value and stands for "certainty factor" (8).

Percentage on the TC data grids indicates the extent of the link between a specific TC and the Keyword selected in the text box above. The same principle applies for the percentages on the other data grids, e.g. the percentage attached to a CP expresses the relationship between this CP and the TC related to it.

A right-click on a row of a data grid produces a pop-up menu giving an explanation of the item.

On the Measurer Interface, there are three more buttons, labeled Def. of Keyword, Ontology and Reference. Clicking on these produces a rich text file explaining and defining specific vocabulary.

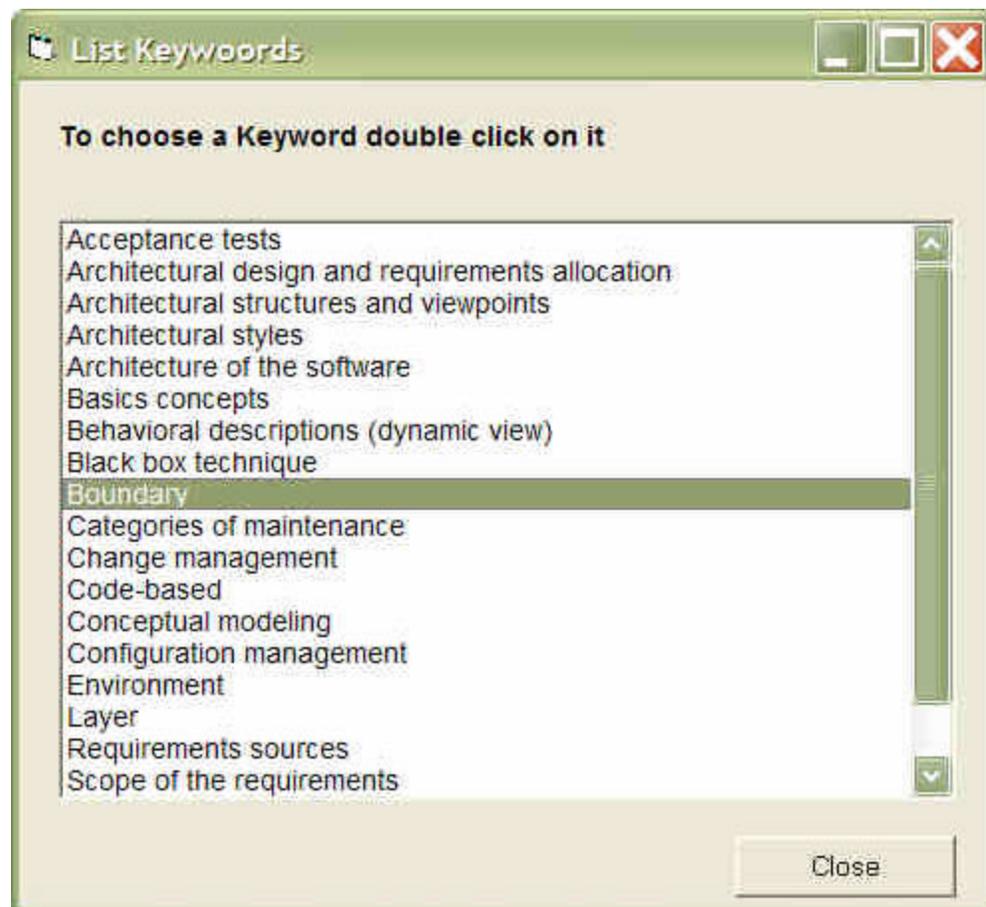


Figure 5: List of Keywords

This screen (Figure 5) comes up when the List of Keywords button has been pressed on the Measurer Interface. It gives the user an overview of the Keywords contained in the Knowledge Base. Keywords are selected by simply double-clicking on the desired one, or highlighting it, and hitting Close. It is also possible to ask for a description by right-clicking on a highlighted item.

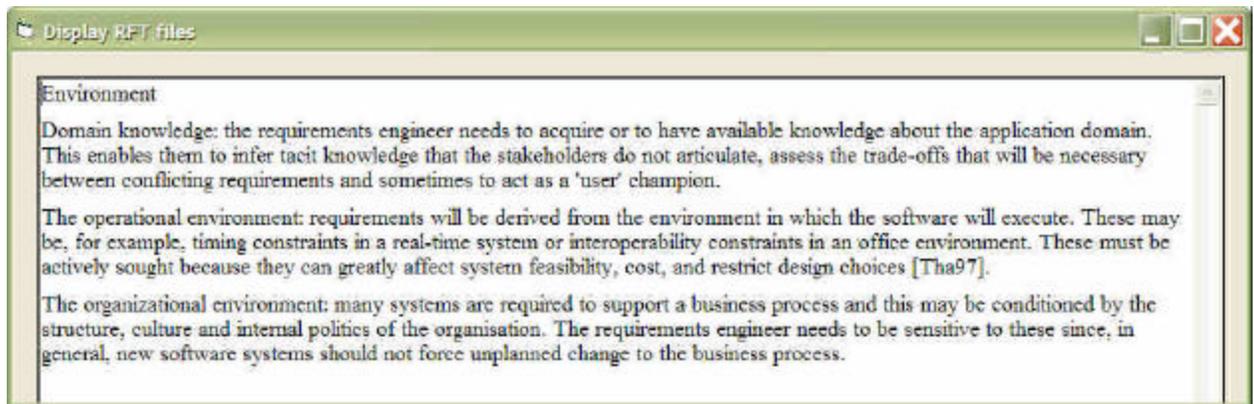


Figure 6: A rich text file (top part as is, no correction)

When the user asks for a long reference or an explanation, this window, which can display rich text files, comes up (Figure 6). User actions are limited to scrolling through the document or closing it.

The image shows a dialog box titled "Password Change (measurer)". It is divided into two main sections. The first section, "Change Your Personal Settings", contains three input fields: "Username (english col):" with the text "jmd", "Enter new password:", and "Confirm new password:". The second section, "Change to your desired language", contains a dropdown menu labeled "Choose default language" with "ENGLISH" selected. At the bottom of the dialog are "OK" and "Cancel" buttons.

Figure 7: User settings

This screen is designed to manage the username, password and desired language of the individual currently logged in. It can be reached from the Measurer Interface by clicking on Settings and the sub-item User on the menu bar.

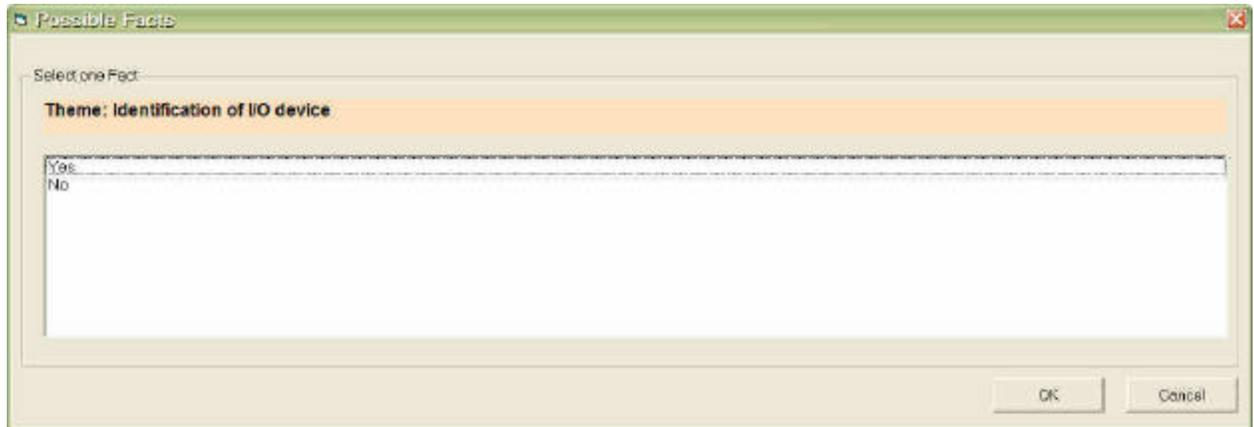


Figure 8: Making a choice

This Possible Facts screen (Figure 8) appears when the user selects a Theme to answer on the screen in Figure 5 and double-clicks on it. The selected Theme to be answered is repeated, the list box underneath it containing multiple valid Facts). As in Windows™, the user can either highlight a choice and hit OK or just double-click on the desired Fact. The screen then closes and the user can see the Fact right next to the Theme in the Chosen Fact column on Form 3.

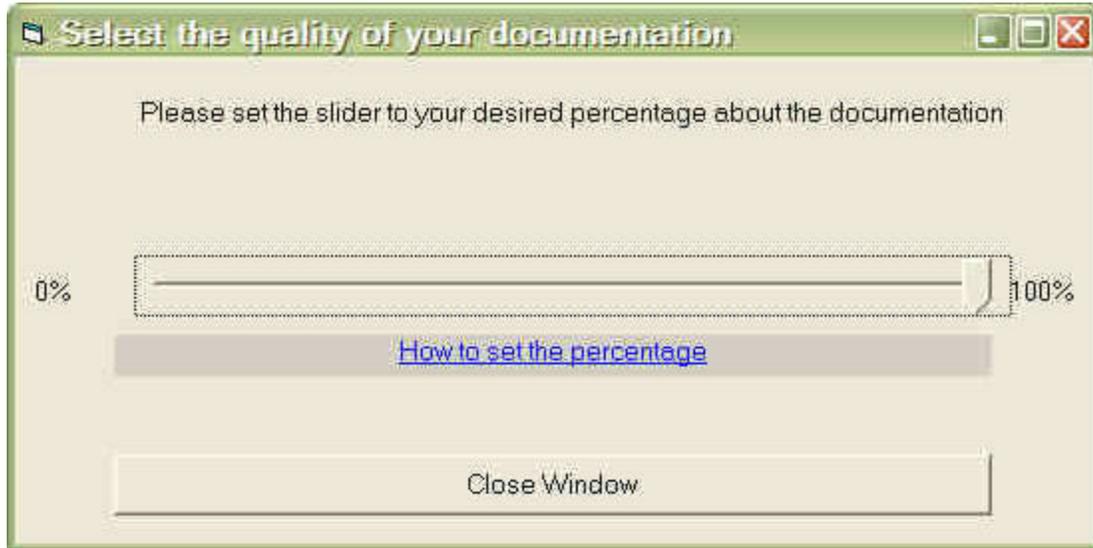


Figure 9: Entering the quality of the documentation

The text should be read: **“Please set the slider to reflect your assessment of documentation quality”**

The quality of the documentation for a particular Theme can be specified by double-clicking in the % doc column of Quality. The input here depends on the Theme selected. Figure 9 then pops up and offers the chance to enter the quality of the documentation in a range between 0% and 100%. Underneath the slider, the hyperlink “How to set the percentage” can be found, indicating suitable ranges for well documented, good, etc. As usual, the Close button saves the setting to the database and closes the window so that the measurer can continue “working” on the problem at hand.

Expert Mode

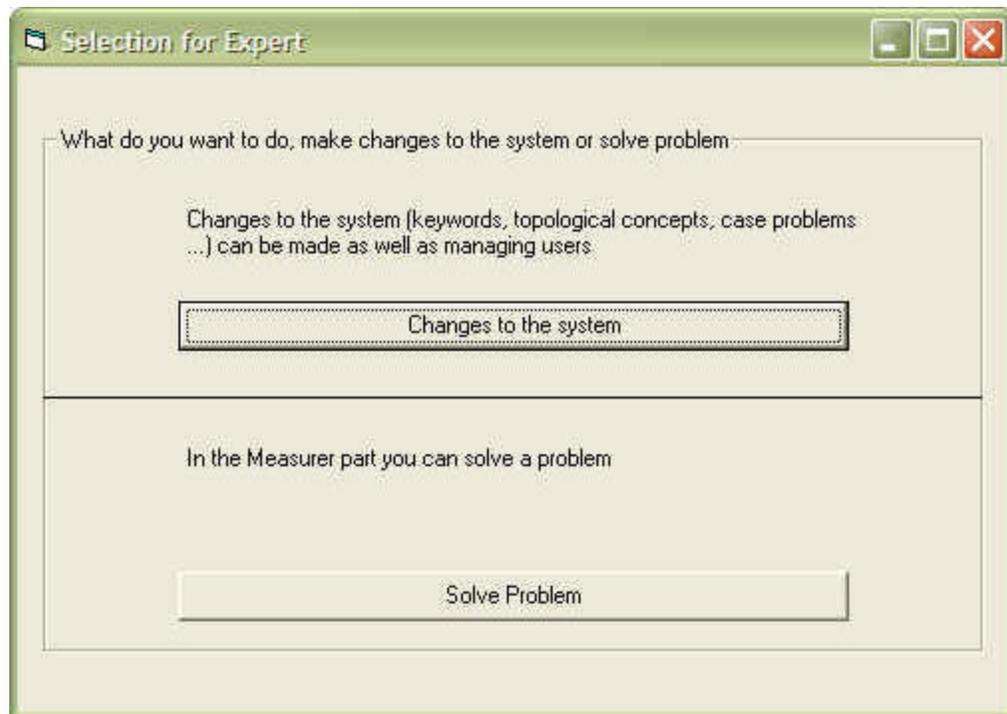


Figure 10: Expert Mode

This is a very basic screen. The expert has access to the same functionality as the measurer, so it seemed convenient to provide a screen through which the expert can choose the scope of his work: “maintaining the system” or “solving a measurement problem.” The expert makes his selection, and is guided by the program towards either solving a problem or maintaining the system.

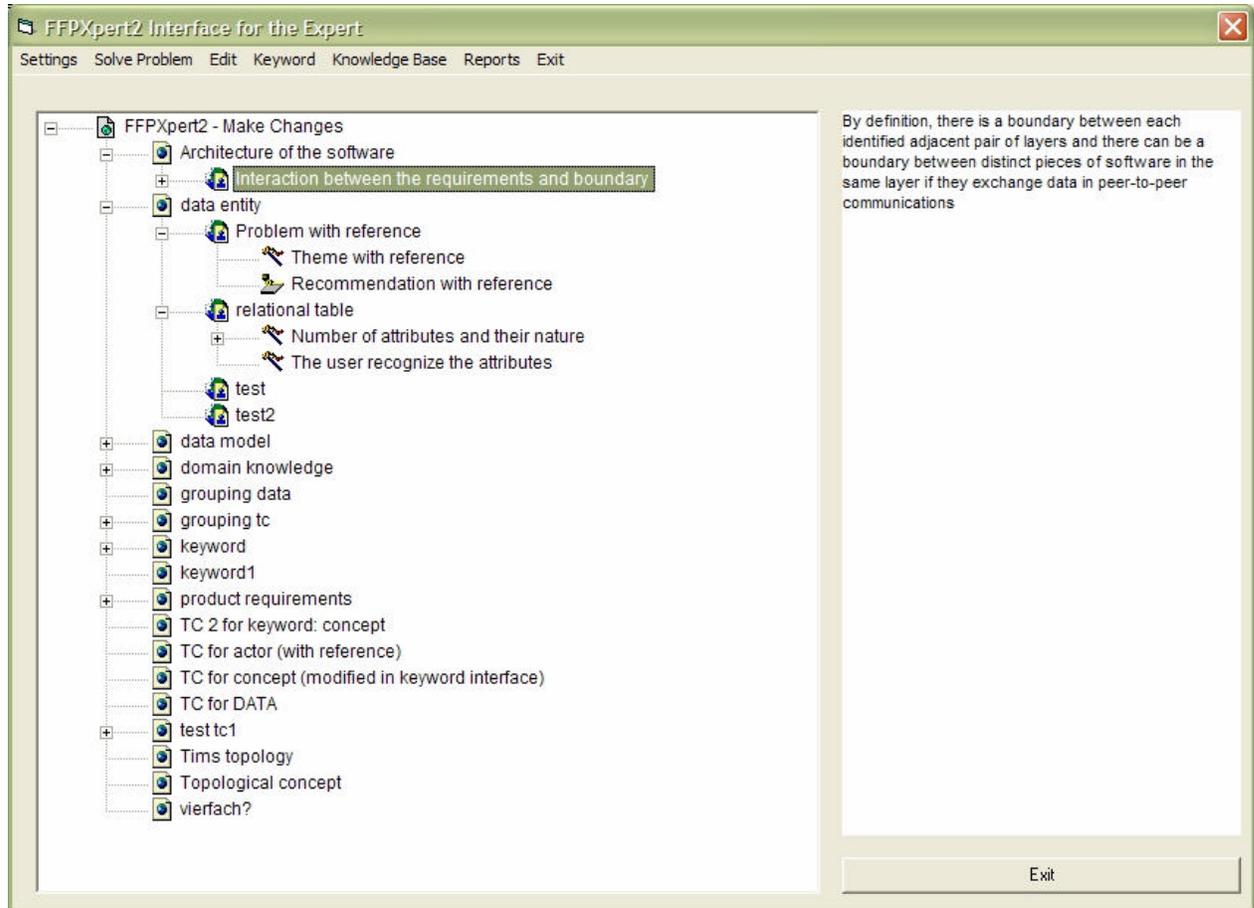


Figure 11: Expert Interface for managing the Knowledge Base

This is the main screen from which the expert operates. The left field is called a Tree view and visualizes the structure of the case-specific knowledge. Underneath the starting point of the Tree view, the root item, there are Topological Concepts, followed by related Case Problems. At the next level can be seen the Themes and Recommendations to which the Case Problem or interest is related. At the fifth level, under Themes, are the possible Facts available for answering a Theme. To see how strong the relationship is between the nodes, the highlighted node can be right-clicked and properties from the pop-up menu selected. This guides the user to screens which provide further details and the option of making adjustments.

When the user selects a node in the Tree view, its description appears in the text box to the right.

Concerning functionality in this window, it is possible to drag and drop nodes, but, unlike Windows Explorer™, COSMICXPERT requires that nodes have to be dropped at the right place, otherwise an error message appears.

The “right place” means:

Facts can only be dropped on Themes,

...Themes on Case Problems,

...Recommendations on Case Problems, and

...Case Problems on Themes;

Topological Concepts have to be bound to a Keyword through the interface function Modify Keyword because of their n:m relationship.

