



## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Améliorations de l'outil d'ingénierie des exigences GenSpec

Pire, Olivier; Pirmez, Nicolas

*Award date:*  
2007

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur  
Institut d'informatique.

**Année académique 2006-2007**

**Améliorations de l'outil  
d'Ingénierie des Exigences  
*GenSpec* :**

Glossaire, Gestion Multi-documents  
et Allocations.

*Olivier Pire et Nicolas Pirmez*

Mémoire présenté en vue de l'obtention du grade de  
Maître en Informatique.



# Résumé

**Résumé** L'ingénierie des exigences (IE) est une activité cruciale du processus de développement de logiciels. Elle permet de faire le lien entre les besoins du client et leurs réalisations. Pour prendre en charge cette activité, l'unité Conception-Automatismes à Hydro-Québec à Montréal a développé un outil de support à l'IE : *GenSpec*. Cet outil permet l'entrée des exigences dans une base de données, quelques vérifications de ces exigences et la génération de documents d'exigences [Buj05]. Notre travail consiste à développer un glossaire avancé et une gestion centralisée des exigences pour *GenSpec*. L'ajout de ces fonctionnalités permet respectivement de gérer l'ambiguïté lexicale des exigences, de centraliser les exigences d'un projet et d'allouer des exigences à des composants (sous-systèmes) établis par le concepteur. Ce mémoire reprend également un état de l'art de plusieurs outils commerciaux de support à l'IE et d'articles de recherche, un exemple illustrant notre solution et une évaluation de notre travail ainsi que de notre démarche.

**Mots clés** : Ingénierie des exigences, outil de support à l'ingénierie des exigences, ambiguïté, glossaire, gestion des exigences

**Abstract** Requirements engineering (RE) is a crucial activity in software development. RE links customers' needs and their realisation. The "Conception-Automatismes" unit of Hydro-Québec in Montreal developed an RE-tool called *GenSpec*. This tool supports requirements definition in a database, some verifications of these requirements and requirements documents generation [Buj05]. We developed an advanced glossary and a centralised requirements management functionality for *GenSpec*. It allows dealing with lexical ambiguity in requirements, centralizing project's requirements and allocating requirements to architectural components (subsystems). This dissertation also contains a state of the art of different commercial RE tools and research articles, an illustration of our solution, and an evaluation of our work and reasoning.

**Keywords** : Requirements engineering, Requirement engineering management tool, ambiguity, glossary, requirements management



# Avant-propos

Ce mémoire est le résultat d'un travail de 9 mois qui a été réalisé dans l'unité Conception-Automatismes à Hydro-Québec à Montréal (Canada) et à l'université de Namur (Belgique). À Montréal, nous avons travaillé sous la supervision de M. René Bujold au développement de nouvelles fonctionnalités pour l'outil de support à l'ingénierie des exigences *GenSpec*. Ce logiciel a reçu de l'office québécois de la langue française, le prix des mérites du français dans les technologies de l'information en mars 2007, dans la catégorie des applications logicielles de Grande organisation et Petite et Moyenne organisation.

Nous voudrions remercier :

Le Professeur Patrick Heymans de l'université de Namur, pour le temps qu'il nous a consacré, son aide et ses nombreux conseils. Nous voudrions également le remercier pour l'opportunité qu'il nous a offerte de réaliser notre stage à Hydro-Québec au Canada.

Le Docteur Raimundas Matulevičius, monsieur Jean-Roch Meurisse et le professeur Vincent Englebert de l'université de Namur, pour leur aide et leurs conseils.

L'ingénieur René Bujold d'Hydro-Québec, Montréal, pour son aide, ses nombreux conseils et sa supervision lors de notre stage. Nous le remercions également pour son aide lors de la rédaction de notre article (cf. Annexe K).

L'ingénieur Alain Sicard d'Hydro-Québec, Montréal, pour sa participation à notre développement.

L'ingénieur Marc Lacroix, chef de l'unité Conception-Automatismes d'Hydro-Québec (Division Équipement), Montréal, pour son aide et sa supervision lors de notre stage.

L'ingénieur Harold Ratté pour sa supervision et ses conseils.

Madame Claire Levesque, pour son aide lors de notre stage et pour ses nombreuses petites attentions quotidiennes.

L'entièreté de l'équipe Conception-Automatismes d'Hydro-Québec, Montréal, pour leur accueil très chaleureux, pour l'attention et l'aide qu'ils nous ont porté durant toute la durée de notre stage et pour nous avoir offert l'opportunité de découvrir le Québec.

Nos familles respectives, qui nous ont permis de réaliser ces études et ce stage à l'étranger. Merci à Celia Kaut et Roxane Draguet pour leur soutien. Merci également à Françoise Honnay et Olivia Pirmez pour la relecture de notre travail.



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Contexte général</b>	<b>5</b>
1.1 L'ingénierie des exigences . . . . .	5
1.1.1 Définition . . . . .	5
1.1.2 Activités . . . . .	7
1.1.3 Importance . . . . .	16
1.1.4 Limitations . . . . .	18
1.2 Les outils de support à l'ingénierie des exigences . . . . .	20
1.2.1 Définition . . . . .	20
1.2.2 Intérêts . . . . .	20
1.2.3 Exigences . . . . .	22
1.2.4 Faiblesses . . . . .	24
1.3 Résumé du chapitre . . . . .	26
<b>2 L'outil de support <i>GenSpec</i></b>	<b>27</b>
2.1 Description . . . . .	27
2.1.1 Généralités . . . . .	27
2.1.2 Usagers . . . . .	28
2.1.3 Fonctionnalités . . . . .	28
2.1.4 Exemple . . . . .	29
2.2 Choix et définition du cadre d'évaluation . . . . .	35
2.2.1 Choix du cadre d'évaluation . . . . .	35
2.2.2 Définition du cadre d'évaluation . . . . .	35
2.3 Application du cadre d'évaluation à <i>GenSpec</i> . . . . .	39
2.3.1 Application du cadre d'évaluation d'exigences fonctionnelles . . . . .	39
2.3.2 Application du cadre d'évaluation d'exigences non-fonctionnelles . . . . .	48
2.3.3 Synthèse de l'évaluation . . . . .	52
2.4 Résumé du chapitre . . . . .	54
<b>3 Problématique</b>	<b>55</b>
3.1 Ambiguïté . . . . .	55
3.1.1 Généralités . . . . .	55
3.1.2 Types d'ambiguïtés . . . . .	56
3.1.3 Origines de l'ambiguïté . . . . .	56
3.2 Gestion multi-documents d'exigences et regroupements d'exigences . . . . .	58
3.2.1 Gestion multi-documents d'exigences . . . . .	59
3.2.2 Regroupements distincts entre l'analyste et le concepteur . . . . .	60
3.3 Résumé du chapitre . . . . .	61
<b>4 État de l'art d'OSIE</b>	<b>63</b>
4.1 Outils . . . . .	63
4.1.1 Telelogic Doors 8.1 . . . . .	65
4.1.2 IRqA 3.5.0 . . . . .	68
4.1.3 REM 1.2.2 . . . . .	71



4.1.4	IBM Rational RequisitePro 7.0.0.0 . . . . .	72
4.1.5	SysML pour Artisan Studio 6.1.21 . . . . .	74
4.2	Synthèse . . . . .	76
4.2.1	Tableau récapitulatif concernant le traitement de l'ambiguïté . . . . .	76
4.2.2	Tableau récapitulatif concernant le regroupement des exigences . . . . .	77
4.3	Résumé du chapitre . . . . .	78
<b>5</b>	<b>Solutions apportées à <i>GenSpec</i></b>	<b>79</b>
5.1	Description du processus suivi . . . . .	80
5.2	Description détaillée des activités du processus suivi . . . . .	82
5.2.1	Élicitation des besoins . . . . .	82
5.2.2	Analyse de l'existant . . . . .	82
5.2.3	Modélisation des objectifs . . . . .	82
5.2.4	Cas d'utilisation . . . . .	86
5.2.5	État de l'art des OSIE . . . . .	86
5.2.6	Prototypes d'interfaces graphiques . . . . .	86
5.2.7	Étude . . . . .	88
5.2.8	Rédaction des exigences . . . . .	89
5.2.9	Conception . . . . .	89
5.2.10	Découpage en incréments et itérations . . . . .	99
5.2.11	Implémentation . . . . .	100
5.2.12	Vérification des exigences (tests) . . . . .	100
5.2.13	Validation de l'incrément ou de l'itération (tests d'acceptation) . . . . .	100
5.2.14	Maintenance des exigences et de la conception . . . . .	101
5.3	Problèmes rencontrés . . . . .	101
5.3.1	Incohérence entre les acteurs du projet . . . . .	101
5.3.2	Connaissance tacite . . . . .	101
5.3.3	Documentation de la base de données . . . . .	101
5.4	Résumé du chapitre . . . . .	102
<b>6</b>	<b>Description technique par un exemple illustré</b>	<b>103</b>
6.1	Exemple illustré . . . . .	103
6.1.1	Définition de termes dans le glossaire . . . . .	104
6.1.2	Gestion multi-documents des exigences . . . . .	108
6.1.3	Conception du système avec l'AECF . . . . .	110
6.2	Intégration avec l'existant . . . . .	112
6.3	Résumé du chapitre . . . . .	113
<b>7</b>	<b>Évaluation de l'approche</b>	<b>115</b>
7.1	Évaluation de la démarche . . . . .	115
7.1.1	Avantages . . . . .	115
7.1.2	Limites . . . . .	116
7.2	Évaluation de l'implémentation . . . . .	117
7.2.1	Description des aspects non implémentés des fonctionnalités . . . . .	117
7.2.2	Avantages . . . . .	118
7.2.3	Limites . . . . .	118
7.3	Résumé du chapitre . . . . .	120
	<b>Conclusion</b>	<b>122</b>
	<b>Bibliographie</b>	<b>125</b>
	<b>Description des annexes</b>	<b>132</b>
	<b>A Cas d'utilisation : Glossaire</b>	<b>135</b>
	<b>B Cas d'utilisation : GMDE et AECF</b>	<b>159</b>

---

<b>C Etude : glossaire</b>	<b>193</b>
<b>D Etude : GMDE et AECF</b>	<b>215</b>
<b>E Exigences : Glossaire</b>	<b>237</b>
<b>F Exigences : GMDE et AECF</b>	<b>263</b>
<b>G Conception : Glossaire</b>	<b>297</b>
<b>H Conception : GMDE et AECF</b>	<b>325</b>
<b>I Sources linguistiques d'ambiguïté</b>	<b>359</b>
<b>J Proposition de schéma conceptuel</b>	<b>373</b>
<b>K Article</b>	<b>377</b>
<b>L Framework</b>	<b>389</b>