The user can also copy or cut and paste nodes, either by employing Edit on the menu bar or by using the Windows-like shortcuts Ctrl+C, Ctrl+X and Ctrl+V. Of course, nodes can be deleted by hitting Delete on the keyboard or using Delete in the Edit menu.

The Settings option on the menu bar is the home of links to screens where personal settings like username, password and language can be modified and the captions of buttons, labels, frames, data grids, etc. can be adapted (Form 11). The user can also choose from three diagnostic models (**Error! Reference source not found.**).

The next item on the menu bar, Solve a Problem, opens the Measurer Interface.

The Keyword menu is needed to add (**Error! Reference source not found.**), modify (Form 15) or delete (Form 20) keywords which cannot be displayed in the Tree view due to their n:m relationship to Topological Concepts. Tree views can only display 1:n relationships.

The Maintain Knowledge option will lead the expert to an interface (Form 21) where case-specific knowledge can be translated into other languages.

On the Reports menu, reports displaying the relationships among all existing Keywords and Topological Concepts, Topological Concepts and Case Problems, Case Problems and Themes, etc. (Form 13) are differentiated from reports created for statistical reasons.

There are two statistical reports (Excel™ format), one providing information about who has entered the system and when, and the other being a “log-file” of the problem-solving process carried out by a measurer answering Themes and providing information about the quality of documentation.

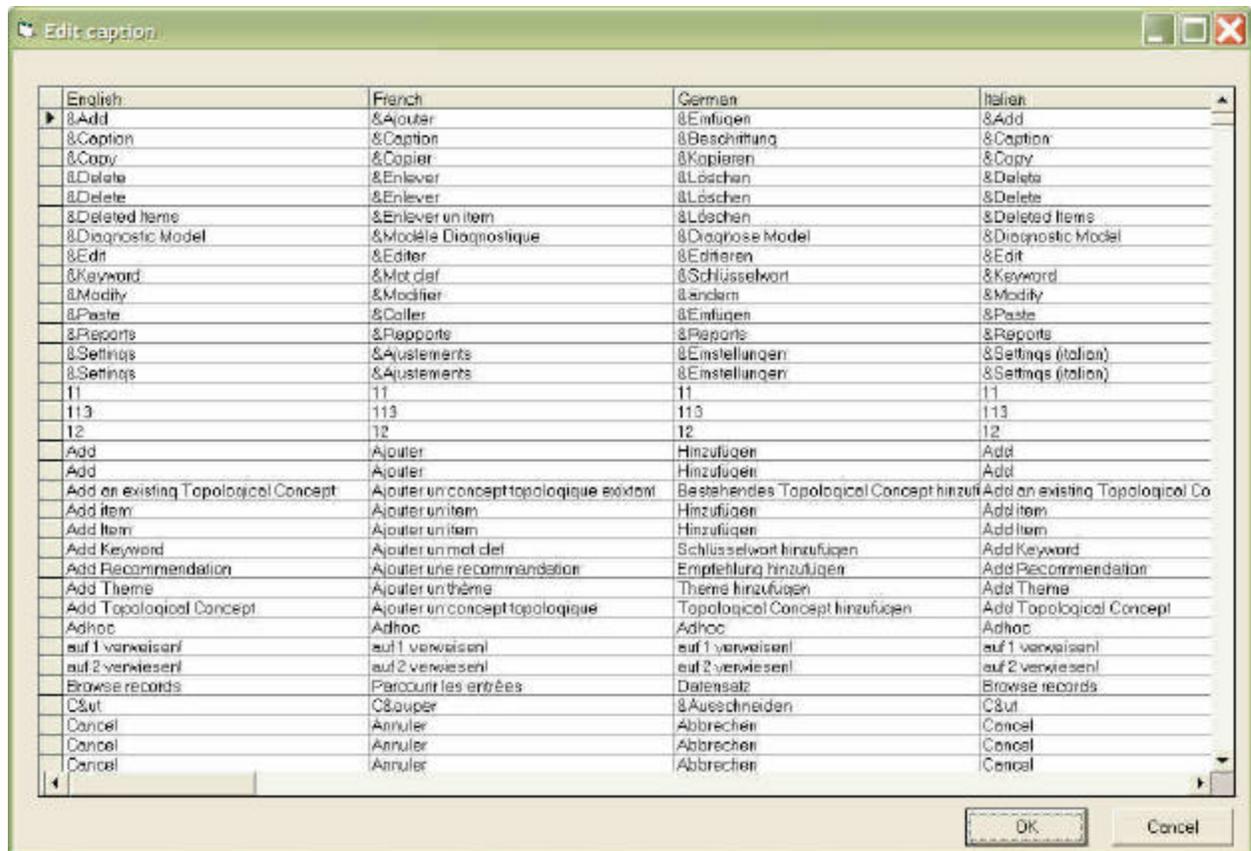
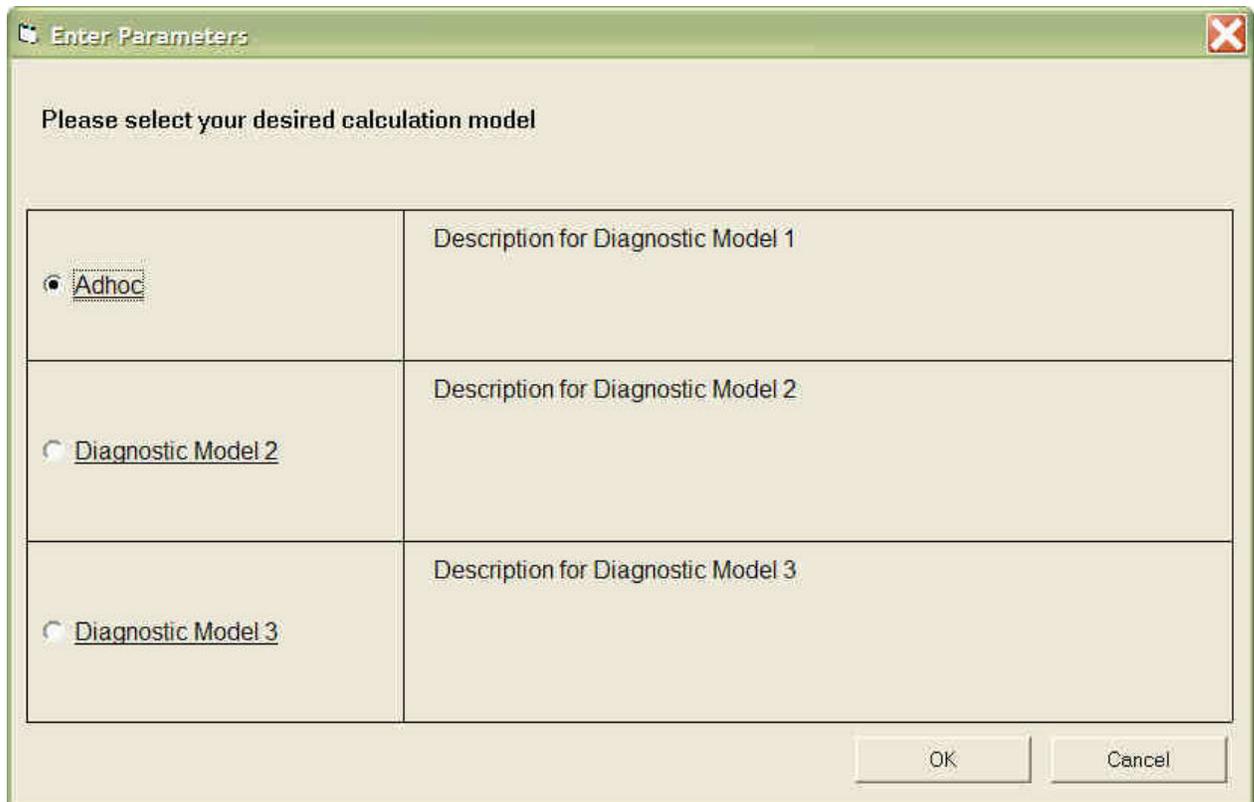


Figure 12: Editing captions

The other language issue involves changing the captions of buttons, menu bars, labels, etc. to match the selected language. The caption itself can be modified on this form, and is accessible through the menu bar (Settings, Caption) for the Expert Interface. There are four columns, one for each of the languages offered. Scrolling further to the right reveals the attribution of the control. Every control is part of a Form, and the location description is unique.



Enter Parameters

Please select your desired calculation model

<input checked="" type="radio"/> <u>Adhoc</u>	Description for Diagnostic Model 1
<input type="radio"/> <u>Diagnostic Model 2</u>	Description for Diagnostic Model 2
<input type="radio"/> <u>Diagnostic Model 3</u>	Description for Diagnostic Model 3

OK Cancel

Figure 13: Enter parameters to select a diagnostic model

In future, it should be possible to select different diagnostic models for computing appropriate Recommendations. However, at the time of development, only the certainty theory had been prepared for implementation. With this screen, the expert can select the desired model, but as yet this choice cannot be implemented.

Relationships between Case Problems and Themes

Percentages and names can be modified

Case Problem	Theme	CP-Theme %
Functional	Do you consider the sensor as a user?	80
Clock signal - RC	The clock signal initiates one or more functional processes.	90
Clock signal - RC	The clock signal initiates functional processes belonging to another layer.	70
Triggering event user press a switch - RC	The switch initiates one or more functional processes.	90
Triggering event user press a switch - RC	The switch initiates an internal process.	30
Cooking Mode - RC	Identification of a data attribute	90
Cooking Mode - RC	More than one instantiation of the data attribute	90
Cooking Mode - RC	Persistence of the data	85
Cooking Mode - RC	Code table	60
Status - RC	More than one instantiation of the data attribute	90
Temperature - RC	More than one instantiation of the data attribute	90
X seconds clock signal - RC	More than one instantiation of the data attribute	90
Status - RC	Persistence of the data	85
Temperature - RC	Persistence of the data	85
X seconds clock signal - RC	Persistence of the data	85
Functional boundary with user and sensor - RC	Data maintain by a tool	100
Status - RC	Code table	75
Status - RC	Identification of a data attribute	100
Temperature - RC	Code table	60
Temperature - RC	Identification of a data attribute	90
X seconds clock signal - RC	Code table	60
X seconds clock signal - RC	Identification of a data attribute	90
Control heater - RC	It operate on a unique and ordered set of data movements performing a set of	90
Control heater - RC	It's triggered by an event (triggering event)	90
Control heater - RC	The triggering event occur outside the boundary of the software	90
Mode switch - RC	Receives data attributes lying outside the software boundary	90
Mode switch - RC	Receives data from only one data group	90
Mode switch - RC	The sub-process does not exit, read or write data.	90
Receive triggering event - RC	Receives data attributes lying outside the software boundary	90
Receive triggering event - RC	Receives data from only one data group	90
Receive triggering event - RC	The sub-process does not exit, read or write data.	90
Received elapsed time - RC	Receives data attributes lying outside the software boundary	90
Received elapsed time - RC	Receives data from only one data group	90

Close

Figure 14: Relationships between case-specific data

To maintain case-specific knowledge, it is very useful to have a screen showing already existing Case Problems and the Themes to which they correspond, with an indication of how strongly they are linked. Of course, it is important to see the whole expert knowledge relationship. Therefore, access is given through the Reports menu, where links can be found displaying the affinity between:

- Keyword – Topological Concept
- Topological Concept – Case Problem
- Case Problem – Theme
- Theme – Fact
- Case Problem – Recommendation

Add a Keyword

Enter new Keyword attributes:

Enter new Keyword

Enter description for new keyword

Enter path of text file: ...

Check for existing keywords to avoid double entries

Add a Keyword

Add Keyword Close

Figure 15: Adding a Keyword

The new keyword is entered on this simple form, along with its description and name (including extension). As an additional feature, the user can browse through a list of keywords to avoid similar entries. Of course, the system does not allow the same keyword to be entered more than once.

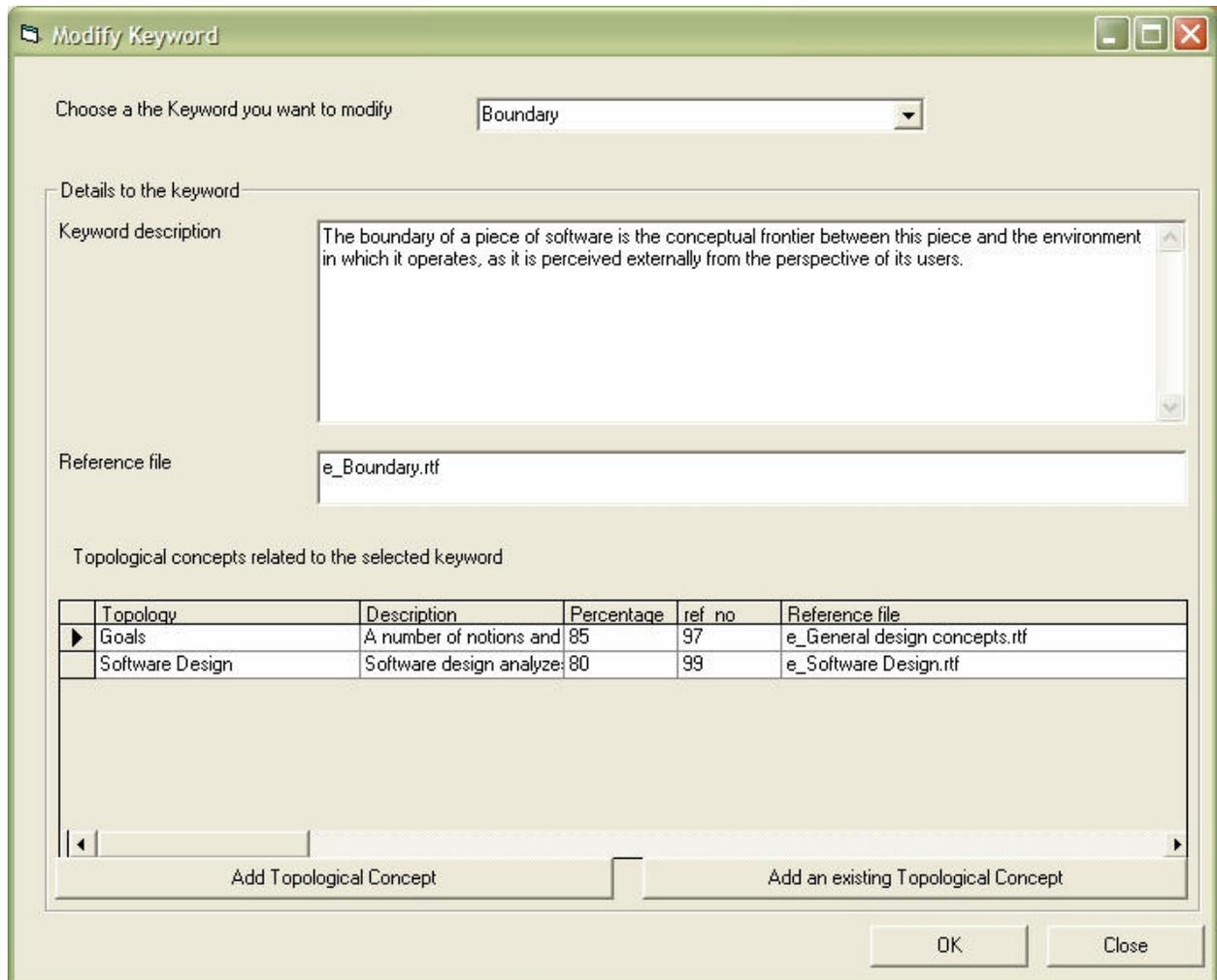


Figure 16: Modifying Keywords

Here, the expert can modify a keyword's description, reference file name or related TCs. The keyword itself cannot be modified, only added or deleted, in order to avoid confusion. There are two buttons to link a TC with the selected keyword. The user can either choose from a list of existing keywords or add a completely new TC.

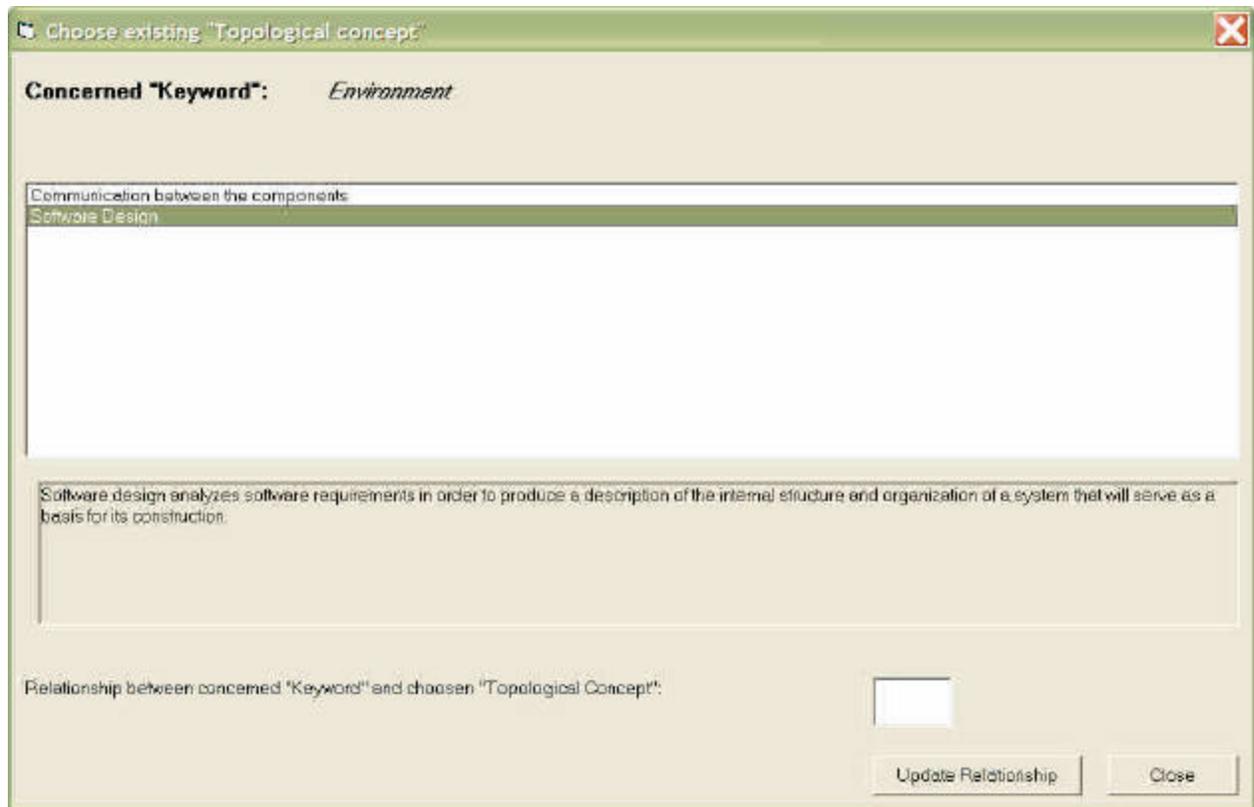


Figure 17: Adding an existing TC to a Keyword

The first line shows the Keyword to which the TC will be added, and the list box contains the available TCs. To add a TC, the user simply highlights that TC and supplies the relationship percentage. The description, reference file and links to related CPs, Themes, Facts and Recommendations are copied as well.

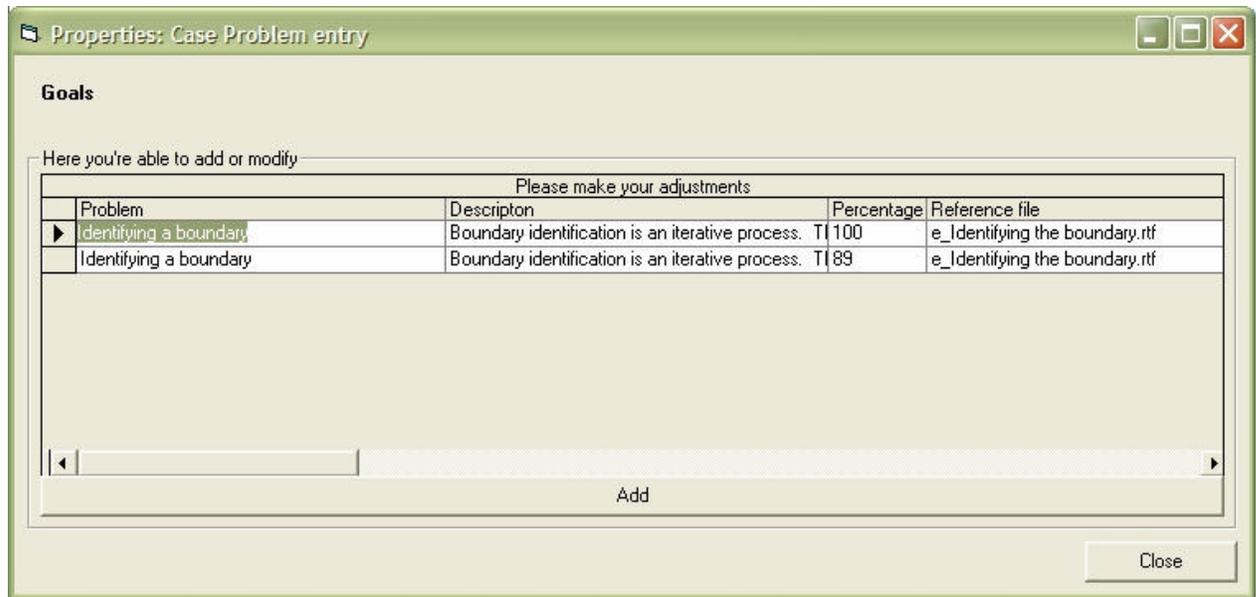


Figure 18: Properties of a Topological Concept

The title within Figure 18 and 19 should be: **“Here is where you can add or modify”**
 To get to this window, the user selects a TC and right-clicks on it in Tree view. A menu pops up from which the Properties item can be chosen (also accessible through the menu bar). In the first line, the TC is displayed, with the related CP in the table underneath. All CP properties can be modified here, or a new problem added by pressing the Add button. To add an existing CP to a TC, the drag and drop function in the Tree view of the Expert Interface should be used.

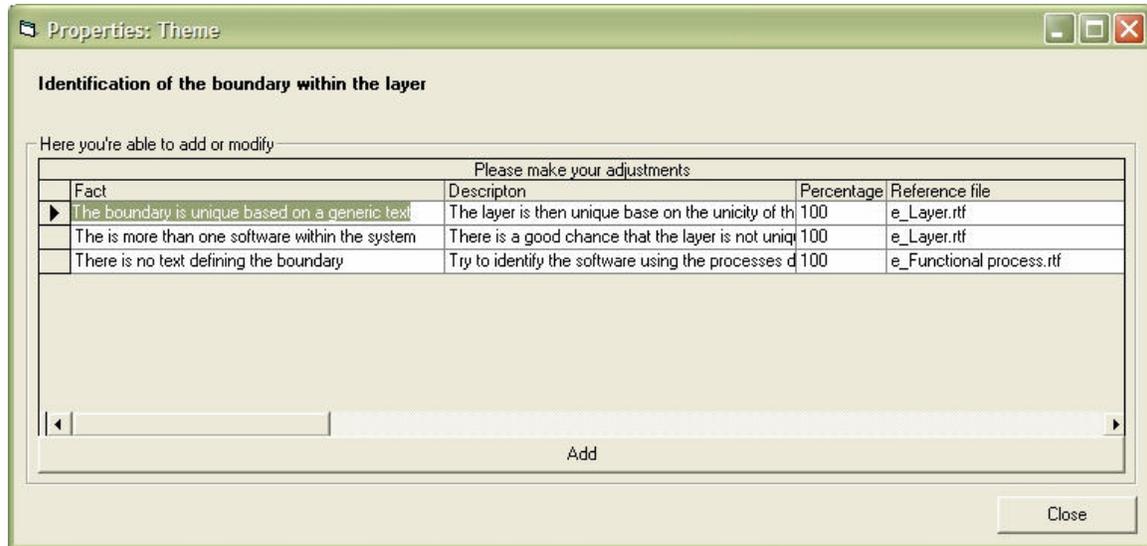
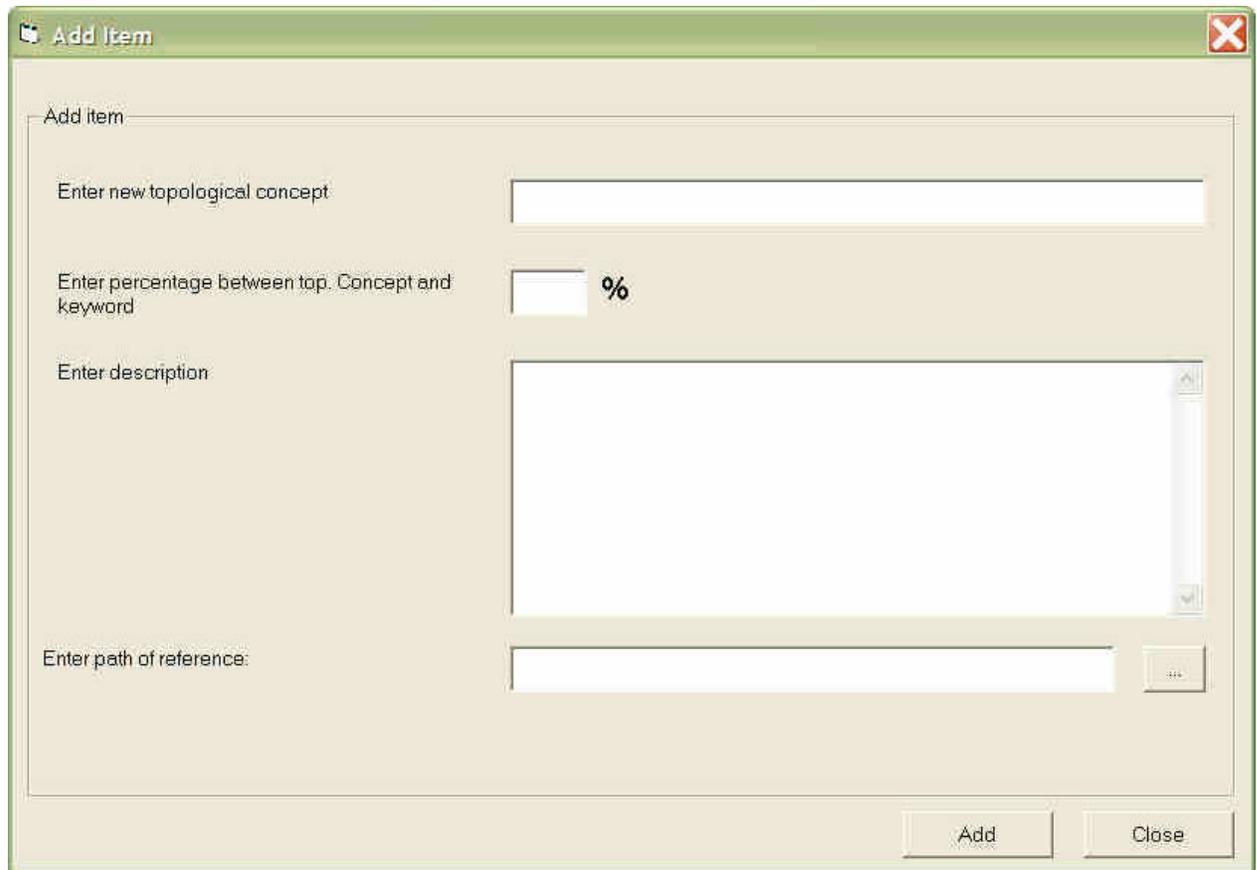


Figure 19: Properties of a Theme

This screen is very similar to the one in Figure 18, except that TC is replaced by Theme and CP is replaced by Facts.



The screenshot shows a window titled "Add Item" with a close button in the top right corner. The window contains the following fields and controls:

- Enter new topological concept:** A single-line text input field.
- Enter percentage between top. Concept and keyword:** A small text input field followed by a percentage sign (%).
- Enter description:** A large multi-line text area with a vertical scrollbar on the right side.
- Enter path of reference:** A single-line text input field with a browse button (three dots) to its right.

At the bottom right of the window, there are two buttons: "Add" and "Close".

Figure 20: Adding an item

The Add Item function is used to enter new TCs, CPs, Themes and Facts. Recommendation uses the same interface, with the slight difference that instead of one percentage for the relationship, a range, a min and a max have to be provided. This screen appears after the Add Topological Concept button in Figure 16 (Modify Keyword) has been pressed. Not all fields are mandatory!

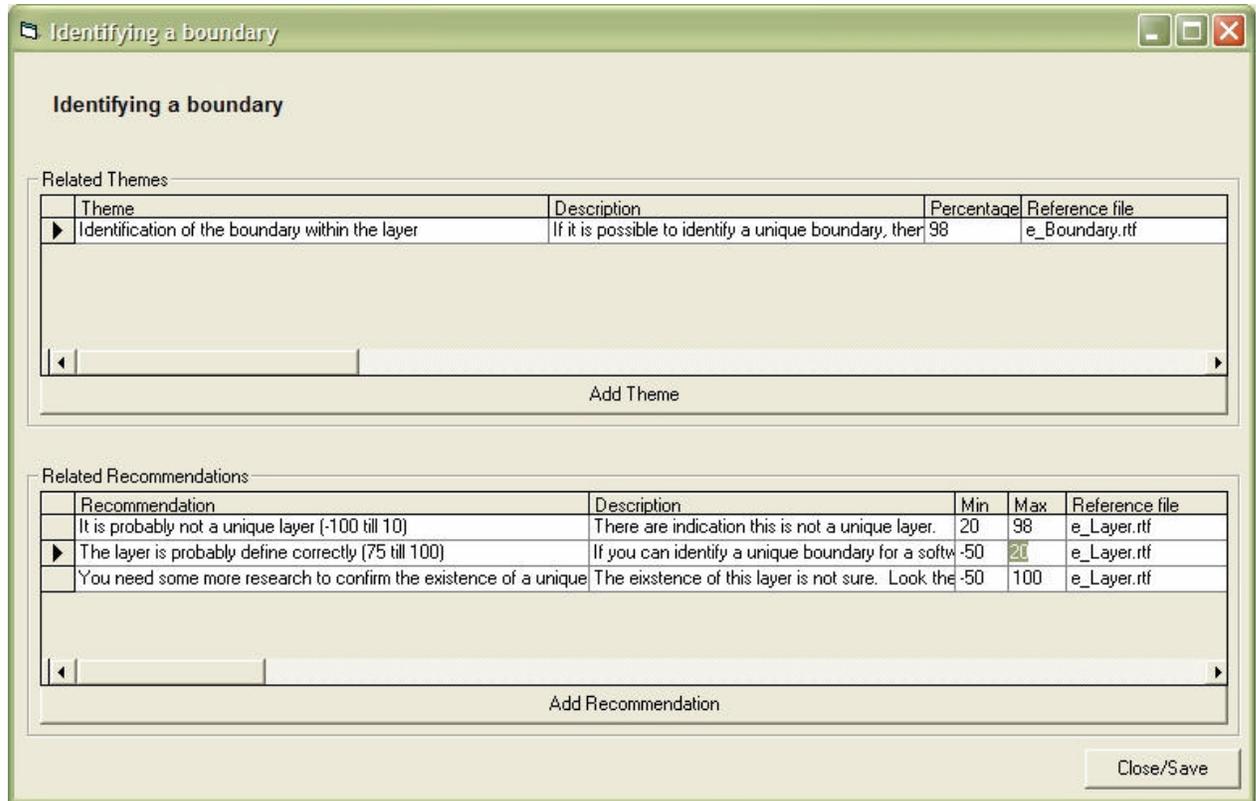
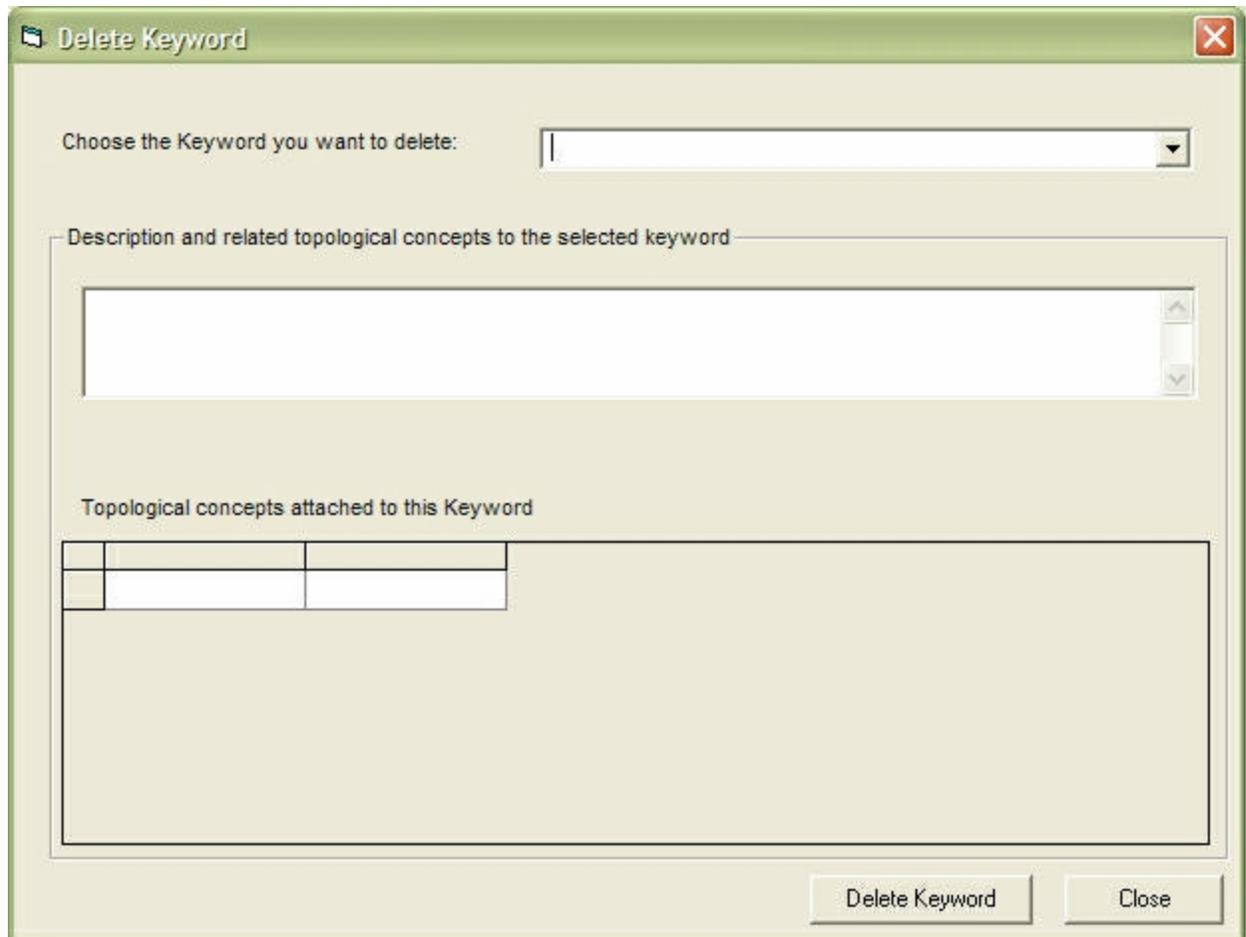


Figure 21: Properties of a Case Problem

Figure 21 is basically the same as Figure 20, with the difference that, from a hierarchical point of view, Case Problems have two different items beneath them: Themes and Recommendations.

Note: Case-specific knowledge has been entered in only one language up to now. Another interface is required to fully take care of the language issue.



Choose the Keyword you want to delete:

Description and related topological concepts to the selected keyword

Topological concepts attached to this Keyword

Delete Keyword Close

Figure 22: Deleting a Keyword

The desired Keyword is selected using the combo box at the top. Once this has been done, the Description and related TC box are populated automatically. On hitting the Delete button, the user will be asked if TCs related only to this particular Keyword are to be deleted as well. TCs related to any other Keyword will not be deleted in any case.