# Table des figures

1.1	<i>Portée de la vérification et de la validation [Eas05]. . . . .</i>	10
1.2	<i>La conception change la situation du problème [Eas05]. . . . .</i>	11
1.3	<i>Entrelacement du problème et de la solution [Eas05]. . . . .</i>	11
1.4	<i>Exemple de hiérarchie de système [Dor99]. . . . .</i>	12
1.5	<i>Exemple d'allocation d'exigences système [Dor99]. . . . .</i>	13
1.6	<i>Exemple de flowdown d'exigences système [Dor99]. . . . .</i>	14
1.7	<i>Définition d'interfaces [Dor99]. . . . .</i>	15
1.8	<i>Coût d'un changement tout au long d'un processus de développement [Sut02].</i>	16
1.9	<i>Evolution d'une exigence selon les 3 dimensions de l'IE [Poh93]. . . . .</i>	19
1.10	<i>Modèle de développement en cascade [Wik07d]. . . . .</i>	21
1.11	<i>Spécification d'exigences basée sur un repository [Fir03]. . . . .</i>	22
2.1	<i>Création d'un projet GenSpec. . . . .</i>	29
2.2	<i>Ouverture d'un projet GenSpec. . . . .</i>	30
2.3	<i>Fenêtre de rédaction des exigences. . . . .</i>	30
2.4	<i>Les liaisons d'une exigence à d'autres exigences. . . . .</i>	31
2.5	<i>Procédure d'essai relative à une exigence. . . . .</i>	32
2.6	<i>Groupes d'exigences à générer. . . . .</i>	32
2.7	<i>Ensemble des vérifications applicables aux exigences. . . . .</i>	33
2.8	<i>Options de génération. . . . .</i>	34
2.9	<i>Cadre d'évaluation d'exigences fonctionnelles pour un OSIE. . . . .</i>	36
2.10	<i>Cadre d'évaluation d'exigences non-fonctionnelles pour un OSIE. . . . .</i>	37
3.1	<i>Hiérarchie de composants établie par le concepteur. . . . .</i>	58
3.2	<i>Relation entre un projet et ses projets GenSpec. . . . .</i>	59
3.3	<i>Exemple de tableau d'allocation (Application) à des composants et fonctions établis manuellement (HQ). . . . .</i>	60
4.1	<i>Liste des mots alternatifs. . . . .</i>	65
4.2	<i>Proposition de remplacement d'un terme. . . . .</i>	66
4.3	<i>Organisation des dossiers, projets et modules [Tel]. . . . .</i>	67
4.4	<i>Proposition de relations [IRq]. . . . .</i>	69
4.5	<i>Diagramme des concepts [IRq]. . . . .</i>	69
4.6	<i>Document d'exigences dans REM. . . . .</i>	71
4.7	<i>Discussions relatives à un terme dans RequisitePro. . . . .</i>	72
4.8	<i>Exemple de caractéristiques des termes. . . . .</i>	73
4.9	<i>Dictionnaire d'objets. . . . .</i>	75
4.10	<i>Matrice de traçabilité [Tel]. . . . .</i>	78
5.1	<i>Processus suivi pour établir la solution. . . . .</i>	81
5.2	<i>Diagramme d'objectifs <math>i^*</math> de la fonctionnalité de glossaire. . . . .</i>	84
5.3	<i>Diagramme <math>i^*</math> associé à la fonctionnalité de GMDE et à l'AECF. . . . .</i>	85
5.4	<i>Diagramme de cas d'utilisation associé au glossaire. . . . .</i>	87
5.5	<i>Prototypage d'interface pour AECF (cf. annexe D). . . . .</i>	88

5.6	<i>Schéma conceptuel entités-associations avant l'intégration des nouvelles fonctionnalités.</i>	90
5.7	<i>Schéma conceptuel entités-associations présentant la fonctionnalité du glossaire.</i>	91
5.8	<i>Schéma conceptuel entités-associations présentant la GMDE.</i>	92
5.9	<i>Schématisation des regroupement d'exigences, des composants et fonctions.</i>	93
5.10	<i>Structure autorisée et interdite pour les composants.</i>	94
5.11	<i>Schéma logique présentant la GMDE.</i>	94
5.12	<i>Schéma conceptuel entités-associations de GenSpec après l'intégration des nouvelles fonctionnalités.</i>	96
5.13	<i>Diagramme de composants associé aux fonctionnalités de glossaire et de GMDE.</i>	98
5.14	<i>Diagramme de classes du glossaire et de la GMDE et l'AECEF.</i>	99
6.1	<i>Fenêtre du glossaire illustrant le terme "rapporteur".</i>	104
6.2	<i>Niveau d'ambiguïté du terme "session".</i>	105
6.3	<i>Répercussion du remplacement du terme "session parallèle" par son synonyme "session simultanée" dans les exigences.</i>	106
6.4	<i>Détection du synonyme "session simultanée" par le terme "session parallèle" dans une exigence.</i>	106
6.5	<i>Exigence accompagnée de définitions dans un document d'exigences.</i>	107
6.6	<i>Arbre des exigences de GenSpec.</i>	108
6.7	<i>Fenêtre de génération Microsoft Word.</i>	109
6.8	<i>Exemple de composants du système.</i>	110
6.9	<i>Tableau de conception.</i>	111
7.1	<i>Alternative au schéma de BD du glossaire.</i>	119





# Liste des tableaux

2.1	Dimension de la représentation. . . . .	39
2.1	Dimension de la représentation. . . . .	40
2.1	Dimension de la représentation. . . . .	41
2.2	Dimension de l'accord. . . . .	42
2.2	Dimension de l'accord. . . . .	43
2.2	Dimension de l'accord. . . . .	44
2.3	Dimension de spécification. . . . .	45
2.3	Dimension de spécification. . . . .	46
2.3	Dimension de spécification. . . . .	47
2.4	Exigences du processus. . . . .	48
2.5	Exigences du produit. . . . .	49
2.6	Exigences externes organisationnelles. . . . .	50
2.7	Exigences externes non-fonctionnelles pour les groupes Business. . . . .	51
2.8	Tableau récapitulatif relatif à l'évaluation par les exigences fonctionnelles. . .	53
2.9	Tableau récapitulatif relatif à l'évaluation par les exigences non-fonctionnelles.	54
4.1	Tableau récapitulatif relatif à la problématique du traitement de l'ambiguïté.	76
4.2	Tableau récapitulatif relatif à la problématique des regroupements d'exigences.	77





# Glossaire

<b>Aspect de fonctionnalité</b>	Sous-ensemble d'exigences d'une fonctionnalité constituant une partie de la fonctionnalité.
<b>Brainstorming</b>	Terme anglais référant à une technique permettant d'apporter des solutions à un problème donné grâce à un recoupement d'idées effectué par le groupe de travail [wik07a].
<b>Cadre d'évaluation</b> ( <i>Framework</i> )	Méthode d'évaluation permettant d'évaluer, d'analyser les outils de support à l'ingénierie des exigences, de définir des catégories et des exigences spécifiques (ou des activités) pour chacune de ces catégories, dans lesquelles des mesures (métriques) spécifiques sont utilisées [Mat05].
<b>Cas d'utilisation</b> ( <i>Use Case</i> )	Représente le comportement affiché par le système sous certaines conditions de manière à satisfaire un objectif de l'un des acteurs [Coc00].
<b>Client</b>	Acteur directement concerné par le développement des nouvelles fonctionnalités que nous avons développées. Dans le cadre du développement des nouvelles fonctionnalités, les clients sont : <ul style="list-style-type: none"><li>– le <b>client 1</b>, correspondant à <i>M. René Bujold</i>, chargé de l'analyse de projets à l'unité Conception-Automatismes à Hydro-Québec et responsable du développement des nouvelles fonctionnalités de <i>GenSpec</i> ;</li><li>– le <b>client 2</b>, correspondant à <i>M. Alain Sicard</i>, chargé de la conception à l'unité Conception-Automatismes à Hydro-Québec.</li></ul>
<b>Composant</b>	Regroupement d'allocations d'exigences, réalisé pour l'activité de conception, correspondant à un sous-système du système à développer.
<b>ConOps</b>	<i>Concept of Operations (ConOps) Document</i> . Un <i>ConOps</i> est un document orienté utilisateur qui décrit les caractéristiques d'un système du point de vue de l'utilisateur. Le document est utilisé pour communiquer toute caractéristique quantitative et qualitative du système à l'utilisateur, l'acheteur, le développeur et aux autres éléments organisationnels. Il est utilisé pour décrire l'organisation de l'utilisateur, les missions et les objectifs organisationnels du système intégré [IEE98].
<b>Diagramme de Gantt</b>	Outil utilisé en ordonnancement et en gestion de projet permettant de visualiser dans le temps les diverses tâches d'un projet. Il permet de représenter graphiquement l'avancement du projet [Wik07b].
<b>Document d'exigences</b>	Terme français employé à Hydro-Québec pour <i>Software Requirements Specification (SRS)</i> . Appelé également document de spécification.

<b>Domaine d'application</b>	Environnement dans lequel les actions du système à construire vont être observées et évaluées. Les exigences sont écrites relativement au domaine d'application [Jac98].
<b>Effet Hawthorne</b>	Décrit la situation dans laquelle les résultats d'une expérience ne sont pas dus aux facteurs expérimentaux mais au fait que les sujets ont conscience de participer à une expérience dans laquelle ils sont testés, ce qui se traduit généralement par une plus grande motivation [Wik07c].
<b>Élicitation</b>	Terme provenant de l'anglais ("Elicitation") faisant référence aux activités de collecte d'informations visant à identifier et clarifier le problème à l'origine du développement.
<b>Exigence dérivée</b>	Exigence provenant de l'activité de flowdown. Suite à l'allocation d'une exigence à un sous-système, cette exigence doit être adaptée selon les caractéristiques propres du sous-système. D'une seule exigence peut alors découler une ou plusieurs exigences. Ces exigences adaptées sont appelées dérivées.
<b>Exigence "nice-to-have"</b>	Exigence qui ne doit pas nécessairement être incluse au projet mais qui fournit une certaine valeur ajoutée à ce projet.
<b>Extrant (<i>Output</i>)</b>	Résultat d'une fonction, non nécessairement final. Par exemple, la fonction qui calcule l'aire d'un demi-cercle pourrait fournir en extrant l'aire du cercle, qu'elle calcule avant de la diviser par deux [Hyd07b].
<b>Flowdown</b>	Spécialisation des exigences selon le(s) sous-système(s) chargé(s) de les mettre en oeuvre.
<b>Fonction</b>	Regroupement d'allocations d'exigences, réalisé pour l'activité de conception, déterminant une des fonctions du système à développer.
<b>Fonctionnalité</b>	Dans le cadre de ce mémoire, le projet à réaliser se décompose en deux fonctionnalités : <ol style="list-style-type: none"><li>1. un glossaire ;</li><li>2. un support d'une gestion multi-documents d'exigences et l'allocation des exigences à des composants et fonctions.</li></ol>
<b>Gestion manuelle des exigences</b>	Expression référant à la réalisation des activités d'ingénierie des exigences sans aucun outil de support spécifique à l'ingénierie des exigences, tout au plus un traitement de texte ou un tableur.
<b>Groupe d'exigences générées</b>	Ensemble d'exigences déterminé par l'utilisateur de <i>GenSpec</i> qui contient les exigences à inclure dans les documents à générer.
<b>Ingénierie des exigences</b>	Ensemble d'activités importantes qui s'intègrent tout au long d'un processus de développement logiciel. Elle a pour but de définir ce que le système doit faire (le <i>quoi</i> ) plutôt que de définir <i>comment</i> il doit le faire.