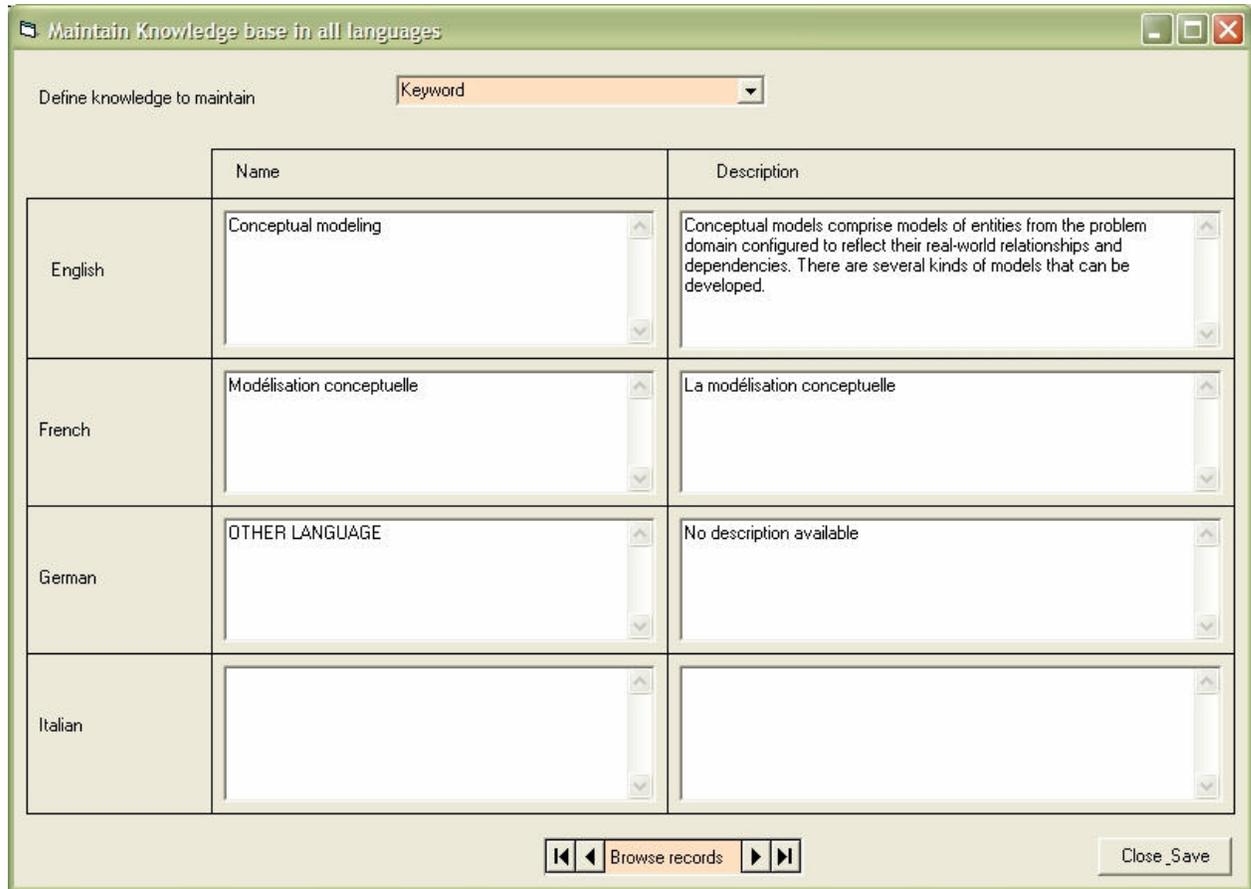


Figure 23: Maintaining case-specific knowledge

The title should be “Maintain Knowledge Base in all languages” or “Define knowledge to be maintained” in Figure 23.

Using the combo box at the top, the user decides what part of the expert knowledge (Keywords, TCs, CPs, Themes, Facts or Recommendations) is to be maintained. From there, the expert can translate each item's name and description into the desired language. Its percentage is independent of the language, so that the Knowledge Base can be used interchangeably between different countries. Usability considerations led to the decision to have just one reference file per item, and each file contains text in all four languages. Translating a (large) reference text will be the final task for an expert to perform. So, to avoid having no reference whatsoever, it was felt that it might be better to at least have access to it in another language.

⁸ To look through the records, use the Browse records control key.

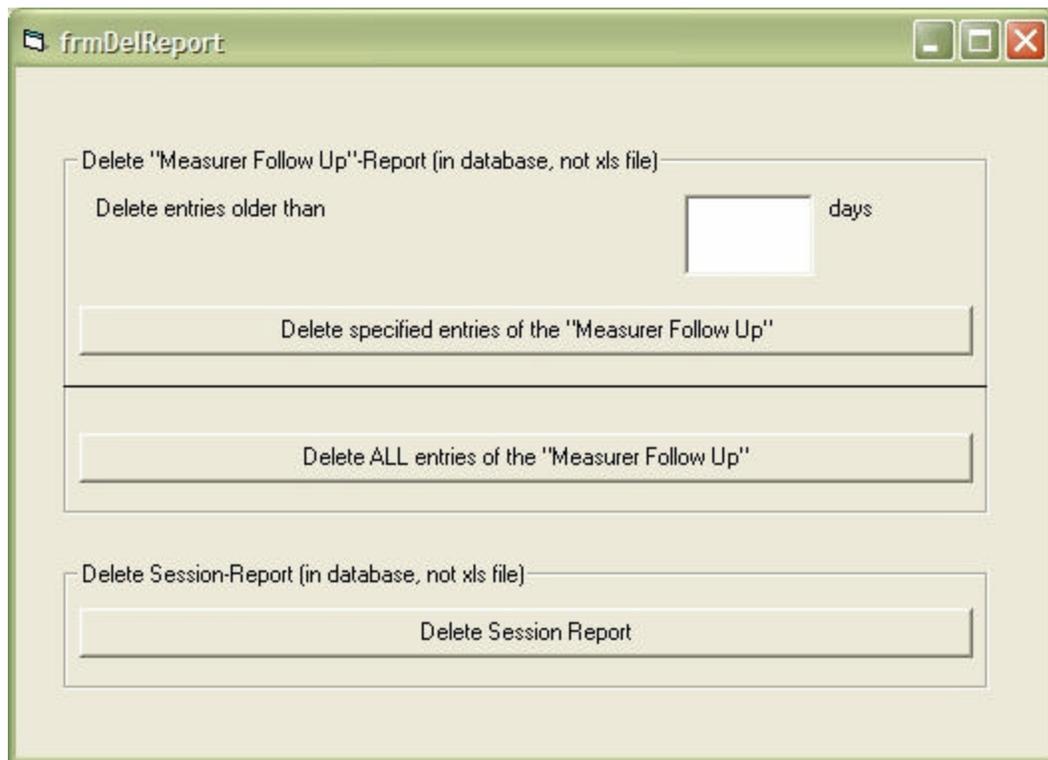
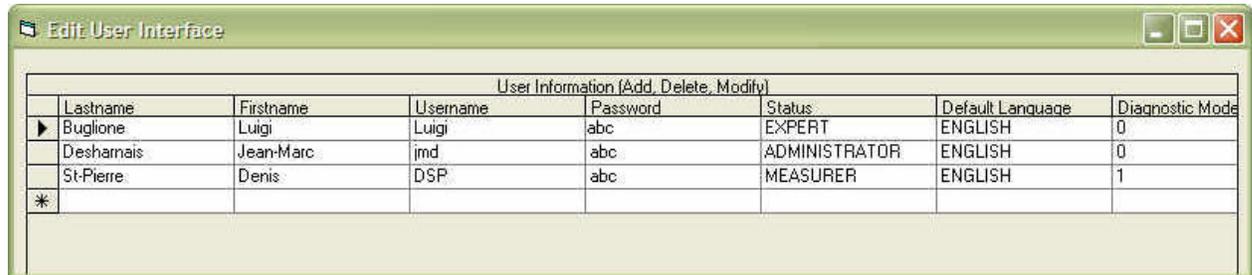


Figure 24: Managing Reports

The Reports menu in the Expert Interface allows the expert to receive measurer follow-up and session information. The object of Measurer Follow-up is to have a record of who searched for what Keywords, and their related Case Problems and Themes, in a session, as well as the Facts chosen, the quality of the documentation and the Recommendations calculated. The Session Report contains the username along with the session number and start and end dates.

To avoid the accumulation of huge tables which are difficult to read, an expert can delete these reports. They can either be deleted immediately or the user can specify that the Measurer Follow-Up be deleted after x days.

Administrator Mode



The screenshot shows a window titled "Edit User Interface" with a table of user information. The table has columns for Lastname, Firstname, Username, Password, Status, Default Language, and Diagnostic Mode. There are three rows of data and a row with an asterisk for adding new users.

User Information (Add, Delete, Modify)							
	Lastname	Firstname	Username	Password	Status	Default Language	Diagnostic Mode
▶	Buglione	Luigi	Luigi	abc	EXPERT	ENGLISH	0
	Desharnais	Jean-Marc	jmd	abc	ADMINISTRATOR	ENGLISH	0
	St-Pierre	Denis	DSP	abc	MEASURER	ENGLISH	1
*							

Figure 25: Adding, deleting or modifying user information

This feature can be accessed only if the user status is that of administrator; otherwise the item appears disabled on the menu bar. With the help of Edit User Interface, a new user can be added, or an existing one deleted or modified, e.g. their status, expiry date, diagnostic model or first and last names. Usually, a user's username, password and default language are maintained by the user.

**Annexe C: Description de l'environnement de développement du
prototype 2**

Développement du prototype 2

François Gruselin, Julien Vilz et Jean-Marc Desharnais
11 décembre, 2002

Développement

Page JSP

Pour développer le prototype nous avons choisi la technologie Java Server Page. Java Server Page (JSP) est une technologie qui a pour but de contrôler le contenu ou l'apparence d'une page Internet par l'utilisation de servlets. Les servlets sont de petits programmes qui sont décrits dans les pages Internet et qui s'exécutent sur le serveur Internet afin de modifier la page Internet avant de l'envoyer vers l'utilisateur qui la demandé. JSP est comparable à la technologie Active Server Page (ASP) de Microsoft. Une page JSP appelle un programme Java qui s'exécute sur la serveur.

Nous avons combiné JSP avec XSL décrit plus haut afin de produire les différentes vues de la base de connaissance et les fonctionnalités de l'outil de diagnostic.

La technologie JSP va nous permettre de transformer du côté du serveur les fichiers XML pour les visualiser. Nous avons voulu éviter de laisser ce travail au navigateur Internet car seule les dernières versions sont capables de le faire et certaine transformation impliquant plusieurs fichiers XML sont plus faciles à réaliser avec la combinaison de JSP et XSL qu'avec XSL tout seul.

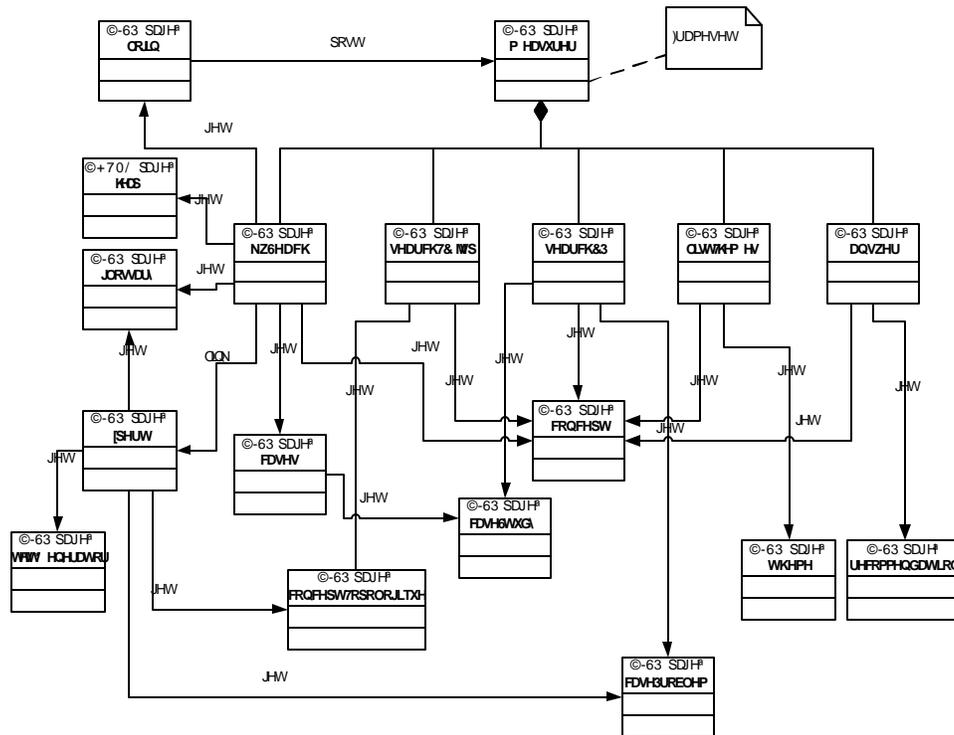


Figure 3 : Carte de CosmicXpert

La Figure 22 : Carte de CosmicXpert montre les liens entre les pages JSP de l'application. Les lien *link* représentent un simple lien hypertexte entre les documents tandis que *post* et *get* représentent un passage d'information entre les pages via les méthode http post et get.

login.jsp

L'application commence par la page login.jsp, elle permet l'ouverture d'une session par l'entrée un login et un mot de passe.

Les pages JSP peuvent recevoir des informations en provenance d'autres pages ou du client grâce à la méthode post ou get.

La page login peut recevoir un attribut « fail » qui indique que l'ouverture de session à échoué.

mesurer.jsp

Measurer.jsp reçoit le login et le mot de passe de la page login.jsp et vérifie l'existence de l'utilisateur. En cas d'échec elle renvoie le paramètre « fail » à la page de login sinon elle affiche les 5 pages qui constitue l'outil de diagnostic : kwSearch.jsp, searchTC.jsp, search.CP.jsp, listThemes.jst et answer.jsp.

kwSearch.jsp

La page kwSearch.jsp affiche le menu de l'application en fonction du type d'utilisateur (« admin » ou « mesurer »). C'est via cette page que l'utilisateur sélectionne un mot clé et peut visualiser les définitions des mots clés.

searchTC.jsp

La page searchTC.jsp est responsable d'afficher les résultats de la recherche des concepts topologiques associés au mot-clé dont l'identifiant est reçu en paramètre lors du chargement de la page.

L'utilisateur sélectionne un concept topologique parmi les résultats de la recherche.

searchCP.jsp

Le concept topologique sélectionné sur la page searchTC.jsp est passé en paramètre à la page searchCP.jsp. Cette dernière affiche tout les cas problème associés au concept topologique sélectionné.

L'utilisateur sélectionne un cas problème parmi les cas problème affichés.

listThemes.jsp

En fonction du concept topologique sélectionné, listThemes.jsp affiche une liste de thèmes ainsi que des listes de propositions pour y répondre.

L'utilisateur choisit un fait parmi les propositions pour chacun des thèmes.

answer.jsp

La page answer.jsp reçoit en paramètre les faits choisis par l'utilisateur sur la page listThemes.jsp. En fonction des faits, cas problème et concept topologique choisis, la page affiche une recommandation.

topologicalConcept.jsp

La page topologicalConcept.jsp reçoit en paramètre le chemin vers un fichier XML de type Concept topologique. Elle est responsable de le transformer et de l'afficher.

Un deuxième paramètre optionnel permet de d'indiquer à la page que l'utilisateur est un expert et qu'il désire voir tout le contenu du fichier.

caseProblem.jsp

CaseProblem.jsp a les mêmes fonctionnalités que la page topologicalConcept.jsp mais elle reçoit le chemin vers un fichier XML de type Cas problème.

caseStudy.jsp

La page caseStudy.jsp a comme responsabilité d'afficher les informations contenues dans les fichiers XML de type étude de cas.

Un paramètre permet d'afficher uniquement les contextes utilisés dans la description des cas problème affiché par la page caseProblem.jsp.

recommendation.jsp

La page recommendation.jsp reçoit le pourcentage calculé par la page answer.jsp en fonction des faits choisis. En fonction de ce pourcentage elle affiche le document de recommandation approprié. Les données proviennent du fichier XML du cas problème sélectionné.

themes.jsp

La page thèmes.jsp reçoit en paramètre l'identifiant d'un thème du concept topologique sélectionné. C'est ce thème que la page doit afficher.

glossary.jsp

La page glossary.jsp permet d'afficher la définition du mot clé dont elle reçoit l'identifiant en paramètre. Si elle ne reçoit pas de paramètre, elle affiche tous les mots clés avec leur définition.

Les mots clés proviennent du fichier glossary.xml

concept.jsp

La page concept.jsp a les mêmes fonctionnalités que glossaire.jsp mais les définitions et mots clés qu'elle affiche sont les concepts liés à l'application CosmicXpert et non les mots clés de la méthode de mesure.

Les concepts proviennent du fichier concept.xml.

xpert.jsp

A partir du fichier xpert.xml, la page xpert.jsp affiche un tableau reprenant les dépendance entre les concept topologique et les mots-clé ainsi que entre les concepts topologique et les cas problème.

Cette page contient l'accès aux rapports permettant la vérification et la validation de la base de connaissance.

testGenerator.jsp

La page testgenerator.jsp produit un rapport permettant la validation des recommandations. Elle fournit les résultats de toutes les combinaisons de réponses aux thèmes pour un cas problèmes et affiche la recommandation adéquate.

Cette page reçoit en paramètre le chemin vers le fichier XML du cas problème a tester.

cases.jsp

La page cases.jsp donne une liste de toutes les études de cas intégrées à la base de connaissance. Elle donne l'accès à l'ensemble des contextes liés aux études de cas et utilisés dans les différents documents de la base de connaissance (lien vers caseStudy.jsp). Ce rapport permet de valider le contenu de ces contextes.

XSL

Pour transformer les fichiers XML en HTML nous avons préféré le langage XSL à des méthodes en Java pour faciliter la maintenance. Nous pouvons en effet modifier l'apparence d'un document sans devoir recompiler l'application. La modification du fichier XSL suffit.

Les pages JSP utilise les fichiers contenant les scripts XSL pour transformer les fichiers XML. Ce processus se fait sur le serveur.

Vous trouverez ci-dessous un tableau reprenant tout les scripts XSL, leur objectif, les paramètres nécessaires ainsi que les pages JSP où ils sont utilisé.

XSL	XSD associé	Description	Paramètre	Utilisé dans
kwSearch.xsl	glossary.xsd	Crée à partir d'un glossaire une liste de choix de mots-clé		kwSearch.jsp
keyword.xsl	glossary.xsd	Affiche la définition et la référence d'un mot clé	kw: identifiant du mot clé à afficher	glossary.jsp, concept.jsp
glossary.xsl	glossary.xsd	Affiche l'entièreté du glossaire		glossary.jsp, concept.jsp
resultTC.xsl	xpert.xsd, tc.xsd	Crée un tableau affichant les concepts topologique associés au mot-clé.	kw: identifiant du mot clé pour lequel on cherche des concepts topologique	searchTC.jsp
tc.xsl	tc.xsd	Affiche un concept topologique pour l'utilisateur		topologicalConcept.jsp
tcXpert.xsl	tc.xsd	Affiche un concept topologique pour un expert (affiche en plus tout les thèmes et les pourcentages)		topologicalConcept.jsp
resultCP.xsl	xpert.xsd, cp.xsd	Crée un tableau affichant les cas problème associés au concept topologique.	TC: le chemin vers le fichier du concept topologique choisi	searchCP.jsp
cp.xsl	cp.xsd	Affiche un cas problème pour l'utilisateur		caseProblem.jsp
cpXpert.xsl	cp.xsd	Affiche un cas problème pour un expert (affiche en plus tout les recommandations et leurs pourcentages)		caseProblem.jsp
case.xsl	cs.xsd	Affiche le résumé d'une étude de cas		caseStudy.jsp
cases.xsl	cs.xsd	Affiche tout les contextes utilisés dans les cas problème pour une étude de cas		caseStudy.jsp
listTheme.xsl	tc.xsd	Affiche un tableau contenant les thèmes et des listes de sélection de faits pour chaque thème		listThemes.jsp
th.xsl	tc.xsd	Affiche les informations concernant un thème	TH: l' identifiant du thème à afficher	theme.jsp
cpTh.xsl	cp.xsd	Affiche l'explication du cas problème		theme.jsp
answer.xsl	cp.xsd	Affiche la réponse et un lien vers la recommandation en fonction du pourcentage calculé à partir des réponse aux thèmes	cf: pourcentage calculé à partir des réponse aux thèmes	answer.jsp

rec.xsl	cp.xsd	Affiche le document de recommandation en fonction du pourcentage	cf: pourcentage calculé à partir des réponse aux thèmes	recommandation.jsp
xpert.xsl recTest.xsl	xpert.xsd cp.xsd	Affiche la recommandation pour le test en fonction du pourcentage à tester	cf: pourcentage calculé à partir des réponse aux thèmes	testGenerator.jsp
cpTest.xsl	cp.xsd	Affiche les informations nécessaire au test concernant un cas problème		testGenerator.jsp

**Annexe D: Procédure de vérification de la base
de connaissances du prototype 1**

Procédure de vérification de la base de connaissance du prototype 1

Vérifier le contenu des documents de référence pour les mots clefs et les concepts topologiques. S'assurer que pour les concepts topologiques on respecte le gabarit. Vous aurez une idée du gabarit en regardant comment la plupart des concepts topologiques sont décrits.

Faire une vérification par cas problème. S'assurer que tous les cas problèmes sont vérifiés. À cet effet voir le document suivant:

Case problems (by file name)	Keywords
eep_Application program interface.rtf	Application program interface
eep_Boundary Rice Cooker.rtf	Boundary
eep_Characteristics of a Triggering Event.rtf (How to...)	Triggering event
eep_Clock signal RC.rtf	Triggering event
eep_Control Heater RC.rtf	Functional process
eep_Control Indicator Lamps RC.rtf	Functional process
eep_Cooking Mode.rtf	Data group
eep_creating a report RG.rtf	Triggering event
eep_Elapsed Time.rtf	Data group
eep_End user layer RG.rtf	Layer
eep_Entry.rtf (How to identify an Entry)	Entry, sub processes
eep_Exit.rtf (How to identify an Exit)	Exit, sub processes
eep_Generate a report RG.rtf	Functional process
eep_Generator layer RG.rtf	Layer
eep_List of parameters RG.rtf	Functional process
eep_Mode Switch.rtf	Entry
eep_parameters RG.rtf	Data group
eep_Read cooking mode.rtf	Read
eep_Read target temperature.rtf	Read
eep_Read.rtf (How to identify a read)	Read, sub processes
eep_Receive elapsed time.rtf	Entry
eep_Receive triggering event.rtf	Entry
eep_Set Cooking Mode RC.rtf	Functional process
eep_Set heater on-off.rtf	Exit
eep_Set status to lamp.rtf	Exit
eep_Set Target Temperature RC.rtf	Functional process
eep_Software users.rtf	User
eep_Status.rtf	Data group
eep_Temperature.rtf	Data group
eep_Triggering event user press a switch RC.rtf	Triggering event
eep_Write mode RC.rtf	Write
eep_Write temperature RC.rtf	Write
eep_Write.rtf (How to identify a write)	Write, sub processes

eep_X seconds Clock Signal.rtf

Data group

Pour chaque document de référence i.e. cas problème, thème, événement, recommandation:
vérifier l'écriture du texte (ex: typos et syntaxe)

s'il suit le gabarit

si le contenu est conforme à ce qui est attendu

Il faut aussi, pour chaque concept, vérifier les pourcentages (principalement les recommandations).

Voir aussi s'il ne manque pas de cas problèmes (ex: cas problème, comment identifier un processus fonctionnel) ou encore des thèmes (maximum de 4 normalement. S'il y a plusieurs exceptions notables on pourra passer à 5 thèmes).

Est-ce que la documentation du cas (ex: Rice Cooker et Report Generator) est claire. Sinon, quelles améliorations suggérez-vous?

Annexe E: Curriculum des experts pour le prototype 1

Denis St-Pierre

Téléphone: (450) 923-0536

DSA Consultation, 5555 Vauclin, Brossard, QC, J4W 1P9

Denis St-Pierre est un consultant senior en amélioration de processus et mesures. Il a fait de la consultation de 1986 à ce jour dans le domaine de l'amélioration des processus (CMM) de la mesure fonctionnelle (FPA et COSMIC-FFP) auprès de corporations nord américaines, asiatiques et européennes. Il a réalisé un grand nombre de projets d'amélioration de processus de mesures. Il a, dans les années 1990, contribué significativement à la norme IFPUG. À partir du milieu des années 1990 il a participé à la création de FFP (Full Function Point), devenue la norme COSMIC-FFP en 2002 (ISO 17961) Il est co-auteur de la méthode de mesure fonctionnelle COSMIC-FFP. Il a réalisé des contrats de mesures fonctionnelles dans le cadre de contrat d'impartition dans plusieurs grandes entreprises. Voici quelques clients de Denis St-Pierre pour la mesure fonctionnelle, l'implantation de programmes de mesures et l'amélioration des processus: Alstom, Bombardier, CGI, Commonwealth Bank of Australia, DMR, EDS, Gouvernement du Québec, Gouvernement du Canada, Hydro Québec, Hewlett Packard, IBM, KPMG, Nissan, Nortel, Sunlife, Telecom (Nouvelle-Zélande) et Telstra (Australie).

Il a aussi participé à plusieurs projets de recherches avec l'UQAM et publié plusieurs articles scientifiques que vous pouvez consulter sur le site du LRGL.

Scolarité

1988 Maîtrise en informatique, Université de Sherbrooke,

Sujet de maîtrise: mesure de logiciels

1986 Baccalauréat Coopératif en informatique, Université de Sherbrooke

1983 Certificat en administration, Université du Québec à Trois-Rivières

1982 DEC en administration

Présentation de Francis Dion

Monsieur Francis Dion est un gestionnaire de projet senior ayant 14 années d'expériences en développement de logiciel et amélioration des processus. Ses talents de gestionnaire et son leadership lui permettent de livrer les projets à temps, dans le respect des budgets et avec un haut degré de qualité.

Il possède une connaissance approfondie des meilleures pratiques de l'industrie, notamment le Rational Unified Process (RUP), le Unified Modeling Language (UML) le Microsoft Solution Framework (MSF) et le Capability Maturity Model – Integrated (CMMI). En tant que spécialiste des processus logiciels et des mesures, il a présenté deux articles à la «International Workshop on Software Measurement». Il a également contribué au «Guide to the Software Engineering Body of Knowledge» (un projet de la IEEE – Computer Society) et au développement de la méthode Cosmic-FFP (mesure de la taille fonctionnelle).

Sa maîtrise du processus de développement et sa compréhension des défis techniques impliqués l'amènent à anticiper les difficultés et à prendre au moment opportun les mesures qui s'imposent. Ses talents de leader, de mentor et de motivateur en font un guide efficace et respecté par les équipes qu'il dirige. Une éthique de travail exceptionnelle et un engagement inflexible envers le client et l'utilisateur final lui permettent de systématiquement livrer à temps un produit rencontrant les attentes de toutes les parties impliquées.

Présentation de HASSAN DIAB
1085, Des Seigneurs # 313
Sherbrooke, Quebec, Canada J1H 5V3
(819) 565 3671. Hassan.Diab@DMI.USherb.CA

Research assistant, Teacher, and metrics specialist. Experience in research and development of software: software measure, analysis and design, relational database modelling and design, development methodologies, review and testing, participating in international conferences, and publishing papers/technical reports in software metrics.

PROFESSIONAL EXPERIENCES

✍

Research Assistant

1996 - 2003

Sherbrooke University, Sherbrooke, Qc, Canada

- ? Define a theoretical framework for measuring software functional size from Rational Rose RealTime (RRRT) specifications

Work in collaboration with CGI and Bell-Sigma companies to define a formal approach for measuring software size from specification written in the B language

- ? Write and publish papers and technical reports treating issues about software measurement
- ? Participate in international conferences and workshops about software metrics
- ? Review and evaluate papers for publishing purposes in *Computer Society*
- ? Identify functional users requirements, design and develop a tool (called *McRose*) measuring the functional software size from RRRT specifications
- ? Define/execute testing scenarios and debug programs in order to validate the correctness of *McRose*

RELATED EXPERIENCES

○

Teacher

1996 -

2003

Sherbrooke University, Sherbrooke, Qc, Canada

- ? Teach courses in software engineering, database, artificial intelligence, software metrics, management of software development, and programming language
- ? Prepare exams, assignments, and solutions

EDUCATION

Ph.D. in Software engineering (in progress) 2003

Sherbrooke University, Sherbrooke, Qc, Canada

M.sc. in Software engineering
B.sc. in Computer Science

1999
1996

``Conservatoire National des Arts et M tiers (CNAM)'' France

M. HO Tuong Vinh

Ph.D., M.Sc., Ing.

POSITION ACTUELLE

Professeur d'informatique à l'Institut de la Francophonie pour l'Informatique de Hanoi (Vietnam): Octobre 2000 – présent

Domaine de recherche : Génie logiciel, Gestion de logiciels, Processus logiciel

ÉDUCATION

Ph.D. en Génie Électrique (Génie informatique) : Janvier 1995- Janvier 1999

École Polytechnique de Montréal, Canada

M.Sc. (Maître ès sciences) (Génie informatique) : Septembre 1991 - Août 1994

Université du Québec à Chicoutimi (UQAC), Canada

Baccalauréat en Génie électrique : Août 1981 – Juin 1986

École Polytechnique de Hanoi, Vietnam

Département de génie électrique

INTÉRÊTS DE RECHERCHE

Génie logiciel, Gestion de logiciels, Processus logiciel

Réseaux de neurones

Reconnaissance des formes, Reconnaissance de la parole

EXPÉRIENCES PROFESSIONNELLES

Chercheur post-doctoral au Laboratoire de recherche en gestion de logiciels, Université du Québec à Montréal : Octobre 1998 – Septembre 2000

Domaine de recherche: Génie logiciel, Gestion de logiciels, Processus logiciel

La méthode de mesure de la taille fonctionnelle de logiciels COSMIC-FFP : implication dans l'élaboration de plusieurs études de cas, la révision du manuel, l'application et la formation

Consultant pour Alcyonix Inc. (Canada) dans un projet dont l'objectif est d'évaluer la maturité du processus logiciel (selon le modèle Capability Maturity Model) des sous-traitants de logiciels pour le compte de Bombardier Inc. (Canada)

EXPERTISE TECHNIQUE

Méthodologies de génie logiciel, gestion de logiciels, mesures de taille fonctionnelle (IFPUG, COSMIC-FFP)

Langages de programmation : C, C++, Java, Pascal, Fortran, Modula

Technologies d'objets orientés

Modélisation de logiciels pour UNIX, WINDOWS, stations SUN, et PCs

Annexe F: Liste des associations internationales: mesure fonctionnelle

Voici la liste des associations de mesures à travers le monde qui font la promotion des points de fonction:

[ASMA](#) (Australia; Australian Software Metrics Users Association)

[BFPUG](#) (Brazilian Function Point Users Group)

[CIM](#) (Canada)

[DANMET](#) (Denmark)

[DASMA](#) (Germany; Deutsche Anwendergruppe für Software Metrik und Aufwandschätzung)

[FFPUG](#) (France; French Function Point Users Group)

[FiSMA](#) (Finland; Finnish Software Metrics Association FiSMA)

[FPUGA](#) (Austria; Function Point Users Group Austria)

[GUFPI](#) (Italy)

[IFPUG](#) (United States; International Function Point Users Group)

[ISBSG](#) (International Software Benchmarking Standards Group Ltd)

[ITB](#) (Platform van **IT-Beroeps-** en vakverenigingen in Nederland)

[JFPUG](#) (Japan; Japanese Function Point Users Group)

[NESMA](#) (Netherlands; Netherlands Software Metrics Users Association)

[SASMA](#) (South Africa; South African Software Metrics Association)

[UKSMA](#) (United Kingdom; United Kingdom Software Metrics Association)

Annexe G: Exemples de "règles locales"

B.1 Introduction

B.1.1 Objective of this document

Present a cohesive set of local rules on how to apply COSMIC 2.1.

B.1.2 Reasoning behind the Local Rules

FFP is a measure standardized by the international organization COSMIC. While most counting rules are described in the COSMIC FFP measurement manual, some rules must be clarified to determine how to apply them. In some cases, local rules make the counting process more efficient.

B.1.3 Target Audience

This document is targeted to measurement specialists that already have some training and COSMIC FFP experience. It is structured as a reference book, so it is intended to be used as such rather than a training material. Use the table of contents and the index to locate a specific counting issue.

B.1.4 How to find your way in this document

This document presents measurement issues grouped by topics (e.g. boundaries, layers, group of data). The table of contents and the index can be used to find a specific detailed topic. If you read this document on screen rather than on paper, you can search for a topic using the “Find” function of Microsoft Word. If you do this be aware that specific topic titles are followed by “&” in hidden (e.g. trigger &) characters in order to facilitate searching. Also, navigation can be facilitated by using the hypertext links.

In this document, extracts from the COSMIC FFP manual are identified in italic characters.

B.2 General Rules

B.2.1 Purpose of the count

The purpose of the count is the reason why the count is done. For example, it can be to estimate the cost of a project, or to size an existing application. These are the two main count purposes: Estimation and Application. You will find in this document different ways of applying FFP local rules for Estimation

counts and Application counts. For example, section 2.6 is about Boundaries and Scope of Estimation counts and section 2.7 is about Boundaries and Scope of Application counts.

B.2.2 Boundary

Boundary definition:

“The boundary of a piece of software is the conceptual frontier between this piece and the environment in which it operates, as it is perceived externally from the perspective of its users. The boundary allows the measurer to distinguish, without ambiguity, what is included inside the measured software from what is part of the measured software’s operating environment.” COSMIC FFP v2.1 Measurement Manual.

B.3 COSMIC-FFP Processes

Here is the definition of a COSMIC-FFP Process:

“A functional process &is a unique and ordered set of data movements (entry, exit, read, write) implementing a cohesive set of Functional User Requirements. It is triggered by an event and, once performed, must leave the software in a coherent state with respect to the triggering event.” COSMIC FFP v2.1 Measurement Manual

Here are additional guidelines to identify FFP process:

On completion of the FFP process the user can exit the application without corrupting their data or leaving their business process incomplete.

The transaction achieves a “business goal” and often have an equivalent in a manual system.

The transaction does not exist for technical or implementation reason.

The transaction completes a single unit of work.

The transaction can be triggered and complete its processing independently, that is, it does not have a direct synchronous link to other functions.

Subset processes are not process on their own. Here are examples of subset processes:

Report which can optionally print or hide fields.

A single transaction which can operate in multiple ways (e.g.: depending on the contents of fields, other fields become optional or mandatory).

An inquiry with multiple selection criteria.

Hint: Sequences of events which end with an “OK” button to save or output data (not “OK” to move to another windows only) are often FFP processes.

Note on FFP processes: take care not to decompose the functionality{xe “Decomposition of the functionality”} beyond the lowest level. For example, the level of calculating individual field, is too low. On the other hand, take care not to decompose enough. For example, Manage New Employee must be decomposed if there are different types of activity: Create a new employee, Change an employee, Delete an employee.

Trigger &: (note: italic text is text extracted from the official FFP Measurement Manual)

Triggering event (-type): A triggering event occurs outside the boundary of the measured software and initiates one or more functional processes. Clock and timing events can be triggering events. Since each identified layer is separated by a boundary, triggering events can occur in one layer and initiate functional processes belonging to another layer.