---

<b>Ingénierie Système</b>	L'Ingénierie Système (ou ingénierie de systèmes) est une démarche méthodologique générale qui englobe l'ensemble des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes [def04].
<b>Intrant</b> ( <i>Input</i> )	Paramètre d'une fonction, élément utilisé par une fonction. Par exemple, les grandeurs des deux côtés d'un rectangle sont des intrants de la fonction qui calcule l'aire de ce rectangle [Hyd07b].
<b>Langage naturel</b>	Langage utilisé pour la communication entre êtres humains. Par exemple la français ou l'anglais.
<b>Modèle</b>	Représentation d'un canevas d'exigences (IEEE 830, Volere, etc.) dans <i>GenSpec</i> , se présentant comme un ensemble d'exigences structuré respectant un canevas pré-défini. Dans <i>GenSpec</i> , toute création d'un nouveau projet <i>GenSpec</i> se fait à partir d'un modèle.
<b>Outils CASE</b> ( <i>Computer Aided Software Engineering</i> )	Ensemble de logiciels organisés autour d'une base de données de spécifications (repository). Leur objectif est de fournir une assistance au développement et à la maintenance de logiciels ou de certains composants d'un logiciel [ddBdD03].
<b>Outils de support à l'ingénierie des exigences</b>	Outils logiciels fournissant une assistance automatisée durant le processus d'ingénierie des exigences [Mat05].
<b>Parties prenantes</b>	Ensemble des acteurs associés à un projet. Il s'agit des clients, utilisateurs, concepteurs, développeurs, managers et toute autre personne concernée de près ou de loin par l'utilisation ou le développement du système.
<b>Regroupement d'exigences</b>	Ensemble hiérarchisé d'exigences établies par un analyste. Typiquement, les exigences d'un projet chez Hydro-Québec sont découpées en plusieurs regroupements.
<b>Repository</b>	Ensemble d'informations (représentées généralement par des bases de données) permettant de stocker et de maintenir tout type d'informations.
<b>Système</b>	Un système ne se limite pas à un système informatique ( <i>software</i> et <i>hardware</i> ), mais inclut également les activités humaines. C'est par le support de celles-ci que le système est utile, on parle alors de " <i>Software-Intensive System</i> ". Le terme système correspond, par concision, dans ce mémoire à " <i>Software-Intensive System</i> ".
<b>Terme</b>	Mot désignant un concept propre à un domaine d'application. Dans le cadre de la solution implémentée, un terme est un mot ou une expression faisant partie du glossaire de <i>GenSpec</i> .

---



# Acronymes

<b>AECF</b>	Allocation des exigences à une hiérarchie de composants et une liste de fonctions.
<b>BD</b>	Base de données.
<b>GMDE</b>	Gestion multi-documents d'exigences.
<b>HQ</b>	Hydro-Québec.
<b>IE</b>	Ingénierie des exigences.
<b>LN</b>	Langage naturel.
<b>OSIE</b>	Outils de support à l'ingénierie des exigences.
<b>RE</b>	Requirements Engineering (Traduction anglaise d' <i>ingénierie des exigences</i> ).
<b>SI</b>	Système d'information.
<b>TE</b>	Type d'entité.
<b>UML</b>	Unified Modelling Language.



# Introduction

L'ingénierie des exigences (IE) est un ensemble d'activités essentielles dans le développement de logiciels. Toute erreur ou incompréhension lors de l'établissement des exigences peut entraîner l'échec ou la sous-utilisation du système développé. Une gestion efficace pour l'établissement, la gestion et le suivi des exigences est nécessaire.

L'unité Conception-Automatismes à Hydro-Québec (HQ) de Montréal développe, sur base de normes internationales (IEEE 830 et ISO/IEC 12207), son propre outils de support à l'IE (OSIE) : “*GenSpec*”. Cet outil est développé en réponse à la complexité d'utilisation des OSIE commerciaux actuels (*Doors*, *IRqA*, etc.).

*GenSpec*, comme beaucoup d'autres OSIE, permet la rédaction des exigences en langage naturel (LN). Aujourd'hui encore, pour de nombreux projets, les exigences sont établies en LN sans utiliser d'autres formalismes tel qu'UML. Le LN présente l'avantage de ne nécessiter aucune formation spécifique ni connaissance particulière. Cependant, son utilisation pour la rédaction des exigences amène inévitablement de l'ambiguïté. Par exemple, l'utilisation du terme “*avocat*” peut référer au fruit ou à la fonction. Les OSIE ne permettent généralement pas de traiter l'ambiguïté. Quelques OSIE présentent des mécanismes basiques de traitement de l'ambiguïté, se limitant souvent à un simple glossaire où une définition est associée à un terme.

La gestion centralisée de l'ensemble des exigences est importante. Différents types de liens doivent pouvoir être établis à partir d'une exigence afin de connaître son origine, l'impact de sa modification sur les exigences liées, le lien avec la conception, son implémentation, etc. *GenSpec* permet l'établissement de tels liens. Cependant, cette gestion est limitée à un seul regroupement d'exigences.

Le modèle de développement d'un logiciel en cascade présente une phase d'IE suivie d'une phase de conception. Ce découpage théorique est difficile à maintenir en pratique, les exigences étant modifiées en permanence. Lors de la conception architecturale d'un projet en sous-systèmes, la nécessité de nouvelles exigences ainsi que le raffinement de certaines exigences peuvent apparaître. Un besoin d'allocation des exigences aux différents sous-systèmes apparaît alors, nécessitant de faire le lien entre l'IE et l'activité de conception.

Par le biais de la collecte, de l'étude, de la conception et de l'implémentation des besoins, nous avons établi des solutions pour les problèmes de *GenSpec*.

Un glossaire, gérant un ensemble d'informations concernant un terme, a été développé. Par un traitement efficace des synonymes, il répond à la problématique de l'ambiguïté lexicale (plusieurs sens associés à un terme). Un mécanisme de détection de termes subjectifs a également été développé.

Une gestion de multiples regroupements d'exigences, appelé gestion multi-documents d'exigences, s'occupe de la gestion générale de l'ensemble des exigences d'un projet dans *GenSpec*. Par le rassemblement de l'ensemble des exigences d'un projet, les liens entre exigences sont gérés automatiquement.

Un mécanisme d'allocation d'exigences permet de réaliser une découpe architecturale du système en sous-systèmes, appelés composants. Un autre mécanisme d'allocation permet également d'établir les fonctions du système.



Ce mémoire, découpé en 7 chapitres, présente la totalité de notre recherche et de notre méthodologie pour arriver à ces solutions.

Le **chapitre 1** introduit la notion d'IE, définit les activités, l'importance et les limitations de l'IE. Nous y retrouvons également une définition des outils de support à l'IE, une description des intérêts et des exigences de ces outils ainsi que les limites d'un tel support.

Le **chapitre 2** fournit une description et une évaluation, selon le cadre d'évaluation de Matulevičius [Mat05], de l'OSIE *GenSpec*.

Le **chapitre 3** identifie les problèmes rencontrés par l'unité Conception-Automatismes à HQ pour gérer l'ambiguïté lors de la rédaction des exigences, pour gérer les regroupements d'exigences distincts et pour réaliser la conception architecturale du système à développer.

Le **chapitre 4** présente un état de l'art de cinq OSIE commerciaux ainsi qu'une synthèse issue de cette analyse et de la recherche dédiée aux problématiques définies au chapitre 3.

Le **chapitre 5** décrit la solution que nous avons apporté à *GenSpec* ainsi que notre démarche.

Le **chapitre 6** fournit une description de la solution que nous avons apportée par le biais d'un exemple appliqué. Une description de l'intégration des nouvelles fonctionnalités avec l'environnement existant est également fournie.

Le **chapitre 7** évalue notre approche, notre implémentation et décrit les exigences rédigées mais non implémentées.

À l'issue de ces chapitres, nous retrouvons la conclusion de notre travail comportant une description des perspectives de développements futurs, relatifs à nos solutions et à *GenSpec* dans sa totalité.





# Chapitre 1

## Contexte général

Ce chapitre fournit une vue générale du domaine de l'ingénierie des exigences ainsi que des outils de support à l'ingénierie des exigences.

### 1.1 L'ingénierie des exigences

Dans cette section, nous allons :

1. définir ce qu'est l'ingénierie des exigences (1.1.1) ;
2. fournir une liste des activités associées à l'ingénierie des exigences (1.1.2) ;
3. discuter de l'importance de l'ingénierie des exigences (1.1.3) ;
4. établir une liste des problèmes liés à l'ingénierie des exigences (1.1.4).

Notons que cette section est largement inspirée de Easterbrook [Eas05] et de Dorfman [Dor99].

#### 1.1.1 Définition

Il paraît clair aujourd'hui que l'**ingénierie des exigences (IE)** constitue un ensemble d'activités importantes qui s'intègrent tout au long d'un processus de développement logiciel. Elle a pour but de définir ce que le système doit faire (le *quoi*) plutôt que de définir *comment* il doit le faire. Boehm, repris par [Sut02], nous dit que :

*“RE is designing the right thing as opposed to software engineering, which is designing the thing right.”*

Easterbrook [Eas05] définit l'IE comme suit :

*“Requirements Engineering (RE) is a **set of activities** concerned with **identifying and communicating the purpose** of a software-intensive system, and the **context** in which it will be used. Hence, RE acts as the bridge between the real world needs of users, customers, and other **constituencies** affected by software system, and the **capabilities and opportunities** afforded by software-intensive technologies.”*

“*Software-Intensive Systems*”, employé dans cette définition, souligne qu'un système ne se limite pas à un software<sup>1</sup> ou à un système informatique (*software* et *hardware*), mais

---

<sup>1</sup>Nous décidons d'employer les termes anglais “*software*” et “*hardware*” afin de marquer la distinction entre les aspects logiciels et matériels d'un système informatique.

inclut également les activités humaines<sup>2</sup>. Cependant, le terme “système” sera utilisé dans tout le reste du document par concision et simplicité.

La définition d'Easterbrook met en évidence que :

- l'IE est un **ensemble d'activités** (*set of activities*) qui se déroule de manière constante tout au long d'un projet. On ne peut limiter l'IE à une phase ou une étape dans un processus de développement logiciel, les exigences étant soumises à des changements constants.
- la **communication** (*identifying and communicating*) en IE est aussi importante que l'analyse.
- la **définition de l'objectif** (*purpose*) du développement est essentielle. Si l'objectif du projet n'est pas ou mal défini, on ne peut rien affirmer en matière de qualité du projet. La qualité fondamentale d'un projet correspondant à la réponse du projet à l'objectif.
- les concepteurs doivent avoir **connaissance du contexte** (*context*) dans lequel le système sera employé, c'est-à-dire dans quelles conditions et comment il sera employé (usagers, procédures organisationnelles, etc.).
- l'ensemble des **partie prenantes**<sup>3</sup> (*constituencies*) doit être identifié. Les intervenants ne se limitent pas uniquement aux clients et usagers du système en cours de développement. Il est nécessaire de considérer la totalité des intervenants, concernés de près ou de loin par l'utilisation ou le développement du nouveau système afin de prendre en compte leurs différents besoins et d'accroître l'acceptation du système en développement.
- les **capacités et opportunités** (*capabilities and opportunities*) doivent être connues pour établir, en fonction des différentes contraintes matérielles et logicielles, ce qu'il est possible de réaliser et ce qui ne l'est pas. Par exemple, les limites en matière de calculabilité de certaines problématiques, etc.

---

<sup>2</sup>Remarquons qu'un nouveau système supportant une activité humaine est susceptible de la modifier de manière significative.

<sup>3</sup>Ensemble des acteurs associés à un projet. Il s'agit des clients, usagers, concepteurs, développeurs, managers et toute autre personne concernée de près ou de loin par l'utilisation ou le développement du système.

### 1.1.2 Activités de l'IE

Cette sous-section présente un ensemble d'activités correspondant à la découverte, l'établissement et la gestion des exigences. Ces activités ne sont pas nécessairement réalisées séquentiellement et peuvent être répétées autant de fois que nécessaire.

Les activités essentielles de l'IE sont :

- l'élicitation<sup>4</sup> ;
- la priorisation<sup>5</sup> ;
- la négociation ;
- la documentation (et la modélisation) ;
- la validation et la vérification.

**Élicitation** L'activité d'élicitation apparaît lorsque la notion d'un problème nécessitant une solution survient. Le but est alors de collecter assez d'informations pour identifier le problème et devenir un expert du domaine.

L'élicitation nécessite l'identification et la consultation des différentes parties prenantes du projet lors de l'acquisition des informations.