Functional process (-type) (Synonym ‘Transaction-type’): A functional process is a unique set of data movements (entry, exit, read, write) implementing a cohesive and logically indivisible set of Functional User Requirements. It is **triggered directly, or indirectly via** an ‘actor’, by an Event (-type) and is complete when it has executed all that is required to be done in response to the triggering Event (-type).

Functional process principles

- a) *A functional process is derived from at least one identifiable Functional User Requirement,*
- b) *A functional process is performed when an identifiable **triggering event** occurs,*
- c) *A functional process contains at least two data movements, an entry and an exit or a write,*
- d) *A functional process contains no more than one self-induced wait state (which may occur when it is completed),*
- e) *A functional process belongs to one, and only one, layer.*

Boundary rule

Start by identifying triggering events, then identify the functional processes enabled by those events. The boundary lies between the triggering events and those functions.

Functional process rules

Subsets of triggering events are not considered different triggering events.

For instance, if a specific event occurrence triggers the entry of a data group comprising data attributes A, B and C, and then another occurrence of the same event-type, triggers an entry of a data group which has values for attributes A and B only, this is not considered a different triggering event-type. It is considered to be the same for the purpose of identifying COSMIC-FFP functional processes. Consequently, only one entry and one functional process are identified, manipulating data attributes A, B and C.

In the context of real-time software, a functional process is also triggered by an event. It terminates when a point of asynchronous timing is reached. A point of asynchronous timing is reached when, in a sequence of data movements, a given data movement is not synchronized with the one preceding it. A point of asynchronous timing is equivalent to a self induced wait state.

Entry rules

Clock-triggered events are considered external. Therefore, an event occurring every 3 seconds is associated with an ENTRY moving one data attribute, for instance. However, the functional process that generates the event periodically is ignored since it occurs, by definition, outside of the software boundary.

Minor variations do not justify different FFP processes, even if they have different external triggers. As a variation of the previous example, if there are separate menu items for creating Permanent Employees and Temporary Employees, then there is still only one process.

Polling

In some circumstances we can contrive that the events of interest in the external world directly generate messages which form the input side of a logical transaction. In other circumstances, a software application must periodically inspect (c'est-à-dire "poll") the status of the external world to determine whether an event of interest has occurred, generating an input message to document a positive result. In either case, the resulting message is regarded as Entry (the detection mechanism is purely implementation detail). The polling mechanism is, of course, triggered by an event in the external world i.e. the passage of a specified amount of time.

From: MARK II CPM 1997 and modified to fit COSMIC FFP

Annexe H: Études de cas

Ces études de cas sont disponibles sur le site du LRGL à l'UQAM.

H.1 Data Warehouse

<http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/>

H.2 Rice Cooker

<http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/>

H.3 Valve Control

<http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/>

H.4 Générateur de rapport

(Voir texte ci-joint)

F.4 Report/Query Generator

Application profile

The Report or Query Generator application is designed to produce reports. The actual example is generic, not specific to a particular Report or Query Generator. Its specifications are presented in the appendix.

- General information about the application to be measured

Application domain: Infrastructure

Application type: MIS

- Available information about the application

Document: Specifications document

IDENTIFY COSMIC-FFP SOFTWARE FUNCTIONAL USER REQUIREMENT

BOUNDARY

User

Human

Layer

It is possible to identify two layers: the end user layer and the report generator layer.

Boundary

Based on the specification, we can identify the application boundary (Figure 1). The Report Generator application gets the capability to define and create their own types of report or query from the end user using parameters (group of data) as Entry, types⁹ of report and query as Processes and Error messages and List of parameters as Exit . It sends information to the software to produce reports.

⁹ It is types of reports because our intention is not to count all the potential reports or queries, but the types of report and query the user can produce, using the functionality of the Report/Query Generator.

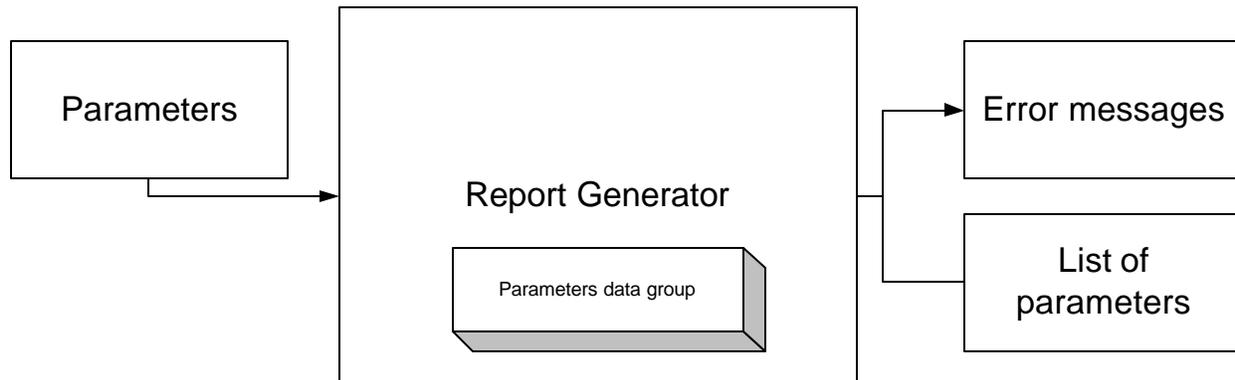


Figure 4 Report Generator

IDENTIFY CANDIDATE COSMIC-FFP TRIGGERING EVENTS, FUNCTIONAL PROCESSES AND DATA GROUPS

At this step, we will identify functional processes (from the user's point of view) that can be considered as candidates for COSMIC-FFP functional processes. Candidate triggering events and data groups will be also identified.

Identify Candidate Triggering Events

From the requirements the following triggering events was identified:

Select the parameters
Ask for a list

Identify Candidate Functional Processes

From the requirements the following functional processes are identified:

Generate a report: As indicated in the requirement each report layout (bar graph, pie chart, and line graph) should be measured as separate FFP process.

Generate a query: As indicated in the requirement each query layout (bar graph, pie chart, and line graph) should be measured like the reports.

List of parameters: As indicated in the requirement, the user can produce¹⁰ a list of parameters available in the database.

Error messages: As indicated in the requirement, if the user makes a mistake, an error message appears.

Identify Candidate Data Group

From the requirements the following data group was identified:
parameters

MAP IDENTIFIED CANDIDATE ITEMS INTO COSMIC-FFP SOFTWARE MODEL

Based on COSMIC-FFP rules the candidate items (i.e. functional processes, triggering events and data groups) are mapped into the COSMIC-FFP software model. The mapping phase must associate each functional process to a triggering event and data group manipulated by it. This phase allows also identifying layers in the context of the COSMIC-FFP software model.

The mapping phase will be performed through a procedure. The procedure also allows assessing if a candidate functional process is a COSMIC-FFP one. Each candidate process must satisfy the following questions in order to be validated as a COSMIC-FFP functional process:

Does it operate on a unique and ordered set of data movements performing a set of FURs?

Is it triggered by an event (triggering event) ?

Does the triggering event occur outside the boundary of the software?

In the following we will assess each identified candidate process.

Generate a report: As indicated in the requirement each report layout (bar graph, pie chart, and line graph) should be measured as separate FFP process.

Does it operate on a unique and ordered set of data movements performing a set of FURs? Yes, the user enters parameters that are processed by the software to create a report layout

Is it triggered by an event (triggering event)? Yes, the human user.

Does the triggering event occur outside the boundary of the software? Yes (outside the boundary of the software)

Generate a report is then a COSMIC-FFP functional process.

Generate a query: As indicated in the requirement each query layout (bar graph, pie chart, and line graph) should be measured like the reports.

Does it operate on a unique and ordered set of data movements performing a set of FURs? No, the user enters parameters that are processed by the software to create a query layout which is already done with the same characteristics based on the requirements.

¹⁰ The case study could be more complex if we add the possibility to the user to maintain the parameters. For the moment we prefer to keep the case study as simple as possible.

Is it triggered by an event (triggering event)? Yes, the human user.

Does the triggering event occur outside the boundary of the software? Yes (outside the boundary of the software)

Generate a query is then not a COSMIC-FFP functional process, because it is not unique.

List of parameters: As indicated in the requirement, the user can produce¹¹ a list of parameters available in the database.

Does it operate on a unique and ordered set of data movements performing a set of FURs? Yes, the user asks for a list of parameters that is produced by the software.

Is it triggered by an event (triggering event)? Yes, the human user.

Does the triggering event occur outside the boundary of the software? Yes (outside the boundary of the software)

Generate a list of parameters is then a COSMIC-FFP functional process

Error messages: As indicated in the requirement, if the user makes a mistake, an error message appears.

Does it operate on a unique and ordered set of data movements performing a set of FURs? Yes, to produce an error message.

Is it triggered by an event (triggering event)? Yes, by the software.

Does the triggering event occur outside the boundary of the software? No (inside the boundary by the software itself)

Generate a query is then not a COSMIC-FFP functional process, because it is not triggered by an event outside the boundary of the software.

¹¹ The case study could be more complex if we add the possibility to the user to maintain the parameters. For the moment we prefer to keep the case study as simple as possible.

IDENTIFY COSMIC-FFP FUNCTIONAL SUB-PROCESSES AND APPLY COSMIC-FFP MEASUREMENT FUNCTION

For each identified functional process, we identify Cosmic-FFP functional sub-processes within it. Details of the identified sub-processes are presented in Table 1.

We use a convention for indexing functional processes: each functional process is assigned an ID number as in the following format: x.y where x is the layer number and y is the ID number of the functional process within the layer.

There are two functional processes: generate a report type (query¹²), list of parameters.

No	ID Process	Description	Trigger	Sub-process description	Sub-process type	Data Group	FFP
1	1.1	Generate a report/query	Select a parameter	Entry of parameters	E	Parameters	1
				Read parameters	R	Parameters	1
				Produce a report type	X	Parameters	
				Error messages	X		1
2	1.2	List of parameters	Ask for a list	Ask list	E		1
				Read parameters	R	Parameters	1
				Produce a list of parameters	W	Parameters	1
				Error messages	X		1

Total COSMIC-FFP Points: 8

¹² Report and query are similar process; therefore it is necessary to measure only one of them or combine them as one process.

Report Generator or Query Generator Description

A Report Generator{xe "Query or Report Generator"} provides business users with the capability to define and create their own displays or reports using data fields which are currently available from existing FFP groups of data. Parameters are designed, developed and delivered so that a user can select the fields (and their source) to be displayed/printed, the criteria for their selection, the format in which they will be displayed, other rules for their display.

Note: Report or Query Generator does not include pre-defined queries or reports. The user of the Report Generator can also produce a list of parameters available.

End user layer:

End user layer's perspective is the whole set of delivered functionality, therefore functions (reports and queries¹³) produced by the users should be counted as long as the query and report parameters are saved and reused.

Generator & layer:

Measure the Generator Functionality.

Boundary

Report Generators may be a separate application which operates as a "utility" or "tool". They may also be functions within the boundary of another application. In that case, the application boundary may include both "local functionality" and the functionality of the generator.

Group of Data

Query or Report Generators reference data from other applications (potential FFP group of data). Generators also maintain parameters and rules selected and entered by the user. These are also FFP groups of data, e.g.: Report Format Files and Selection Criteria Files.

There may be instances of other logical groups of data that can be maintained/read by processes available via the Generator. If they deliver functionality and are maintained or

¹³ In addition to created queries and reports, read entities must be considered FFP groups of data as well as maintained entities used to save parameters to generate queries and reports.

read by the Generator, they can be considered to be FFP groups of data. An example would be a maintained list of authorized users of the report facilities.

FFP Processes

Generator parameter and rules inputs

If a function allows the user to enter and save parameters then it is a potential FFP process.

Each report layout (bar graph, pie chart, line graph) is counted as separate FFP process. In our example we assume only one report layer. The user can also produce a query layout with the same characteristics.

If there are mistakes in the process, error messages appear.

Annexe I: Formules du prototype 1

Dans le prototype 1, codé en Visual Basic de Microsoft (version 6.0), on retrouve un ensemble de formules qui sont résumées dans le trois prochaines pages en pseudo code.

Followed, pseudo code examples are written down, performing tasks from the “Path of the measurer”

- Searching a Topological Concept
- Giving priority to Topological Concepts
- Finding Case Problems
- Giving priority to Case Problems
- Showing Themes
- Interpreting the answer

1. NewSearch_click()

Disable button

Connect to database

Retrieve kw_id from selected keyword

IF keyword exist THEN

IF former search has been done THEN

Call procedure MeasurerFollowup

Else save kw_id

Reset table attributes to default values;

show to yes in Topology, Problem_gen, Question_gen, Recommendation_gen;

calculated to no in Recommendation_gen; user_answ = no, qual_doc = 100 in

Question_gen; selected = no in choice_gen

Filldatagrids

FormerSearch = True

Else Error message

Enable button

2. Filldatagrid()

If recordsets are open Then

Close recordsets and release space

Retrieve appropriate cases from database

Topological concepts: linked with the selected keyword, show = yes, lang_id = current language, topology != not translated into this language

Case Problems: problems who belong to TC which are related to the selected keyword, lang_id = current language, show = yes

Themes: Themes where TC_ID is related to the selected keyword, show = yes, lang_id = current language

Recommendations whoms TC_ID are linked with the chosen keyword, show = yes, calculated = yes, lang_id is the current

Formatdatagrids

3. MeasurerFollowUp()

Initialise database

If Former Search has been done Then

 save former KW_ID for Report

If user hit exit button Then

 save new kw_id

If FormerSearch has been done Then

 Insert the following useractivities into SUBSESSION: userid, session id, keyword id

 Insert into table ANSWER the attributes user id, session id, keyword id, tc id, p id,

 q_id, a_id, qual_doc, relationship percentage between Theme and Fact

 Insert into Results relevant data about the calculation, user id, session id, kw id, tc id,

 p id, r id, probability.

Close database connection

4. Initialisedatabaseconnection(connectionstring)

If connection is already open

 Do nothing

Else set global object as new connetion

 set cursortype to “useclient”

 open database with connectionstring (udl file)

5. Closedatabaseconnection

If connection to database is open Then

 close connection and release space for the connection object

6. Formatdatagrids

Set width of the datagrid columns to the desired size and hide unwanted columns which are necessary for update reasons

7. calculateRecom (ByVal p_id as integer)

connect to database

store information (p_id, a_id, q_chioce_perc, qual_doc) of the concerned case problem in a recordset

Select Case “number of Themes already answered for this Case Problem”

Case 1 – $CF(CP) = CF(Theme1) = qual_doc * q_choice_perc * P_Q_perc$

Case 2 – $CF(CP) = CF(Theme1) * CF(Theme2)$

Case 3 – $CF1(Theme) = CFcombine[CF(Theme1), CF(Theme2)]$

$CF(CP) = CFcombine[CF1(Theme), CF(Theme3)]$

Case 4 - $CF1(Theme) = CFcombine[CF(Theme1), CF(Theme2)]$

$CF2(\text{Theme}) = CFcombine[CF(\text{Theme3}), CF(\text{Theme4})]$

$CF(\text{CP}) = CFcombine[CF1(\text{Theme}), CF2(\text{Theme})]$

Reset former calculations by setting `calculated = 0` where `P-ID` is recent one

Set `calculated = -1` and `probability = CF(CP)` of current `P_ID`

Requery Recommendations

Rebind Recommendations datagrid

Close recordsets and release space

Annexe J: Analyses statistiques

Univariate Analysis of Variance

Between-Subjects Factors

		N
EXP	1	14
	2	11
PROTO	Non	12
	Oui	13

Descriptive Statistics

Dependent Variable: BONS

EXP	PROTO	Mean	Std. Deviation	N
1	Non	24.00	12.64	7
	Oui	48.43	12.92	7
	Total	36.21	17.65	14
2	Non	33.60	9.71	5
	Oui	33.67	13.43	6
	Total	33.64	11.31	11
Total	Non	28.00	12.08	12
	Oui	41.62	14.74	13
	Total	35.08	14.96	25

Tests of Between-Subjects Effects

Dependent Variable: BONS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Eta Squared
Corrected Model	2129.592 ^a	3	709.864	4.603	.013	.397
Intercept	29913.135	1	29913.135	193.986	.000	.902
EXP	40.843	1	40.843	.265	.612	.012
PROTO	919.734	1	919.734	5.964	.024	.221
EXP * PROTO	909.748	1	909.748	5.900	.024	.219
Error	3238.248	21	154.202			
Total	36133.000	25				
Corrected Total	5367.840	24				

a. R Squared = .397 (Adjusted R Squared = .311)

Interprétation: Pour les bonnes réponses, la valeur de F pour la relation entre les groupes et l'utilisation ou pas du prototype est de $F(1,21) = 5,900$ avec $p = 0,024$, soit inférieure à 0,05. La p-valeur se trouvant en dessous de 0,05, le résultat confirme qu'il y a un effet de prototype entre les deux groupes (EXP) de l'expérimentation même s'il n'est pas significatif sur la moyenne des deux groupes.

Univariate Analysis of Variance

Between-Subjects Factors

		N
EXP	1	14
	2	11
PROTO	Non	12
	Oui	13

Descriptive Statistics

Dependent Variable: MAUVAIS

EXP	PROTO	Mean	Std. Deviation	N
1	Non	41,43	17,68	7
	Oui	16,86	6,54	7
	Total	29,14	18,07	14
2	Non	27,00	15,75	5
	Oui	17,67	9,91	6
	Total	21,91	13,12	11
Total	Non	35,42	17,77	12
	Oui	17,23	7,91	13
	Total	25,96	16,19	25

Levene's Test of Equality of Error Variances

Dependent Variable: MAUVAIS

F	df1	df2	Sig.
1,204	3	21	,332

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.

a. Design: Intercept+EXP+PROTO+EXP * PROTO

Tests of Between-Subjects Effects

Dependent Variable: MAUVAIS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected	2673,05	3	891,01	5,175	,008
Intercept	16246,9	1	16246,9	94,35	,000
EXP	284,31	1	284,31	1,651	,213
PROTO	1762,05	1	1762,05	10,23	,004
EXP * PROTO	355,92	1	355,92	2,067	,165
Error	3615,90	21	172,186		
Total	23137,0	25			
Corrected	6288,96	24			

a. R Squared = ,425 (Adjusted R Squared = ,343)

Interprétation: Pour les mauvaises réponses, la valeur de F pour l'utilisation ou pas du prototype est de $F(1,21)=10,233$ avec $p=0,004$. La p-valeur se trouvant au dessus de 0,05, le résultat confirme que les deux groupes ne sont pas différents et sont additifs (H_0)

Oneway

ANOVA

BONS

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2129.592	3	709.864	4.603	.013
Within Groups	3238.248	21	154.202		
Total	5367.840	24			

Contrast Coefficients

Contrast	EXPPROTO			
	1	2	3	4
1	1	-1	0	0
2	0	0	1	-1
3	1	0	-1	0
4	0	1	0	-1

Contrast Tests

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
BONS	Assume equal variances	1	-24.43	6.64	-3.680	21	.00
		2	-6.67E-02	7.52	-.009	21	.99
		3	-9.60	7.27	-1.320	21	.20
		4	14.76	6.91	2.137	21	.04
	Does not assume equal variances	1	-24.43	6.83	-3.576	11.994	.00
		2	-6.67E-02	6.99	-.010	8.876	.99
		3	-9.60	6.46	-1.487	9.886	.16
		4	14.76	7.34	2.011	10.549	.07

Interprétation: Les résultats des analyses de ces deux groupes pris séparément i.e. avec une analyse de contraste (ANOVA – CONTRAST). Nous avons fait 4 catégories.

Les deux premières catégories sont les suivantes:

- comparaison, pour le groupe 1, des deux sous groupes (avec et sans prototype)
- comparaison, pour le groupe 2, des deux sous groupes (avec et sans prototype)

Les résultats sont les suivants:

- pour le groupe 1, la valeur statistique pour le contraste 1 est de $t = -3,68$ avec $p = 0,001$, soit inférieure à $0,05$. Le résultat confirme que les deux sous groupes (avec et sans prototype) du groupe 1 sont différents ;
- pour le groupe 2, la valeur statistique pour le contraste 2 est de $t = 0,01$ avec $p = 0,99$, soit supérieur à $0,05$. Le résultat confirme que les deux sous groupes (avec et sans prototype) du groupe 2 ne sont pas différents.

Les deux autres catégories comparent les non et les oui entre chaque groupe:

- comparaison entre l'expérience 1 et 2 des deux sous groupes sans prototype
- comparaison entre l'expérience 1 et 2 des deux sous groupes avec prototype

Les résultats sont les suivants:

- pour les deux sous groupes sans prototype (non), la valeur statistique pour le contraste 3 est de $t = -1,32$ avec $p = 0,20$, soit supérieur à $0,05$. Le résultat confirme que les deux sous groupes sans prototype des groupes 1 et 2 ne sont pas différents;
- pour les deux sous groupes avec prototype (oui), la valeur statistique pour le contraste 4 est de $t = 2,317$ avec $p = 0,04$, soit inférieure à $0,05$. Le résultat confirme que les deux sous groupes avec prototype des groupes 1 et 2 sont différents.

Correlations

Descriptive Statistics

	Mean	Std. Deviation	N
BONS	35,08	14,96	25
MAUVAIS	25,96	16,19	25

Correlations

		BONS	MAUVAIS
BONS	Pearson Correlation	1,000	-,297
	Sig. (2-tailed)	.	,150
	N	25	25
MAUVAIS	Pearson Correlation	-,297	1,000
	Sig. (2-tailed)	,150	.
	N	25	25

Un premier test de corrélation (Pearson) indique qu'il n'y a pas de corrélation significative entre les bonnes et les mauvaises réponses : $r = -0.297$, $p = 0.189$ de l'ensemble de l'échantillon. Un deuxième test de corrélation (corrélation de Spearman qui tient compte du rang de la réponse) indique aussi qu'il n'y a pas de corrélation significative entre les bonnes et les mauvaises réponses : $r_s = -0.143$, $p = 0.475$ de l'ensemble de l'échantillon.

Nonparametric Correlations

Correlations

		BONS	MAUVAIS
Spearman's rho	BONS	Correlation Coefficient 1,000	-,175
		Sig. (2-tailed) .	,403
		N 25	25
	MAUVAIS	Correlation Coefficient -,175	1,000
		Sig. (2-tailed) ,403	.
		N 25	25

Correlations

Descriptive Statistics

	Mean	Std. Deviation	N
BONS	36,04	14,47	24
MAUVAIS	24,04	13,32	24

Correlations

		BONS	MAUVAIS
BONS	Pearson	1,000	-,139
	Sig. (2-tailed)	.	,516
	N	24	24
MAUVAIS	Pearson	-,139	1,000
	Sig. (2-tailed)	,516	.
	N	24	24

Nous avons réalisé une analyse de sensibilité en enlevant le cas dont la mauvaise réponse est la plus haute (72) dans l'échantillon afin de vérifier si la corrélation entre les bonnes et les mauvaises réponses est toujours non significative. A cet effet, nous avons refait les tests de Pearson et Spearman. Les résultats sont les suivants : $r = -0.102$, $p = 0.475$ et $r_s = -0.040$, $p = 0.847$. Ces deux tests confirment à nouveau qu'il n'y a pas de corrélation significative.

Annexe K: Résultats des bonnes réponses

Commentaires sur les réponses

Ce sont les réponses telles que compilées. Cette documentation contient des termes anglais que nous avons conservé tels quels. Nous avons aussi noté des fautes de français que nous n'avons pas corrigées.

Description	JMD	M1O	M2	M3O	M4	M5	M6	M7O	M8O	M9O	M10	M11	M12O	M13	M14O	M16O	M17	M18O	M19	M20	M21	M22	M23O	M25O	M26O	M27O
Administrateur se branche au système	FP	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
Se branche au système	TE	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Entre alias et mot de passe	Entry	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
Valide inf sur formulaire (suppose 1 groupe de données)	Read	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Affiche la page du participant	Exit	1	0	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
Messages d'erreur (alias ou mot de passe invalide)	Exit	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Participant se branche au système	FP	1	1	1	1	0	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
Se branche au système	TE	1	1	1	1	0	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	0	0
Entre alias et mot de passe	Entry	1	0	1	1	0	0	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1	0
Valide inf sur formulaire (suppose 1 groupe de données)	Read	1	0	1	1	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0
Affiche la page du participant	Exit	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
Messages d'erreur (alias ou mot de passe invalide)	Exit	1	1	1	1	0	1	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1	1	1	1	1
Essai manqué pour branchement	FP	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
Se branche (ou essai)	TE	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Entrée Alias et mot de passe	Entry	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Validation	Read	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Refus (3 fois) ou plus de 21 utilisateurs	Exit	1	0	1	0	0	1	1	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
S'abonner au pool	FP	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
Sélectionner l'abonnement	TE	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Entrée des champs nom, prénom, courriel, tel. Alias, passe, adresse, type	Entry	1	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1
Validation (2 groupes de données: participant et passe)	Read	1	1	0	1	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1
Validation (2 groupes de données: participant et passe)	Read	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
Ajouter un participant	Write	1	1	0	1	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	1	1	1	0	0	1
Courriel	Exit	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	1	1
Messages d'erreur (formulaire incomplet, alisa existant, arobas manquant, etc.)	Exit	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Activer-désactiver un participant	FP	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1
Administrateur sélectionne l'option d'activation	TE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Lecture du formulaire d'activation et participant (1 Groupe de données)	Read	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	1
Affichage du formulaire d'activation et nom des participants	Exit	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
Sélection d'un participant par l'administrateur (active)	Entry	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1	0	0	1	1	1	1
Ecriture de l'activation (1 groupe de données)	Write	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1
Message d'erreur ou retour à la page d'administration	Exit	1	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	1
Classement de la semaine	FP	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0	0	0	0
Sélection du formulaire	TE	1	1	1	1	0	0	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	0	0	0
Année et semaine saisie	Entry	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	1	1	1	0	0	1	1	1	0	0	1	0	1	1	1	0	0	0	0	1	1	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Visualisation du formulaire demandé	Exit	1	1	1	1	0	0	1	1	1	1	0	1	0	0	1	1	1	0	0	0	1	1	0	0	0
Message d'erreur (semaine non sélectionnée)	Exit	0	1	1	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0

Description	JMD	M1O	M2	M3O	M4	M5	M6	M7O	M8O	M9O	M10	M11	M12O	M13	M14O	M16O	M17	M18O	M19	M20	M21	M22	M23O	M25O	M26O	M27O
Classement général	PF	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
Sélection du formulaire	TE	1	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
Année et semaine saisie	Entry	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	1	0	1	0	1	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Visualisation du formulaire demandé	Exit	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	1	0	0
Message d'erreur (semaine non sélectionnée)	Exit	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Meilleure performance de la semaine	PF	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
Sélection du formulaire	TE	1	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
Année et semaine saisie	Entry	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	1	1	1	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Visualisation du formulaire demandé	Exit	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	1	0	0
Message d'erreur (semaine non sélectionnée)	Exit	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Statistiques peresonnelles par semaine	PF	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
Sélection du formulaire	TE	1	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
Année et semaine saisie	Entry	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Visualisation du formulaire demandé	Exit	1	0	0	1	0	1	1	1	1	0	0	1	0	0	1	1	1	0	0	0	1	1	1	0	0
Message d'erreur (semaine non sélectionnée)	Exit	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Historique par semaine pour la saison précédente	PF	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
Sélection du formulaire	TE	1	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
Année et semaine saisie	Entry	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	0
Lecture des informations (hyp: 2 GD)	Read	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Visualisation du formulaire demandé	Exit	1	0	0	1	0	1	1	1	1	0	0	1	0	0	1	1	1	0	1	0	1	1	1	0	0
Message d'erreur (semaine non sélectionnée)	Exit	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0

Annexe L: Exemples de mauvaises réponses

Mesureur 1

ID	Type of component	Complete Identification	N M
101TR01	Triggering Event	Le participant demande à se brancher au systeme	
101P01	Functional process	Le participant complete le formulaire de branchement	
101P01EN01	Entry	Entrée l'Alias et le mot de passe	
101P01R01	Read	Le système valide l'alias et le mot de passe	
101P01EX01	Exit	Le système affiche un message d'erreur	
101P01EX02	Exit	Le système affiche la page personnelle du participant	
101TR02	Triggering Event	L'Administrateur demande à se brancher au systeme	
101P02	Functional process	L'Administrateur complete le formulaire de branchement	
101P02EN02	Entry	Entrée de l'Alias et du mot de passe	
101P02R01	Read	Le système valide l'alias et le mot de passe	
101P02EX01	Exit	Le système affiche la page personnelle du participant	
101P02EX02	Exit	Le système affiche un message d'erreur	
101P02W01	Write	Le système met à jour le nombre de tentative de branchement	1
101P02R02	Read	Le système consulte nombre de tentatives	2
101P02EX03	Exit	message d'erreur nombre de tentative maximal atteint	3
101P02EX04	Exit	Réinitialisation des champs	
102TR021	Triggering Event	Le participant sélectionne l'option abonnement	
102P01	Functional process	Le participant saisit les informations afin de s'abonner	
102P01EN01	Entry	le participant complète le formulaire d'abonnement	
102P01R01	Read	Le système valide les infos du data groupe ``participant``	
102P01R02	Read	Le système valide les infos du data groupe ``type user``	
102P01W01	Write	Ajouter participant	
102P01E01	Exit	Envoi courriel confirmation	
102P02EX01	Exit	Message d'erreur pour remplir les champs obligatoire	
102P03EN01	Entry	alias saisi	4
102P03R01	Read	Système accède aux alias choisis	5
102P03EX01	Exit	Message d'erreur indiquant alias déjà choisi	6
102P04EX01	Exit	Message d'erreur indiquant courriel non valide	7
103TR01	Triggering Event	L'administrateur sélectionne l'option d'activation d'un participant	
103P01	Functional process	Activer un participant	
103P01EN01	Entry	Formulaire d'activation d'un participant	
103P01R01	Read	Liste des alias des participants	
103P01W01	Write	Sauvegarder l'user comme activé	
103P01EX02	Exit	retourne l'user à la page d'administration	
107TR01	Triggering Event	le participant sélectionne l'option "Résultats et statistiques"	8
107P01	Functional process	visualiser résultats et statistiques	9
107P01EN01	Entry	sélectionner	10
107P01R01	Read	accès page formulaire des statistiques	11
107P01R02	Exit	Afichage formulaire statistiques	12
107TR02	Triggering Event	le participant sélectionne l'option "Classement de la semaine"	
107P02	Functional process	Classement de la semaine	
107P02EN01	Entry	Année et semaine saisis	
107P02R01	Read	Accès PF_PARTIE	
107P02R02	Read	Accès PF_EQUIPE	
107P02W01	Write	Sauvegarde sur PF_CHOIX	
107P02EX01	Exit	Affichage Classement de la semaine	
107TR02	Triggering Event	le participant sélectionne l'option "Classement général"	

107P02	Functional process	Classement général	
107P02EN01	Entry	Année et semaine saisis	
107P02R01	Read	Accès PF_PARTIE	
107P02R02	Read	Accès PF_EQUIPE	
107P02W01	Write	Sauvegarde sur PF_CHOIX	13
107P02EX01	Exit	Affichage Classement général	
107TR02	Triggering Event	le participant sélectionne l'option "meilleur performance"	
107P02	Functional process	meilleur performance	
107P02EN01	Entry	Année et semaine saisis	
107P02R01	Read	Accès PF_PARTIE	
107P02R02	Read	Accès PF_EQUIPE	
107P02W01	Write	Sauvegarde sur PF_CHOIX	14
107P02EX01	Exit	Affichage meilleur performance	
107TR02	Triggering Event	le participant sélectionne l'option "Statistiques par semaine"	
107P02	Functional process	Statistiques par semaine	
107P02EN01	Entry	Année et semaine saisis	
107P02R01	Read	Accès PF_PARTIE	
107P02R02	Read	Accès PF_EQUIPE	
107P02W01	Write	Sauvegarde sur PF_CHOIX	15
107P02EX01	Exit	Affichage Statistiques par semaine	
107TR02	Triggering Event	le participant sélectionne l'option "Historique par semaine"	
107P02	Functional process	Historique par semaine	
107P02EN01	Entry	Année et semaine saisis	
107P02R01	Read	Accès PF_PARTIE	
107P02R02	Read	Accès PF_EQUIPE	
107P02W01	Write	Sauvegarde sur PF_CHOIX	16
107P02EX01	Exit	Historique par semaine	
		N M = nombre de mauvaises réponses (en gris)	
		Chiffre répété 2 fois: même mauvaise réponse	

Mesureur 2

ID	Type of component	Complete Identification	N M
ucs101-7	Functional process	brancher au syteme comme administrateur	
ucs101-8	Triggering Event	brancher au syteme comme participant	
ucs101-3	Entry	entrer donnée	
ucs101-4	Read	comparer alias	
ucs101-5	Read	comparer mots de passe	1
ucs101-6	Functional process	afficher la page d'administrateur	
ucs101-7	Functional process	afficher la page de l'utilisateur	
ucs101-9	Exit	erreur de validation	
ucs101-11	Exit	refusé d'accès (erreur superieur a 4)	
usc102-1	Functional process	selectionner l'option abonnement	2
usc102-2	Triggering Event	afficher le formulaire d'abonnement	3
usc102-3	write	saisir des attributs	4
usc102-4	Exit	erreur sur le champ de saisie	5
usc102-5	Exit	erreur sur l'alias	6
usc102-6	Exit	erreur sur le courriel	7
usc102-7	Exit	format des champs	8
usc102-16	Functional process	validation des données saisies	9
usc102-17	Triggering Event	affichage des données de validation	10
usc102-18	Functional process	envoyer un couriel de confirmation aux participants	11
usc102-19	Triggering Event	courriel envoyer	12
usc102-20	Read	lire l'alias et mot de passe	
usc102-22	Read	lire le mot de passe	
usc102-23	write	ecrir l'alias et mot de passe	
usc103-1	Functional process	selectionner l'option d'activation	
usc103-2	Triggering Event	afficher la liste des alias	
usc103-3	Functional process	choix de l'utilisateur	
usc103-4	Triggering Event	afficher le formulaire d'activation	
usc103-5	write	enregistrer le changement	
usc103-6	Functional process	retour a la page de l'administrateur	
usc107-1	Functional process	selctionner resultat	
usc107-2	Triggering Event	afficher resultat	
usc107-3	Triggering Event	afficher formulaire	
usc107-4	Functional process	choix des resultat statistique	
usc107-5	Functional process	fermeture de l'horizon des résultat statistique	
usc107-6	Triggering Event	ouvrir la pagec personnel	
usc107-7	Exit	erreur aucun resultat disponible	
usc107-7	Exit	Message d'erreur (semaine non sélectionnée)	