L'activité d'élicitation répond à quatre besoins essentiels [Eas05]. À titre d'exemple, nous utilisons un système de gestion de rendez-vous<sup>6</sup> :

1. **l'identification du problème** : permet de définir l'objectif du projet, la "vision" des demandeurs.  
*Exemple* : "La planification des rendez-vous nous coûte trop cher."
2. **les limites du problème** : selon la vision des demandeurs, le problème est délimité.  
*Exemple* :  
- "Construire un système qui planifie les rendez-vous" ou  
- "Construire un système qui maintient les calendriers des personnes"
3. **l'identification des scénarios de solution** : selon le problème, on définit quel processus business est le plus adéquat.  
*Exemple* : "Toute personne qui veut planifier un rendez-vous va chez la secrétaire, donne les détails et la secrétaire s'occupe du reste".
4. **la portée de la solution** : selon un processus business donné, on définit quels sont (et de quelle manière) les éléments qui devraient être automatisés.  
*Exemple* :  
- "Le système prend en entrée les détails des requêtes de planification et fournit une solution en sortie".  
- "La solution provient de l'interaction entre la secrétaire et le système".

Cependant, l'activité d'élicitation présente diverses difficultés [Eas05, Sut02] :

1. **une grande diffusion de la connaissance du domaine** : la connaissance peut être distribuée en de nombreuses sources. De plus, il y aura généralement des conflits entre la connaissance de ces différentes sources.
2. **la connaissance tacite** : toute personne familière avec un domaine rencontre des difficultés pour exprimer la connaissance qu'elle emploie quotidiennement inconsciemment. De cette façon, certains éléments majeurs peuvent être omis par l'utilisateur.  
*Exemple* : lors de la conduite d'une voiture, à moins d'être novice, un conducteur

<sup>4</sup>Provient du terme anglais "Elicitation". Nous l'employons afin de caractériser l'"extraction des exigences".

<sup>5</sup>Provient du terme anglais *Prioritization*. Correspond à l'établissement de priorités dans l'implémentation des exigences.

<sup>6</sup>En anglais : "Meeting scheduler".

ne réfléchit pas au changement de vitesse ni au chemin qu'il emprunte quotidiennement. Ce sont des compétences acquises pour lesquelles le conducteur rencontrerait des difficultés s'il devait les exprimer.

3. **l'observabilité limitée** : les clients ne sont pas toujours libres de s'occuper du système actuel afin de déterminer la réelle problématique. La présence d'un observateur, pour comprendre le problème, peut entraîner un changement de la définition du problème (Hawthorne Effect<sup>7</sup>).
4. **les biais** : une personne n'est pas toujours disponible ou ne veut pas toujours exprimer ce que l'analyste devrait savoir. Une pression extérieure, la crainte de perdre leur emploi par l'arrivée du nouveau système, etc. entraîne des "mensonges" et "omissions" de certaines parties prenantes.
5. **l'ambiguïté** : toute partie prenante essayant de reporter ce qu'elle fait, le fait généralement de manière imprécise. Les langages naturels, tel que le français ou l'anglais, sont de merveilleux moyens pour exprimer des idées. Malheureusement, les expressions produites par ces moyens ne sont pas toujours précises [Sut02].

**Prioritisation** La prioritisation permet de sélectionner les exigences et l'ordre de leur implémentation.

Les clients énoncent en général beaucoup d'exigences qu'il est nécessaire de trier :

- les exigences qui *doivent* être incluses ;
- les exigences qui devraient être exclues ;
- les exigences "nice-to-have"<sup>8</sup>.

Pour effectuer ce tri, il est important de connaître pour chaque exigence :

- son importance pour le client ;
- le coût de son implémentation, en terme de temps et d'argent, etc. ;
- le risque lié à son implémentation.

**Négociation** La négociation permet de répondre aux conflits éventuels survenant entre individus, groupes, organisations ou différents rôles joués par une personne. En IE, un conflit se base essentiellement sur une incohérence logique. Van Lamsweerde et al., repris par Eastbrook [Eas05], nous disent :

*"[A] conflict is a divergence between goals – there is a feasible boundary condition that makes the goals inconsistent".*

La négociation est un travail collaboratif. Les parties prenantes tentent de trouver un accord qui satisfait autant que possible l'ensemble des parties.

---

<sup>7</sup>Situation dans laquelle les résultats d'une expérience ne sont pas dus aux facteurs expérimentaux mais au fait que les sujets ont conscience de participer à une expérience dans laquelle ils sont testés, ce qui se traduit généralement par une plus grande motivation [Wik07c].

<sup>8</sup>Ce sont des exigences qui ne doivent pas nécessairement être incluses au projet mais qui fournissent une certaine valeur ajoutée au projet.

**Documentation** La documentation permet la réalisation d'un document d'exigences<sup>9</sup>. Ce document reprend l'ensemble des exigences établi par les parties prenantes concernant le système à développer. Un document d'exigences peut être réalisé en :

- **langage naturel (LN)** : le LN est extrêmement expressif et flexible mais il est inadéquat pour capturer les relations clés entre les différents éléments. Le LN présente certaines limites :
  - compréhension arbitraire de la signification ;
  - verbosité [Web07].
  - les 7 pêchés capitaux [Hey05] :
    1. Bruit : élément(s) qui n'apporte(nt) rien d'utile (*Exemple* : redondance) ;
    2. Silence : caractéristique(s) du problème non évoquée(s) dans le texte.
    3. Surspécification : texte contenant des éléments de solution et non du problème.
    4. Contradiction : texte contenant des caractéristiques incompatibles.
    5. Ambiguïté : éléments du texte interprétables de plusieurs manières, entraînant une compréhension arbitraire de la signification.
    6. Référence en avant : des concepts référencés dans le texte avant d'être définis.
    7. Repentir : éléments essentiels définis tardivement ou en passant.

*Exemples* : français, anglais, etc.

- **notation semi-formelle** : la notation semi-formelle capture la structure et une partie de la sémantique. La notation semi-formelle est essentiellement visuelle pour une communication rapide avec les parties prenantes.  
*Exemples* : Diagrammes UML<sup>10</sup>, tableaux, LN structuré, etc.
- **notation formelle** : La notation formelle établit une sémantique précise permettant d'effectuer des raisonnements, de vérifier la cohérence, etc.  
*Exemples* : Z<sup>11</sup>, etc.

Notons que dans tous les cas l'**utilisation du LN est inévitable**. Même si le LN n'est utilisé que dans la conception initiale, l'ambiguïté inhérente à son utilisation pose problème lors de la transition du LN vers un autre formalisme.

La **documentation** permet de :

- guider l'élicitation, pouvant aider à :
  - établir les questions à poser ;
  - faire apparaître des exigences cachées.
- fournir une mesure de la progression.  
*Exemple* : La complétude de la documentation entraîne la complétude de l'élicitation.
- découvrir des problèmes par des incohérences dans la documentation.  
*Exemples* :
  - des exigences en conflits ou infaisables ;
  - de la confusion sur la terminologie, la portée, etc. ;
  - désaccords entre les parties prenantes.
- vérifier la compréhension des besoins des parties prenantes.

<sup>9</sup>Terme français employé à HQ pour *Software Requirements Specification* (SRS). Appelé souvent aussi document de spécification.

<sup>10</sup>*Unified Modelling Language*.

<sup>11</sup>Notation formelle pour la spécification des systèmes informatiques, basée sur l'algèbre des ensembles et le calcul des prédicats [not07].



**Validation et vérification** La validation et la vérification concernent la revue des exigences. La vérification consiste à établir la concordance entre les exigences et l'implémentation. Cependant, cette concordance n'assure pas la validation des exigences.

La validation des exigences contrôle si le système établi rencontre les souhaits des différentes parties prenantes [Eas05].

La figure 1.1 présente la distinction entre vérification et validation.

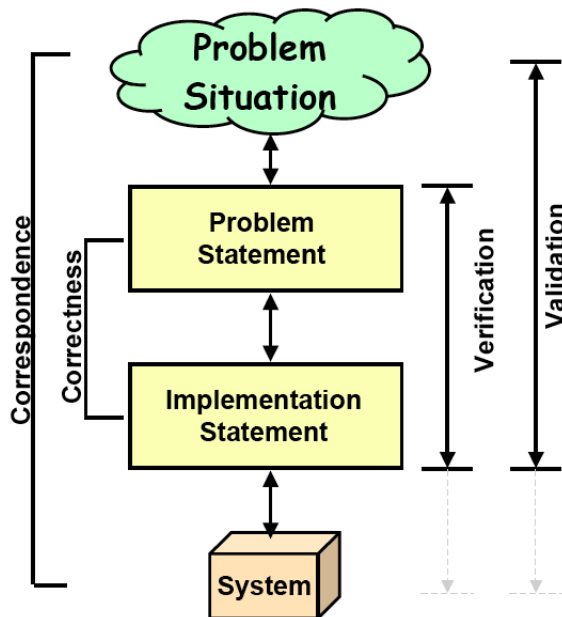


FIG. 1.1 – Portée de la vérification et de la validation [Eas05].

En théorie, les activités de l'IE sont totalement dissociées des activités de conception. Cependant, en pratique, on remarque que *la distinction entre l'IE et la conception n'est pas réalisable car toute conception entraîne une nouvelle vision de la situation du problème* (cf. FIG. 1.2).

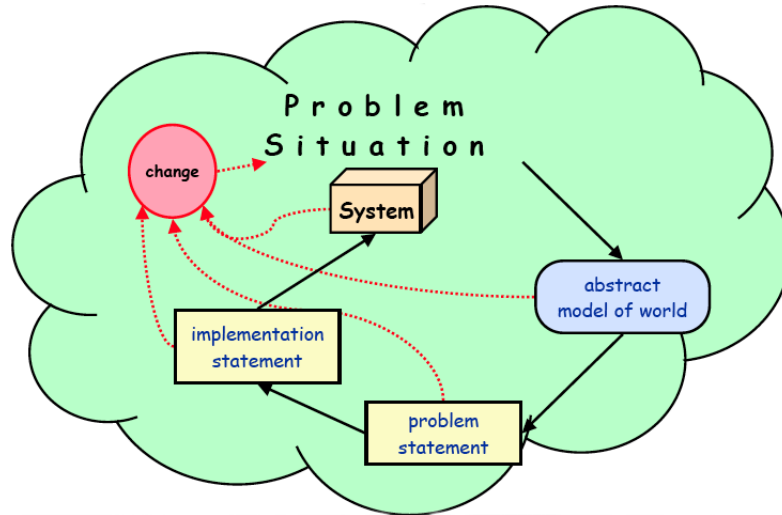


FIG. 1.2 – La conception change la situation du problème [Eas05].

Des allers et venues constants entre l'IE et la conception sont alors nécessaires pour (cf. FIG. 1.3) :

- répondre aux interrogations provenant de la conception ;
- établir des exigences plus détaillées ;
- tenir à jour les exigences et l'implémentation.

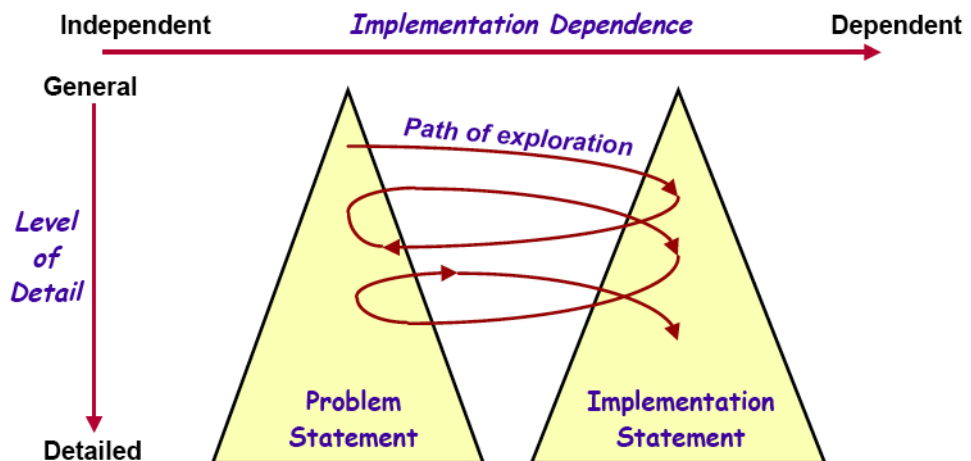


FIG. 1.3 – Entrelacement du problème et de la solution [Eas05].

Les activités de l'IE en lien avec la conception peuvent différer selon le choix de l'organisation. Nous avons choisi ici de suivre une découpe d'activités qui, selon nous, correspond le plus au mode de fonctionnement de l'unité Conception-Automatismes à Hydro-Québec (HQ) ainsi qu'à leur OSIE *GenSpec*. Cette découpe d'activités se base sur une représentation textuelle des exigences en LN.

Les **activités liées à la conception architecturale à partir des exigences** sont [Dor99] :

- la décomposition (et l'abstraction) ;
- l'allocation ;
- le flowdown<sup>12</sup>, et la traçabilité ;
- l'établissement des interfaces du système.

**Décomposition** La décomposition est le découpage du problème en sous-problèmes de moindre taille. Ce découpage correspond à une subdivision du système en sous-systèmes plus simple.

Chaque sous-système est géré séparément pour ensuite être intégré dans un système global répondant au problème général (cf. FIG. 1.4). Pour les systèmes les plus complexes, ce processus est réitéré pour chaque sous-système à divers niveaux de décomposition [LW03].

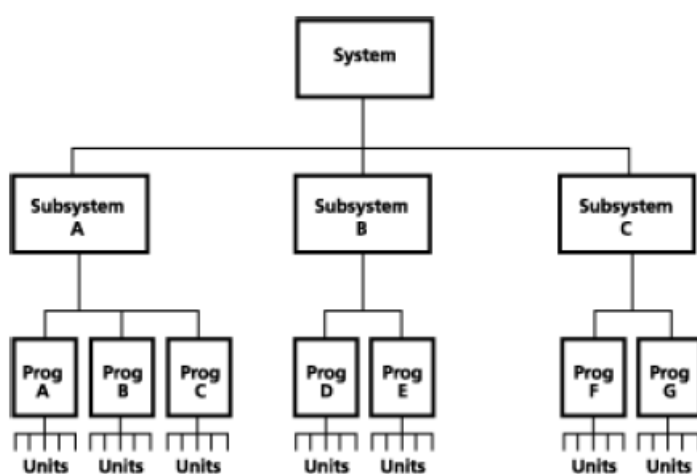


FIG. 1.4 – Exemple de hiérarchie de système [Dor99].

**Allocation** L'activité d'allocation permet d'allouer les exigences à un ou plusieurs sous-systèmes définis lors de la phase de décomposition.

Lors de l'allocation, il apparaît souvent que :

- les exigences du système nécessitent des modifications (ajouts, suppressions et corrections) ;
- les définitions des sous-systèmes ne sont pas correctes.

Le processus d'allocation est itératif, amenant à une allocation complète des exigences du système. La figure 1.5 montre la répartition d'un ensemble d'exigences système dans différents sous-systèmes ; chaque exigence étant allouée à au moins un sous-système de niveau inférieur. Dans la figure, les sous-systèmes de niveau inférieur ont été appelés “*Subsystem*” (sous-système) mais en pratique les noms sont arbitraires, l'accent étant porté sur les niveaux de décomposition.

<sup>12</sup>Terme sans équivalence en français. Correspond à la spécialisation des exigences selon le(s) sous-système(s) chargé(s) de les mettre en oeuvre.

SYSTEM REQUIREMENTS	SUBSYSTEM A	SUBSYSTEM B	SUBSYSTEM C
SYS 001	X	X	
SYS 002	X		X
SYS 003		X	
SYS 004	X	X	X
SYS 005			X
SYS 006	X	X	
SYS 007			

FIG. 1.5 – Exemple d'allocation d'exigences système [Dor99].

**Flowdown** Suite à l'allocation, l'activité de flowdown adapte chaque exigence allouée à un sous-système, chaque sous-système possédant ses caractéristiques propres. D'une seule exigence allouée, peuvent découler une ou plusieurs exigences. Ces exigences sont appelées des *exigences dérivées*.

Les exigences dérivées ressemblent fortement à celles de plus haut niveau. Cependant, celles-ci peuvent être différentes dans le cas où les ingénieurs reconnaissent la capacité que doit avoir le sous-système de plus bas niveau pour répondre à l'exigence de plus haut niveau.

À titre d'exemple, reprenons l'exigence définie plus haut lors de la définition de l'*élicitation* :

Exigence 1 : “*Le système prend en entrée les détails des requêtes de planification et fournit une solution en sortie.*”

Supposons que le *Système de gestion de rendez-vous* soit décomposé en plusieurs sous-systèmes. Le sous-système *Input / Output* est chargé des interactions avec l'utilisateur. L'exigence 1, par l'allocation et le flowdown au sous-système *Input / Output*, devient les exigences dérivées suivantes :

Exigence 1.1 : “*Le système prend les détails des requêtes par l'intermédiaire du clavier*” ;

Exigence 1.2 : “*Le système affiche la solution de la requête à l'écran*”.

On voit alors que les exigences 1.1 et 1.2 sont dérivées de l'exigence 1 selon les caractéristiques du sous-système *Input / Output*.

Notons que les exigences dérivées peuvent être allouées à d'autres sous-systèmes lors d'une découpe hiérarchique plus fine. Plus on descend dans la hiérarchie, plus le niveau de détail est important. Les exigences système sont générales par nature, tandis que les exigences de niveaux inférieurs sont très spécifiques.

La figure 1.6 montre le résultat d'un processus de flowdown correspondant à un ensemble complet d'exigence pour chaque sous-système. Par exemple, l'exigence *SYS001* correspond aux exigences *SSA001* et *SSA002* dans le sous-système A et à l'exigence *SSB001* dans le sous-système B.

SYSTEM REQUIREMENT	SYSTEM A REQUIREMENT	SYSTEM B REQUIREMENT	SYSTEM C REQUIREMENT
SYS 001	SSA 001 SSA 002	SSB 001	–
SYS 002	SSA 003 SSA 004 SSA 005	–	–
SYS 003	–	SSB 002 SSB 003	–
SYS 004	SSA 006 SSA 007	SSB 004 SSB 005 SSB 006	SSC 003
SYS 005	–	–	SSC 004 SSC 005
SYS 006	–	SSB 007 SSB 008	–
SYS 007	SSA 008 SSA 009	SSB 009	–

FIG. 1.6 – Exemple de flowdown d'exigences système [Dor99].

**Traçabilité** Durant les processus d'allocation et de flowdown, le nombre d'exigences augmente rapidement. Garder la trace de toutes ces exigences est essentiel afin d'assurer que toutes les exigences soient correctement rédigées, allouées et dérivées et ce, à tous les niveaux, sans perte ni apparition de nouvelles exigences. La traçabilité permet de sauvegarder l'ensemble des liens depuis et vers une exigence.

La traçabilité est un support pour la validation des exigences. Elle permet, pour toute modification apportée à une exigence X, de connaître l'impact ou de répercuter facilement les modifications aux exigences Y associées. La traçabilité concerne non seulement les liens entre les exigences mais aussi les liens d'une exigence à ses origines et son implémentation.

**Interfaces** Avant le développement des exigences systèmes, les interfaces externes du système doivent être établies. Ces interfaces externes constituent les moyens d'interactions entre le système et le monde extérieur. De plus, à chaque niveau de décomposition, lors des processus d'allocation et de flowdown, des interfaces internes doivent être établies entre les différents sous-systèmes. Chaque sous-système peut posséder deux types d'interfaces :

- les **interfaces externes**, définies au plus haut niveau, qui sont associées aux sous-systèmes chargés de leur réalisation ;
- les **interfaces internes**, définies au niveau du sous-système, qui constituent les interfaces sous-système-à-sous-système.

La figure 1.7 présente une définition d'interfaces. *A* représente une interface externe du système global. Cette interface représente un extrant<sup>13</sup> produit par le système. Une fois le découpage du système effectué, l'interface externe *A* est allouée au sous-système 1 et une interface interne *B* est réalisée entre les sous-système 3 et 4.

<sup>13</sup>Correspond au terme anglais *output*.

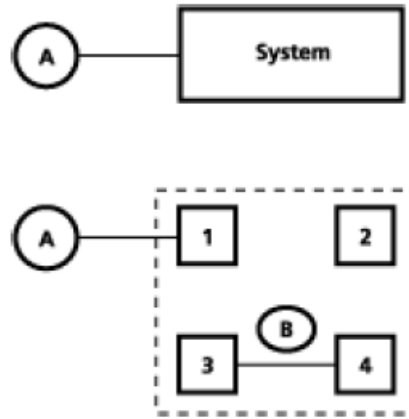


FIG. 1.7 – Définition d'interfaces [Dor99].

### 1.1.3 Importance de l'IE

L'IE joue un rôle capital à plusieurs égards dans un projet de développement.

**Coûts des erreurs** Les erreurs produites lors de la phase d'IE s'avèrent souvent catastrophiques pour le développement d'un projet. Plus longtemps une erreur reste non détectée, plus coûteuse sera sa correction [Sut02]. Une erreur d'exigence peut coûter 100 fois plus qu'une erreur de programmation lorsqu'elle est découverte lors de la phase de test. La courbe en traits continus sur la figure 1.8 présente cette situation (de loin la plus fréquente). Peu de ressources ont été associées aux premières phases du développement. Il en découle une mauvaise compréhension du système à réaliser et de nombreuses corrections (coûteuses) à effectuer.

La courbe en traits discontinus représente un développement chanceux, où malgré le peu de ressources accordées aux phases en amont, les exigences ont été raisonnablement bien comprises lors du codage. Cette situation ne se produit jamais en pratique.

Tandis que la courbe en pointillée montre que si davantage de ressources sont consacrées aux phases en amont (requirements, specifications, design), les coûts nécessaires pour la réalisation du projet sont (en théorie) moindre.

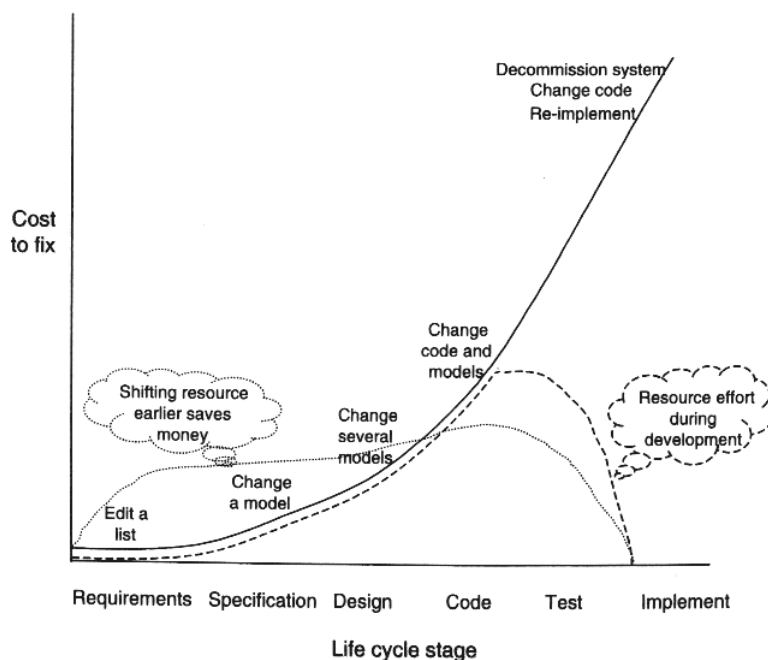


FIG. 1.8 – Coût d'un changement tout au long d'un processus de développement [Sut02].