Mesureur 3

ID	Type	Complete Identification	N M
UCS101_1	Triggering Event	Compléter les champs du formulaire de branchement	
UCS101_2	Triggering Event	le participant se brancher au système	
UCS101_3	Entry	Alias et Mot de passe : l'utilisateur complète le champ de formulaire de branchement	
UCS101_4	Functional process	le participant se brancher au système	
UCS101_5	Functional process	Le système affiche la page personnelle du participant.	
UCS101_6	Read	Le système accède au fichier des "mot de passe" (PF_PARTICIPANT). C'est implicite	
UCS101_7	Exit	Le système affiche la page personnelle du participant.	
UCS101_8	Exit	Affichage du message d'erreur.	
UCS101_8	Exit	Affichage du message : "Alias ou mot de passe invalide! ..."	1
UCS101_9	Write	Le système vide le contenu des champs du formulaire.	2
UCS101_10	Triggering Event	l'administrateur se brancher au système	
UCS101_11	Triggering Event	l'administrateur compléter les champs du formulaire de branchement	
UCS101_12	Entry	Alias et Mot de passe : l'utilisateur complète le champ de formulaire de branchement	
UCS101_13	Functional process	l'administrateur du pool se brancher au système	
UCS101_14	Read	Le système accède au fichier des "mot de passe" (PF_PARTICIPANT). C'est implicite	
UCS101_15	Exit	Le système affiche la page principale de l'administrateur	
UCS101_16	Exit	Affichage du message : "Alias ou mot de passe invalide! ..." (idem)	3
UCS101_16	Exit	Affichage du message d'erreur.	4
UCS101_17	Write	Le système vide le contenu des champs du formulaire.	
	Triggering Event	L'alias ou le mot de passe n'est pas valide	
	Entry	L'alias ou le mot de passe n'est pas valide	
	Functional process	taritement du cas ou le mot de passe ou l'alias n'est pas valide	
	Exit	Affichage du message : "Alias ou mot de passe invalide! ..." (idem)	4
		Le système vide le contenu des champs du formulaire.	
UCS101_9	Triggering Event	l'utilisateur à essayer de se brancher à pour une quatrième fois.	
UCS101_10	Entry	Données saisies non valide	
UCS101_9	Functional process	Vidage du contenu des champs du formulaire (par le système)	
UCS101_	Exit	Affichage du deuxième message.	
	Write	Le système vide le contenu du formulaire.	
UCS101_11	Functional process	Le système bloque la possibilité de branchement à l'utilisateur après la troisième tentative	5
UCS102_1	Functional process	Le participant accède au formulaire d'abonnement	
UCS102_2	Triggering Event	L'utilisateur sélectionne l'option d'abonnement	
UCS102_3	Entry	Saisi des informations dans les champs du formulaire	
UCS102_4			
UCS102_4	Exit	Le système envoie un courriel de confirmation d'abonnement.	6
UCS102_5	Exit	Le système affiche le formulaire d'abonnement.	7
UCS102_6	Exit	Le système fournit un accès à la page personnelle du participant.	8
UCS102_7	Exit	Affichage du message "Vous devez remplir tous les champs précédés d'une étoile"	9
		Affichage du message qui indique que l'alias choisi par le participant est déjà existant dans le système.	10
UCS102_8	Exit		
UCS102_9	Exit	Affichage du message : "L'adresse de courriel est invalide! ..."	11
UCS102_10	Exit	Affichage du message : "Le mot de passe doit être entre a et 10 caractères de long, ..."	12
UCS103_1	Functional process		
UCS103_1	Entry	L'administrateur sélectionne l'option d'activation d'un participant.	
	Entry		
UCS103_2	Triggering Event	L'administrateur sélectionne l'option d'activation d'un participant.	
UCS103_3	Functional process	Accès au formulaire d'activation d'un participant	13
UCS103_4	Exit	Affichage du formulaire d'activation d'un participant.	14

UCS103_5	Triggering Event	L'administrateur du pool sélectionne le participant à activer.	15
UCS103_6	Entry	Liste des alias présente dans le système.	16
UCS103_7	Functional process	L'administrateur active le participant.	
UCS103_8	Exit	Affichage des informations du participant dans le formulaire d'activation.	
UCS103_9		Affichage de la page d'administration du pool.	
UCS103_10	Write	Sauvegarde du changement	
UCS107_1	Functional process	Visualisation des résultats et statistiques.	
UCS107_2	Triggering Event	Le participant sélectionne l'option des résultats et statistiques.	
UCS107_3	Entry	Le participant sélectionne dans le formulaire, l'année et le mois.	
UCS107_4	Exit	Affichage u formulaire des résultats et statistiques.	
UCS107_5	Exit	Ouverture de la page personnelle du participant.	
UCS107_6	Exit	Affichage du message : "aucun résultat disponible dans le système pour la semaine sélectionnée".	

Mesureur 4

ID	Type of component	Complete Identification	N M
UCS101_1	Functional process	Se brancher au systeme	
UCS101_2	Triggering Event	Execution du systeme	
UCS101_3	Entry	Alias	
UCS101_4	Entry	Mot de passe	1
UCS101_5	Read	lire PF_PARTICIPANT	
UCS101_6	Exit	Page personnelle du participant	
UCS101_7	Exit	Message d'erreur 1	
UCS101_8	Exit	Message d'erreur 2	2
UCS101_9	Functional process	Se brancher au systeme	
UCS101_10	Triggering Event	Execution du systeme	
UCS101_11	Entry	Alias	
UCS101_12	Entry	Mot de passe	1
UCS101_13	Read	lire PF_PARTICIPANT	
UCS101_14	Exit	Page personnelle de l'administrateur	
UCS101_15	Exit	Message d'erreur 1	
UCS101_16	Exit	Message d'erreur 2	2
UCS102_1	Functional process	S'abonner au Pool	
UCS102_2	Triggering Event	Selectionner l'option d'abonnement	
UCS102_3	Entry	Nom	
UCS102_4	Entry	prenom	3
UCS102_5	Entry	courriel	4
UCS102_6	Entry	numero de telephone	5
UCS102_7	Entry	Alias	6
UCS102_8	Entry	Mot de passe	7
UCS102_9	Entry	adresse postale	8
UCS102_10	Entry	type de participation	9
UCS102_11	Read	lire PF_PARTICIPANT	
UCS102_12	Write	PF_PARTICIPANT	
UCS102_13	Write	PF_TYPE_USAGER	10
UCS102_14	Exit	courriel pour le participant	
UCS102_15	Exit	Message : tous les champs precedes de *	
UCS102_16	Exit	Message : alias deja existant	11
UCS102_17	Exit	Message : adresse courriel incorrecte	12
UCS102_18	Exit	Message : mot de passe incorrecte	13
UCS103_1	Functional process	Activer un participant	
UCS103_2	Triggering Event	Option activer un participant	
UCS103_3	Functional process	selectionner un participant parmi la liste affichee	14
UCS103_4	Read	lire PF_PARTICIPANT	
UCS103_5	Entry	no_participant	
UCS103_6	Write	PF_PARTICIPANT	
UCS103_7	Exit	informations participant sur le formulaire	15
UCS103_8	Functional process	Desactiver un participant	16
UCS103_9	Triggering Event	Option activer un participant	17
UCS103_10	Functional process	selectionner un participant parmi la liste affichee	18
UCS103_11	Read	lire PF_PARTICIPANT	19
UCS103_12	Entry	no_participant	20
UCS103_13	Write	PF_PARTICIPANT	21

UCS103_14	Exit	informations participant sur le formulaire	22
UCS103_15	Exit	liste des participants existants	23
UCS103_16	Functional process	selectionner un participant parmi la liste affichee	24
UCS103_17	Triggering Event	option activer participant	25
UCS103_18	Read	lire PF_PARTICIPANT	26
UCS103_19	Entry	no_participant	27
UCS103_20	Exit	liste des participants existants	28
UCS107_1	Functional process	visualiser les resultats et statistiques	29
UCS107_2	Triggering Event	option resultats et statistiques	30
UCS107_3	Entry	annee	31
UCS107_4	Entry	semaine	32
UCS107_5	Read	lire PF_PARTICIPANT	33
UCS107_6	Read	lire PF_RESULTAT_PARTICIPANT	34
UCS107_7	Exit	resultats (nb_victoire, nb_defaite)	35
UCS107_7'	Exit	Message : aucun resultat disponible	36
UCS107_8	Functional process	Afficher les resultats selon les options : de la semaine	
UCS107_9	Triggering Event	option : classement de la semaine	
UCS107_10	Entry	option : classement de la semaine	
UCS107_11	Read	lire PF_PARTIE	
UCS107_12	Exit	resultats de la semaine	
UCS107_13	Functional process	Afficher les resultats selon les options : classement general	
UCS107_14	Triggering Event	option : classement GENERAL	
UCS107_15	Entry	option : classement GENERAL	
UCS107_16	Read	lire PF_PARTICIPANT	
UCS107_17	Read	lire PF_PARTIE	
UCS107_18	Exit	resultats : CLASSEMENT GENERAL	
UCS107_19	Functional process	Afficher les resultats selon l'option : MEILLEURE PERFORMANCE	
UCS107_20	Triggering Event	option : MEILLEURE PERFORMANCE	
UCS107_21	Entry	option : MEILLEURE PERFORMANCE	
UCS107_22	Read	lire PF_PARTICIPANT	
UCS107_23	Read	lire PF_PARTIE	
UCS107_24	Exit	resultats :MEILLEURE PERFORMANCE	
UCS107_25	Functional process	Afficher les resultats selon l'option : STATISTIQUES PAR SEMAINE	
UCS107_26	Triggering Event	option : STATISTIQUES PAR SEMAINE	
UCS107_27	Entry	option : STATISTIQUES PAR SEMAINE	
UCS107_28	Read	lire PF_PARTICIPANT	
UCS107_29	Read	lire PF_PARTIE	
UCS107_30	Exit	resultats :STATISTIQUES PAR SEMAINE	
UCS107_31	Functional process	Afficher les resultats selon l'options : HISTORIQUE	
UCS107_32	Triggering Event	option : HISTORIQUE	
UCS107_33	Entry	option : HISTORIQUE	
UCS107_34	Read	lire PF_PARTICIPANT	
UCS107_35	Read	lire PF_ANNEE	
UCS107_36	Read	lire PF_PARTIE	
UCS107_37	Exit	resultats :HISTORIQUE	

Mesureur 5

ID	Type of component	Complete Identification	N M
USC101-1	Functional process	Se Brancher au système	
USC101-2	Boundary	Le clavier du client et la connection internet	1
USC101-3	Triggering Event	Le participant ou l'administrateur entre son identification et mot de passe et appuie sur enter ou clique sur le bouton branchement	2
USC101-4	Entry	Alias et mot de passe provenant du formulaire de branchement	
USC101-5	READ	Le systeme lit le mot de passe dans la table PF_PARTICIPANT et valide le mot de passe	
USC101-6	READ	Le systeme fait la sommation des connections dans le systeme.	3
USC101-7	Read	Le systeme lit le type de l'utilisateur (administrateur ou participant)	4
USC101-8	Write	MET NONBRE CONNEXION a 1	5
USC101-9	Exit	Message trop de connection.	6
USC101-10	Exit	Message Alias ou mot de passe invalide !	
USC101-11	Exit	Message Connexion accepté	
USC101-12	Exit	Le formulaire de branchement vide	
USC101-13	Exit	La page personnel du participant.	
USC101-14	Exit	La page principal de l'administrateur du pool	
USC102-1	Functional process	Afficher le formulaire d'abonnement	
USC102-2	Boundary	La souris du client et la connection internet	7
USC102-3	Triggering Event	L'utilisateur clique sur abonnement dans la page principal.	
USC102-4	Entry	Le clic sur le bouton abonnement	8
USC102-5	Read	Le contenu de la page web sur le serveur linux	9
USC102-6	Read	Lire les options d'utilisation entre par l'administrateur (USC104)	10
USC102-6	Exit	Afficher le formulaire d'abonnement a l'ecran du client	
USC102-7	Exit	Afficher un message que l'abonnement n'est plus disponible.	
USC102-8	Functional process	Saisir Information dans le formulaire d'abonnement	
USC102-9	Boundary	Le clavier et la souris du client et la connection internet	
USC102-10	Triggering Event	L'utilisateur clique sur confirmer dans le formulaire d'abonnement	
USC102-11	Entry	Le systeme lit les champs nom, prénom,courriel,alias,mot de passe et confirmer mot de passe dans le formulaire d'abonnement.	
USC102-12	Write	Ecrire dans la table PF_Participant les informations nom, prénom,courriel,alias,mot de passe	
USC102-13	Read	Le systeme lit les alias dans la table PF_Paticipant	11
USC102-14	Exit	Message qui nous incite a remplir les champs precedes d'une étoile dans le systeme et retour au formulaire d'abonnement.	12
USC102-15	Exit	Message qui indique que l'alias existe deja retour au formulaire d'abonnement	13
USC102-16	Exit	Message qui indique que le courriel ne contient pas d'arobas et retour au formulaire d'abonnement.	14
USC102-17	Exit	Message qui indique que l'alias ou le mot de passe n'est pas valide et retour au formulaire d'abonnement	15
USC102-18	Exit	Message de confirmation de l'abonnement et sortie du systeme.	16
USC103-1	Functional process	L'administrateur accede au formulaire d'activation d'un participant.	
USC103-2	Boundary	La souris du client et la connection internet (page web)	
USC103-3	Triggering Event	L'administrateur clique sur le lien HTML Activer un participant.	
USC103-4	Entry	Le clique sur le lien html dans la page principal de l'administrateur.	
USC103-5	Read	Lire le contenu de la page web Formulaire d'activation sur le serveur linux.	17
USC103-6	Exit	Afficher le Formulaire d'activation a travers le fureteur internet de l'administrateur.	18
USC103-7	Functional process	L'administrateur selectionne le participant à activer	19
	Boundary	La souris du client et la connection internet (page web)	7
USC103-8	Triggering Event	L'administrateur selectionne l'annee d'activation et l'alias du concurant à partir du liste concurante.	
USC103-9	Entry	La selection de l'administrateur pour le participant et l'année a travers la page web.	20
USC103-10	Read	Lire la list des alias pour l'annee choisi.	21
USC103-11	Read	Lire l'information Nom, PRENOM,Courrier électronique, téléphone dans la table PF_PARTICIPANT	22

USC103-12	Exit	Afficher cette information dans le formulaire d'activation de l'administrateur.	23
USC103-13	Functional process	L'administrateur active le participant.	24
USC103-14	Boundary	La souris du client et la connection internet (page web)	7
USC103-15	Triggering Event	L'administrateur active sur le bouton sauvegarder	25
USC103-16	Entry	La selection entre active et deactive	26
USC103-17	Write	Ecrire le bit actif dans la table	27
USC103-18	Exit	Retourne a la page d'administration du pool.	
USC107-1	Functional process	Le participant accède au formulaire des resultats et des statistique.	28
USC107-2	Boundary	La souris du client et la connection internet (page web)	7
USC107-3	Triggering Event	Le clique dans la page personnel du participant du lien html RESULTATS ET STATISTIQUE	29
USC107-4	Read	Le systeme lit le contenu de la page de resultats et statistique sur le serveur Linux (WEB)	30
USC107-5	Exit	La page des resultats et statistiques s'affiche dans le fureteur internet du participant.	31
USC107-6	Functional process	Le participant sélectionne les résultats et les statistiques a visualiser.	32
USC107-7	Boundary	La souris du client et la connection internet (page web)	7
USC107-8	Triggering Event	Le clique sur le lien html dans la page web des résultats et statistique.	33
USC107-9	Entry	La selection du type de resultats désirée par le participant.	34
USC107-10	Read	Lire les tables de resultats dans la base de donnees MySQL selon le format choisi.	35
USC107-11	Read	Lire les donnees de la page sur le serveur linux (WEB)	36
USC107-12	Exit	Afficher la page web des resultat dans le fureteur internet du client.	37
USC107-13	Functional process	Le participant quitte le formulaire des resultats et statistique.	38
USC107-14	Boundary	La souris du client et la connection internet (page web)	7
USC107-15	Triggering Event	Le clique sur le lien html dans la page web des résultats et statistique pour quitter.	39
USC107-16	Read	Lire la page personnel du participant dans le serveur web.	40
USC107-17	Exit	Afficher la page web des resultat dans le fureteur internet du client.	

Annexe M: Spécifications présentées aux mesureurs débutants

Commentaires sur la documentation fournie

Voici un exemple de documentation fournie aux mesureurs débutants. Il s'agit des deux premières étapes de la première spécification. Cette documentation contient des termes anglais que nous avons conservé tels quels. Nous avons aussi noté de nombreuses fautes de français que nous n'avons pas corrigées.

UCS101. Se brancher au système

Brève description

Ce cas d'utilisation décrit comment les participants et l'administrateur du « pool » se branchent au système pour accéder à leur page personnelle.

Flow of Events

Basic Flow

Étape 1 : **Le participant complète le formulaire de branchement**

Ce cas d'utilisation débute lorsque l'utilisateur, un participant ou l'administrateur du pool, complète les champs du formulaire de branchement et se branche au système.

Le formulaire contient les champs suivants:

 Alias

 Mot de passe

Étape 2 : **Le système valide les informations du formulaire**

Le système valide les informations (alias et mot de passe) du formulaire et affiche la page correspondant au profil de l'utilisateur. Pour un participant, le système affichera la page personnelle du participant, alors que pour l'administrateur du pool, il affichera la page principale de l'administration du pool.

Alternative Flows

A1. Alias ou mot de passe invalide

Étape 2 : **Le système valide les informations du formulaire**

Lors de la validation de l'alias et du mot de passe par le système, si l'une ou l'autre des données saisies n'est pas valide, c'est-à-dire que la combinaison des deux informations ne correspond à aucun utilisateur dans le système, alors le système affichera le message suivant dans le formulaire de branchement :

« Alias ou mot de passe invalide! Veuillez entrer à nouveau les paramètres, en prenant soin de vérifier la casse. »

Le système vide le contenu des champs du formulaire et le traitement reprend à l'étape 1.

A1. 3 essais de branchement infructueux

Étape 2 : **Le système valide les informations du formulaire**

Lors de la validation de l'alias et du mot de passe par le système, si l'utilisateur à essayer de se brancher à pour une quatrième fois, le système ne valide plus les informations du formulaire. Il affiche plutôt le message suivant :

« Alias ou mot de passe invalide! L'accès au système vous est refusé dans la session en cours. Vous devez fermer votre navigateur et accéder le site pour pouvoir essayer à nouveau. Si vous ne vous souvenez plus de votre alias ou de votre mot de passe, veuillez communiquer avec l'administrateur du Pool (<adresse du courriel de l'administrateur du Pool>). »

Le système vide le contenu des champs du formulaire et le traitement est arrêté.

Special Requirements

SR1. Limite du nombre d'essai de branchement à 3

Le système doit bloquer la possibilité de branchement à l'utilisateur après la troisième tentative consécutive de branchement dans la même session Internet.

SR2. Accès simultané de 20 utilisateurs

Le système doit permettre l'accès simultané à l'application jusqu'à concurrence de 21 utilisateurs, incluant l'administrateur du Pool.

Preconditions

Le participant est abonné

Pour pouvoir se brancher à l'application, un participant doit d'abord s'être abonné au système (UCS102).

Postconditions

Aucune.

Extension Points

Aucun.