Une étude du *Standish Group* [Gro94] portant sur les projets de développement aux Etats-Unis, tel que cité par [Eas05], nous donne les 3 facteurs principaux d'échec d'un projet de développement :

- Le manque d'implication des parties prenantes ;
- Les exigences incomplètes ;
- Les exigences changeantes.

Nous remarquons que les trois causes majeures d'échec sont associées à la gestion des exigences.

**Objectifs** L'IE a pour but d'identifier les objectifs d'un projet de développement. Une mauvaise compréhension de ces objectifs entraîne un logiciel de mauvaise qualité. Nous retrouvons l'objectif d'un système dans les activités humaines. Par exemple, l'objectif d'une banque provient de ses activités de business et de leurs besoins pour leurs clients. Cependant, la découverte de l'objectif est souvent complexe car nous retrouvons de nombreux types de personnes et d'activités. Ces types de personnes ou d'activités peuvent avoir des intérêts parfois incompatibles.

Nous donnons un exemple, repris de [Sut02], pour représenter ceci :

*“London Ambulance Service’s Computer Aided Dispatch System. This was intended to replace the manual system of answering emergency telephone calls from members of the public, finding out where the emergency was and the dispatching one the available ambulances to the location of the accident. (...) The regional Health Authority development team specified the requirements. There was no consultation or opportunity for feedback until the system was ready for deployment. At that stage, performance problems were so bad that just getting the system to work was an uphill struggle. Poor requirements analysis failed to detect several problems : radio blackspots where the ambulance crews could not be contacted, poor user interfaces on the mobile data terminals which resulted in the ambulance crews not reporting call progress accurately, with the knock-on effect that the system database became inaccurate (...)”*

Cet exemple montre l'importance d'une bonne compréhension des objectifs sous-jacents au système.

Néanmoins, dans le cas hypothétique où l'objectif est clairement défini et qu'aucune erreur n'a été commise dans l'ensemble des exigences, l'IE permet en théorie :

- d'établir un **accord** entre les développeurs, clients, et usagers sur le travail à effectuer et sur les critères d'acceptation du système à délivrer.
- de définir une base pour l'**estimation des ressources nécessaires** (coûts, quantité de personnel et formations, équipement, temps)
- d'**améliorer l'utilisabilité, la maintenabilité**, etc. du système.
- réaliser l'objectif du projet en **réduisant les ressources** nécessaires (moins d'omissions, d'incompréhensions) [Dor99].

**Document d'exigences** Réaliser un document d'exigences est une pratique courante de la gestion des exigences. Il présente quatre utilités majeures :

1. **Communication** : explique le domaine d'application et le système à développer ;
2. **Contrat** : exprime l'accord et l'engagement ;
3. **Base pour l'évaluation du software** : constitue un support pour les tests, la validation et la vérification. On y retrouve l'information permettant de vérifier si le système répond aux exigences ou non ;
4. **Base pour le contrôle du changement** [Eas05].



### 1.1.4 Limitations

La tâche de l'IE est complexifiée par deux facteurs :

- **Les ressources humaines et le software sont fortement liés.** Les activités humaines et le software se chevauchent. Le software change les activités et inversement.
- **Les problèmes auxquels le software doit répondre sont “wicked”<sup>14</sup>.** Il n'y a pas de définition définitive du problème, il n'y pas de règles pour arrêter l'élaboration de la solution. La difficulté est de décider quand l'analyse des exigences devrait s'arrêter et ce, pour deux raisons :
  1. **“l'inconsistance des usagers”** : les gens ont tendance à découvrir de nouvelles exigences une fois que l'analyste pense que tout ce qu'ils veulent a été relevé.
  2. **“l'instabilité du problème”** : à l'aide de ressources infinies, il est théoriquement possible de rédiger des exigences presque parfaites. On parle d'exigences “presque parfaites” car le monde, et donc les exigences, sont en constante évolution. Les exigences réfèrent obligatoirement à des exigences “d'hier” car depuis, le monde et l'opinion des usagers ont évolué. Les exigences sont donc, au mieux, un compromis [Sut02].

En conséquence, l'IE présente certaines limitations :

- L'IE n'est **pas nécessairement un processus séquentiel** : idéalement, la définition du problème est réalisée avant la définition de la solution (cf. sous-section 1.1.2). Cependant, aucune obligation d'ordre de rédaction n'est imposée. Les activités d'IE se réalisent continuellement tout au long du processus de développement.
- La **définition** du problème est **imparfaite** : il s'agit d'une approximation du “monde réel”. Elle est variable, contient des imprécisions et omet des informations.
- Établir une spécification n'est **pas toujours rentable** : l'analyse des exigences a un coût et son bénéfice sera variable d'un projet à l'autre. La taille d'un projet est une des variables à considérer. S'il s'agit d'un projet de petite taille, le bénéfice associé à l'ensemble des activités de l'IE sera moindre que pour un projet de taille importante.
- La définition du problème ne devrait **jamais être considérée comme fixée** : le changement est inévitable et devrait être planifié. Une façon d'incorporer les changements périodiquement devrait exister.

Les trois objectifs majeurs de l'IE concernant la documentation de la future application sont de réaliser cette documentation (cf. FIG. 1.9) :

- de la manière la plus **complète** (*specification*) ;
- de la manière la plus **formelle** (*representation*) ;
- avec l'**accord** de l'ensemble des parties prenantes sur les exigences (*agreement*).

La figure 1.9 représente l'évolution d'une exigence selon ces **trois dimensions** tout au long du processus d'IE. Au début de ce processus, les usagers peuvent avoir une vision différente des besoins ou seulement des idées partielles ou incomplètes de ce qu'ils veulent. Au départ, les exigences sont informelles dans leur représentation, imprécises et personnelles car elles sont soutenues par des individus différents qui entrent souvent en conflit.

Cependant, le résultat désiré de l'IE est très différent. Idéalement, il serait un ensemble complet de spécifications du système, avec les contraintes des ressources disponibles, utilisant un langage formel et accepté par l'ensemble des parties prenantes [Sut02]. Cette situation est représentée comme étant le point ultime (idéal) à atteindre.

La figure 1.9 souligne également que **le processus à suivre** pour atteindre le point ultime **est irrégulier**. Il n'existe pas une seule façon de faire en IE, l'IE dépendant du problème et du contexte du projet de développement.

<sup>14</sup>Terme anglais utilisé par Easterbrook [Eas05] référant à “mal fichus”.

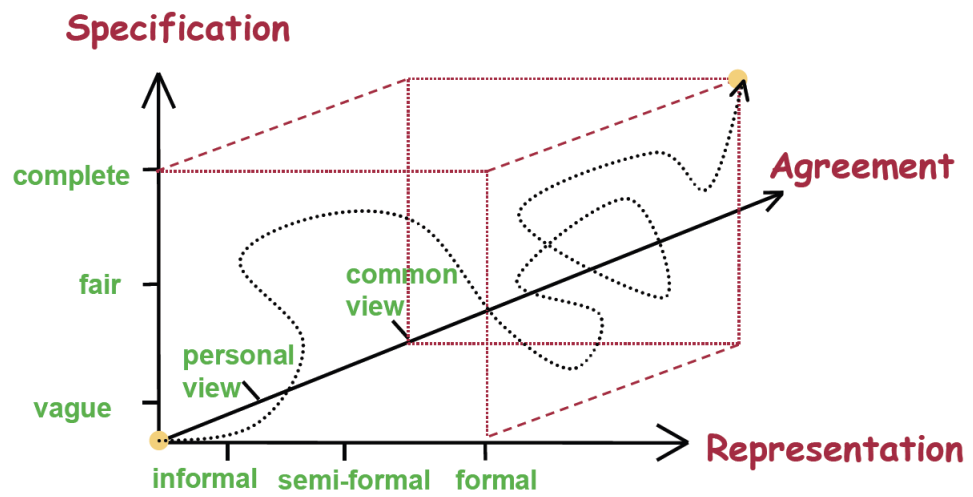


FIG. 1.9 – Evolution d'une exigence selon les 3 dimensions de l'IE [Poh93].

## 1.2 Les outils de support à l'ingénierie des exigences

Dans cette section, nous allons :

1. définir ce qu'est un OSIE (1.2.1) ;
2. établir les intérêts pour les OSIE (1.2.2) ;
3. établir une liste d'exigences que devrait supporter tout OSIE (1.2.3) ;
4. établir une liste des faiblesses des OSIE (1.2.4).

### 1.2.1 Définition

*Les outils de support à l'ingénierie des exigences (OSIE) sont des outils logiciels fournissant une assistance automatisée durant le processus d'IE* [Mat05]. Ceux-ci suivent différents canevas de documents d'exigences (Volere, IEEE830, etc.), ont différentes capacités et différents degrés de maturité en fonction de leur applicabilité à des projets d'IE. Ils permettent de garder les spécifications cohérentes, à jour et accessibles [IEE04].

Adapté de [IEE04], les OSIE supportent les tâches essentielles de l'IE :

- l'élicitation et la documentation des exigences ;
- la décomposition et l'abstraction des exigences en regroupements ainsi que leur allocation à ces regroupements (sous-systèmes) ;
- la dérivation des exigences à des niveaux de détails plus importants au sein d'autre regroupements (flowdown) ;
- l'établissement de la traçabilité entre les exigences et la conception, l'implémentation et les tests ;
- la définition et la liaison de cas d'utilisation et de jeux de tests aux exigences pour la validation et la vérification des exigences ;

### 1.2.2 Intérêts

Cette sous-section, largement inspirée de Firesmith [Fir03], présente les intérêts à l'origine du développement et de l'utilisation d'OSIE.

**Années 1970 et 1980** Dans les années 1970 et 1980, la gestion des exigences est perçue comme une étape simple, distincte du développement. Après l'élicitation des exigences fonctionnelles d'un projet, un simple traitement de texte est utilisé pour gérer manuellement le document d'exigences<sup>15</sup>. Le modèle de développement en cascade (cf. FIG. 1.10) entraîne qu'un document d'exigences est utilisé par l'ensemble des intervenants d'un projet comme une base solide et stable dans le temps et ce, quels que soient leurs objectifs.

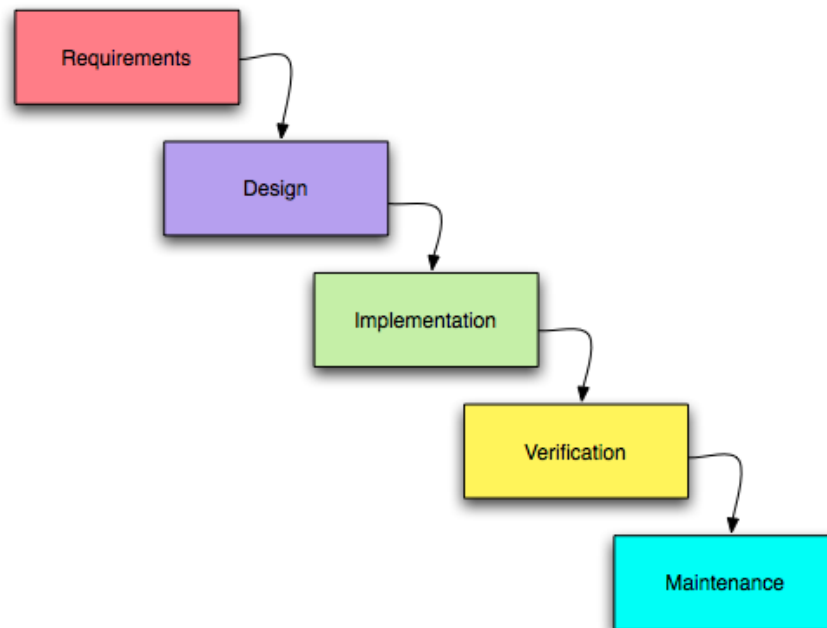
Cette manière de gérer les exigences s'avère réductrice. Malheureusement, pour beaucoup de projets, cette façon de faire reste toujours prédominante [Fir03].

**Nouveaux cycles de développement** L'apparition de nouveaux cycles de développement (itératif, incrémental, etc.), fournissant une alternative au modèle en cascade, apporte un regard neuf sur la gestion des exigences.

Le modèle en cascade se base sur l'hypothèse que l'ensemble des exigences rédigées après l'étape d'IE est connu et stable. Ces exigences sont alors gelées pour être utilisées dans les phases de conception (*design*), d'implémentation (*implementation*), de tests (*verification*) et de maintenance (*maintenance*). Or, les exigences sont très souvent instables dans le temps.

Les cycles itératifs, incrémentaux, etc. de développement reconnaissent la nécessité d'une gestion continue des exigences. Une gestion manuelle des exigences s'avère insuffisante car un document d'exigences nécessite beaucoup de temps et de moyens pour sa réalisation et

<sup>15</sup>Expression correspondant à la réalisation des activités d'IE sans aucun OSIE ou au maximum à un traitement de texte ou un tableur.

FIG. 1.10 – *Modèle de développement en cascade [Wik07d].*

sa maintenance [Fir03]. Un OSIE fournit une réponse efficace à ces besoins permettant une gestion appropriée des exigences.

**Évolution des systèmes** La taille et la complexité des systèmes n'ont cessé de croître ces dernières décennies. Cela va de paire avec la complexification des organisations. De nos jours, de nombreux systèmes (tels que les ERP *SAP* ou *PeopleSoft*) permettent de gérer un grand nombre de tâches de l'entreprise. Suite aux multiples accords ou fusions entre entreprises, nous retrouvons également des systèmes "multi-entreprises" chargés d'effectuer la liaison entre les systèmes propres à chaque entreprise.

Une pression importante des marchés et les changements organisationnels nécessitent une réactivité importante.

Les exigences des systèmes actuels se sont, par conséquent, multipliées, complexifiées, diversifiées et leur gestion s'est accélérée. De plus, outre les exigences fonctionnelles, nous retrouvons des exigences de qualités, de données, des contraintes de conception, d'implémentation et de tests. Un OSIE est une nécessité pour gérer, maintenir et utiliser un nombre important d'exigences [Fir03].

**Méthodes et notations** Des canevas de document d'exigences tels que Volere et IEEE 830, ainsi que des langages de modélisation tel qu'UML, se sont répandus afin d'améliorer la gestion des exigences. Ils permettent de modéliser différents aspects d'un système avec objectif d'établir une image complète du système à réaliser.

Ces formalismes ont un impact non négligeable sur les documents d'exigences [Fir03]. Un OSIE permet un support et une application plus aisée de ces formalismes.

### 1.2.3 Exigences relatives aux OSIE

Cette sous-section présente une liste non exhaustive d'exigences que devrait satisfaire tout OSIE. L'intention n'est pas de définir une liste définitive d'exigences, mais bien de mettre en évidence les exigences principales qu'est censé supporter tout OSIE. Ces exigences ne sont volontairement pas prioritisées, leur importance est relative au type et aux besoins de chacun des intervenants du projet [IEE04, Fir03]. Notons également que ces exigences sont très diverses. Elles dépendent de nombreux facteurs tels que le domaine du logiciel développé, la taille du projet et de l'organisation, le nombre de réutilisations et la maturité du processus [IEE04].

**Support aux usagers (n° I)** Les OSIE doivent fournir à leurs usagers une interface graphique leur permettant de créer des projets, des exigences et de gérer les informations de ces projets [Fir03]. L'ensemble des fonctionnalités présentes doit être décrit dans un manuel d'aide afin de fournir aux usagers le support nécessaire pour la compréhension de ces fonctionnalités [HD97].

**Basé sur un repository (n° II)** Toute exigence, quel que soit son format, textuel (ou graphique) ou référence à des documents externes [IEE04], doit être maintenue et identifiable dans une collection d'exigences qui repose sur le principe de repository<sup>16</sup> au lieu d'un document en format papier [Fir03, LD04]. Cette pratique permet de produire des exigences structurées de meilleure qualité pour satisfaire les besoins des intervenants [Fir03]. Les OSIE permettent alors de sauvegarder des exigences à un moment donné pour qu'elles puissent servir de base à un développement futur [IEE04].

La figure 1.11 illustre le principe de repository [Fir03]. Deux repository sont présentés, l'un jouant le rôle de catalogue des exigences existantes et destinées à la réutilisation (*Organizational Reuse Repository*), l'autre servant à maintenir les exigences d'un projet (*Endeavor Requirements Repository*). C'est grâce à ce dernier repository que les exigences des clients, les exigences réutilisables et toute autre information de gestion sont maintenues. Il sert de base à l'analyse des exigences réalisées par les OSIE.

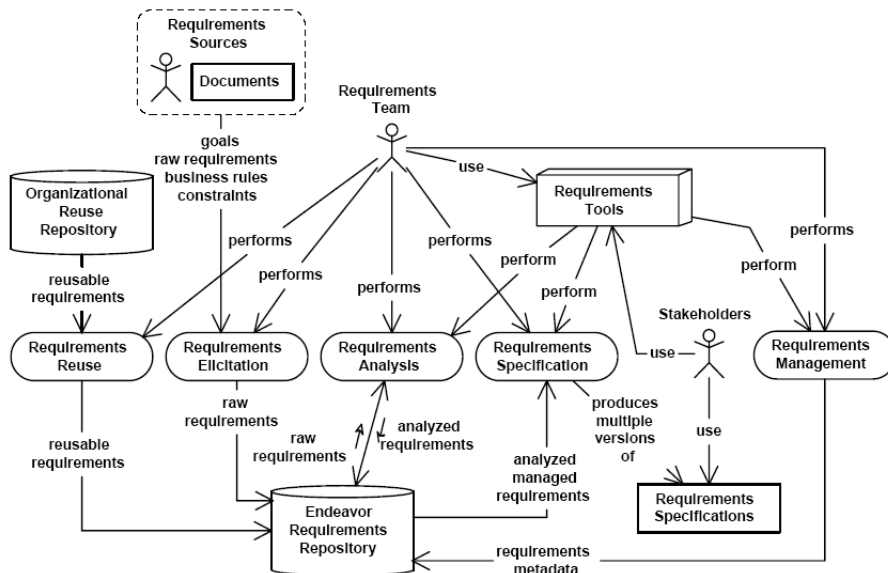


FIG. 1.11 – Spécification d'exigences basée sur un repository [Fir03].

<sup>16</sup>Ensemble d'informations (représentées généralement par des bases de données) permettant de stocker et de maintenir tout type d'informations.

**Gestion des exigences adaptée aux besoins (n° III)** Certains OSIE proposent de créer des vues paramétrables sur les exigences pour ne mettre en évidence que celles qui sont pertinentes aux besoins spécifiques des intervenants. Tel que décrit précédemment dans la sous-section 1.2.2, un document d'exigences peut être utilisé par plusieurs types d'intervenants. Ces derniers ont des besoins distincts selon leur rôle dans le projet [Fir03]. Qu'il s'agisse d'un responsable, d'un concepteur, d'un développeur, d'un testeur ou d'un administrateur de projet, l'intérêt pour les exigences est relative à un niveau d'abstraction. Par exemple, l'information qu'utilise le responsable du projet afin de décider de l'approbation des exigences, est différente de celle requise par le testeur [Fir03]. C'est pour cette raison que les OSIE doivent fournir aux usagers de multiples possibilités de gérer eux-mêmes les exigences. Par exemple, la possibilité de décomposer des exigences entre elles pour faire apparaître plusieurs niveaux de détail [HD97], ou encore la possibilité de travailler selon différentes vues sur les mêmes exigences [IEE04]. Ces vues diffèrent, par exemple, par le niveau de détails qu'elles entretiennent avec les exigences, le type d'exigences fonctionnelles ou non, ou encore le type de documents (spécifications ou rapports) [Fir03].

**Traçabilité entre les exigences et gestion du changement (n° IV)** Des liens entre les exigences doivent pouvoir être établis dans un même projet ou dans des projets distincts. Ces liens amènent de la traçabilité entre les exigences, ou entre différents éléments du système [IEE04]. Un tel mécanisme de traçabilité met en évidence les références sources des exigences, la propagation des modifications entre les exigences [LD04] et met en relation les cas d'utilisation avec les cas de tests pour s'assurer de la couverture complète de ces cas d'utilisation [HD97]. Notons que les modifications relatives aux exigences - l'historique - doivent être commentées et enregistrées dans un repository tout en établissant un mécanisme de versions permettant de retracer l'historique des modifications [IEE04]. Les modifications des exigences sont très fréquentes, notamment lorsque les usagers finaux découvrent les premières ébauches de prototypes et souhaitent que des changements (liés par exemple à l'interface graphique) soient apportés aux exigences [LD04]. Le support du changement dans les exigences et la capacité à retracer les exigences jusqu'à leur source sont considérés comme essentiels pour de nombreux praticiens [HD97].

**Génération de documents d'exigences (n° V)** Les exigences d'un projet doivent pouvoir être générées, grâce à l'OSIE, sous différents types de documents, officiels ou arbitraires selon le contexte, et respecter des canevas d'exigences (standardisés) préalablement définis [LD04] (par exemple le standard IEEE 830). Les OSIE doivent être paramétrables pour permettre aux usagers d'inclure dans la génération de documents toute information qu'ils jugent utile. Ces OSIE offrent donc la possibilité d'inclure dans la génération, soit une partie des exigences, soit toutes les exigences d'un projet [IEE04, Fir03].

**Importation de données (n° VI)** Les OSIE doivent permettre l'importation des informations partir de documents existants ou d'exigences existantes [IEE04, Fir03]. Les OSIE doivent disposer d'interfaces entre eux leur permettant de rendre utilisable les informations traitées par chacun de ces outils. L'importation doit se réaliser de manière interactive avec l'utilisateur pour qu'il puisse vérifier à tout moment la pertinence des exigences qu'il souhaite importer [IEE04].

**Vérifications (n° VII)** Les OSIE doivent fournir des mécanismes de vérification relatifs aux exigences. Ces vérifications mettent en évidence d'éventuelles incohérences, notamment linguistiques ou liées à la structure des exigences [IEE04].

**Traitement de l'ambiguïté (n° VIII)** Dans le but de réduire l'ambiguïté des exigences d'un projet, les OSIE doivent maintenir un glossaire relatif aux composants et aux exigences pour définir la terminologie employée dans le système [LD04]. Maintenir un glossaire est souvent perçu comme étant une exigence "nice-to-have", cependant, de nombreux praticiens affirment que la terminologie nécessite d'être clairement définie en début de projet [HD97].

**Support et intégration de l'ensemble des activités du processus d'IE (n° IX)** La gestion des exigences par les OSIE suggère une couverture totale des activités relatives à l'IE. Les OSIE ne doivent donc pas être isolés dans le processus de développement. Il doivent être perçus comme des éléments critiques et intégrés dans un environnement de développement d'un projet [Fir03].

**Environnement distribué (n° X)** La gestion suggère que les différents usagers des OSIE puissent, d'une part, travailler sur les documents d'exigences en parallèle en accord avec leurs droits et leurs rôles dans un projet [Fir03] et d'autre part, travailler à distance sur ces exigences. Les OSIE doivent permettre de travailler dans un environnement distribué [Fir03].

**Sécurité (n° XI)** Les informations traitées par les OSIE peuvent être confidentielles. Les OSIE doivent fournir une sécurité adaptée. Cette attente peut être satisfaite par l'utilisation de mécanismes de chiffrement de données, de privilèges des intervenants, d'identifications, de transactions, etc [IEE04].

**Facteurs de qualité (n° XII)** D'autres facteurs de qualité sont également à prendre en compte, tels que la complétude, l'internationalisation, la performance, l'extensibilité, l'utilisabilité et la convivialité [Fir03]. Les OSIE doivent fournir la possibilité de classer les exigences (notamment selon des priorités), et d'évaluer le coût de réalisation de chacune des exigences [HD97].

## 1.2.4 Faiblesses

Cette sous-section présente les faiblesses les plus courantes des OSIE commerciaux actuels. Les faiblesses sont accompagnées d'un numéro pour référencer aux exigences relatives aux OSIE décrites dans la sous-section 1.2.3.

**Ambiguïté (VIII)** L'utilisation d'un LN, par l'intermédiaire d'un OSIE ou non, entraîne inévitablement de l'ambiguïté lors de la rédaction des exigences. Un glossaire évolué permet une réduction importante de l'ambiguïté. Peu d'OSIE commerciaux possèdent un mécanisme de glossaire approprié. Dans le meilleur des cas, une simple définition en LN peut être associée à un terme. Le chapitre 3 comporte une description détaillée des problèmes liés à l'ambiguïté.

**Complexité des interfaces usagers (XII)** Un OSIE est souvent trop complexe pour l'utilisateur [Mat05]. Les OSIE nécessitent "*une formation de plusieurs jours, voire de plusieurs semaines. Ils exigent de surcroît, plusieurs mois de pratique pour les maîtriser*" [Buj04]. Ces OSIE doivent supporter l'entrée d'exigences et de leurs méta-informations par une interface conviviale. L'interface ne se limite pas à l'insertion et à la maintenance d'exigences mais également à la génération de documents d'exigences et de rapports [Fir03].

Un OSIE possède de nombreuses fonctionnalités. Il est difficile pour un usager de percevoir le rôle général d'un OSIE, non pas par la complexité mais par la quantité de fonctionnalités fournies [Mat05].

**Formalisme unique (III)** Il est recommandé de rédiger les exigences selon les méthodes préférées de l'utilisateur (typiquement le LN) et du concepteur (typiquement des diagrammes de modélisation tels que les cas d'utilisation<sup>17</sup>, diagramme d'activités, etc.) afin d'améliorer la communication par un compromis entre différents formalismes. Cependant, les OSIE ne permettent généralement qu'une description en LN et peu ou pas de possibilités de modélisation. À l'inverse, certains OSIE fournissent d'importantes capacités de modélisation mais sont trop rigoureux et ne permettent pas l'utilisation du LN [LD04].

<sup>17</sup> Terme français référant à *Use Case*.

**Manque d'interopérabilité (IX)** Les OSIE ont un rôle critique dans un environnement de développement intégré. Ils doivent communiquer avec les autres OSIE [Fir03]. Cependant, peu d'OSIE commerciaux possèdent une interface avec d'autres outils, si ce n'est ceux qui sont développés par le même vendeur [Mat05].

**Manque de collaboration (X)** Les exigences d'un projet doivent pouvoir être rédigées simultanément par plusieurs auteurs. L'ensemble des parties prenantes doivent pouvoir également accéder aux exigences afin de les consulter, les évaluer et les approuver [Fir03]. La majorité des OSIE sont des applications autonomes et ne fournissent pas de moyen de collaboration, de coopération et d'accord entre les différents intervenants d'un projet. Dans le meilleur des cas, les outils les plus performants partagent les informations au niveau de la BD sur l'intranet de l'organisation [Mat05].

**Manque de traçabilité (IV)** Les OSIE doivent fournir un mécanisme de traçabilité sur la totalité du processus de développement. Ils incluent la traçabilité des exigences avec la conception, l'implémentation et les tests [Fir03].

**Non emploi d'un repository (II)** Un OSIE doit permettre la représentation des exigences comme des objets individuels à un haut niveau de détail [Fir03] dans un repository. Cependant, peu d'OSIE sont basés sur des repository comme suggéré par Firesmith (cf. FIG. 1.11) [Fir03]. Dans le meilleur des cas, ces outils possèdent des BD pour le stockage d'informations mais pas pour la réutilisation [Mat05].

**Outils CASE et non OSIE (I, III, IX)** Certains OSIE devraient être classifiés comme outils CASE<sup>18</sup> et non comme simple OSIE. De par la complexité des outils CASE et des documents qu'ils produisent, la communication avec l'ensemble des intervenants non-techniques peut s'avérer plus complexe. La communication entre les concepteurs et les usagers finaux peut même se voir diminuée par l'utilisation de tels outils [LD04].

Au delà de l'ensemble de ces faiblesses, Matulevičius [Mat05] établit une liste des limitations majeures des OSIE commerciaux :

- une faible représentation des canevas d'exigences ;
- une faible maintenance de la traçabilité ;
- un manque de participation des usagers ;
- un manque de moyens de collaboration ;
- un manque de support à la réutilisation d'exigences.

---

<sup>18</sup> *Computer Aided Software Engineering* : Ensemble de logiciels organisés autour d'une base de données de spécifications (repository). Leur objectif est de fournir une assistance au développement et à la maintenance de logiciels ou de certains composants d'un logiciel [ddBdD03].



## 1.3 Résumé du chapitre

À l'issue de ce chapitre, nous avons défini l'IE ainsi que les OSIE.

L'IE établit l'objectif du développement d'un système. Elle comporte un ensemble d'activités (élicitation, traçabilité, validation, etc.) permettant de gérer et de maintenir au mieux l'ensemble des exigences. L'IE, jouant un rôle capital, est à la base du développement d'un projet. L'impact d'une erreur commise lors des activités relatives à l'IE est beaucoup plus important qu'une erreur commise en fin de projet. L'établissement d'exigences de qualité permet une bonne gestion de la totalité d'un projet. Cependant, l'IE présente certaines limitations notamment vis-à-vis de la stabilité des exigences, celles-ci étant constamment en évolution.

Les OSIE peuvent contribuer à une meilleure gestion de l'ensemble des activités de l'IE tout au long d'un processus de développement. Ils fournissent un meilleur suivi de l'évolution des exigences et améliorent la qualité des exigences. Cependant aucun OSIE commercial ne répond à l'ensemble des exigences (exigences dans un repository, représentations sous de multiples formalismes, etc.) définies dans la section 1.2.3.

## Chapitre 2

# L'outil de support *GenSpec*

Ce chapitre comporte :

1. une description de l'OSIE *GenSpec* (2.1) ;
2. une description du *cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles des OSIE*<sup>1</sup> [Mat05] (2.2) ;
3. une évaluation de l'OSIE *GenSpec* selon ce cadre d'évaluation (2.3).

Notons que la version de *GenSpec* analysée dans ce chapitre correspond à la version 6.4.15, précédant le développement de notre solution.

### 2.1 Description

Dans cette section, nous allons :

1. fournir une description générale de *GenSpec* (2.1.1) ;
2. décrire les différents types d'utilisateurs de *GenSpec* (2.1.2) ;
3. établir une liste des fonctionnalités proposées par *GenSpec* (2.1.3) ;
4. fournir un exemple mettant en évidence les principales fonctionnalités de *GenSpec* (2.1.4).

Notons que cette section est largement inspirée de Bujold [Buj04, Buj05].

#### 2.1.1 Généralités

*GenSpec* est un OSIE gratuit<sup>2</sup> développé à l'unité Conception-Automatismes de la société Hydro-Québec (HQ) à Montréal pour améliorer son processus d'IE. HQ est un important distributeur, transporteur et producteur d'électricité en Amérique du Nord [Hyd07a].

“[*GenSpec* permet] l'entrée des exigences dans une base de données, quelques vérifications automatiques de ces exigences et la génération de documents d'exigences. (...) *GenSpec* a été développé notamment à partir de normes internationales et de documents de la NASA et de la Défense des États-Unis d'Amérique :

1. *exigences hiérarchisées* ;
2. *une seule exigence par paragraphe* ;
3. *un numéro de référence unique par exigence* ;
4. *etc.*” [Buj05].

---

<sup>1</sup> *Frameworks for Functional and Non-functional RE-tools Requirements.*

<sup>2</sup> La dernière version de *GenSpec* est disponible à l'adresse : <ftp://ftp.hydro.qc.ca/expedition/GenSpec>

*GenSpec* a été développé pour répondre également aux problèmes majeurs des OSIE commerciaux [Buj04] :

*“System Architect, IRqA, Rational RequisitePro, etc. [sont des outils] puissants. Ils offrent (...) des facilités de traçabilité des exigences, de conception et même de génération de code logiciel. Par contre, de façon générale :*

1. *ils sont complexes : ils exigent tous une formation de plusieurs jours, voire plusieurs semaines. Ils exigent, de surcroît, plusieurs mois de pratique pour les maîtriser ;*
2. *ils offrent peu de flexibilité quant au format des documents générés : ils ne respectent pas le gabarit et même la langue des documents de l’entreprise ;*
3. *(...) il y est ardu de structurer les exigences, de les uniformiser et d’utiliser des renvois.*

*(...) [GenSpec] n’offre aucune facilité de conception ni de génération de code logiciel. Par contre, il est simple, offre beaucoup de flexibilité quant au format des documents générés [et permet de structurer et gérer les exigences].”*

### 2.1.2 Usagers

Il existe cinq types d’usagers distincts dans *GenSpec*, correspondant aux rôles qu’un acteur joue dans un projet. Ces différents types d’usagers sont :

- le **Visiteur** : il possède le droit de consulter et de générer des documents.
- le **Commentateur** : il a les mêmes droits qu’un Visiteur sauf qu’il peut apposer des commentaires aux exigences.
- le **Testeur** : il a les mêmes droits qu’un Visiteur mais il peut gérer les différentes fonctionnalités liées aux tests.
- l’**Utilisateur** : il a accès à l’ensemble des fonctionnalités de *GenSpec*, si ce n’est qu’il ne peut gérer les droits d’accès des différents usagers ni générer de versions officielles de documents d’exigences.
- l’**Administrateur** : il possède tous les droits vis-à-vis d’un projet *GenSpec* (il peut gérer les droits d’accès des différents usagers d’un projet, rédiger des exigences, générer un document d’exigences, etc.), il a accès à l’ensemble des fonctionnalités de *GenSpec*.

### 2.1.3 Fonctionnalités

Les fonctionnalités offertes par l’OSIE sont [Buj05] :

- la **rédaction des exigences** : elle permet à plusieurs usagers de rédiger ou modifier des exigences simultanément ;
- la **caractérisation des exigences** : elle permet d’associer à une exigence un ensemble d’attributs tel qu’un numéro identifiant, une priorité, un fichier joint, etc ;
- la **structuration et liaison des exigences** : elle permet d’établir des liens entre les exigences et de naviguer entre les exigences liées dans un même regroupement d’exigences (correspondant à un projet établis par *GenSpec*) ;
- l’**évaluation de conformité aux exigences** : elle permet de définir des procédures d’évaluation de conformité, ainsi que les résultats de ces procédures pour les exigences en vue de générer un rapport d’évaluation ;
- le **contrôle des exigences** : il permet d’éviter des incohérences (exigence sans énoncé, incohérence dans la hiérarchie des exigences, fonction sans intrant, etc.) entre les exigences et offrant un vérificateur d’ortographe et de grammaire ;
- la **normalisation des exigences** : elle propose à l’analyste des moyens permettant d’uniformiser les énoncés des exigences en fonctions de leur type ;

- la **configuration des documents d'exigences** : elle fournit de multiples options de génération de documents d'exigences ;
- la **gestion de l'historique des exigences** : elle permet de consulter les modifications apportées aux exigences et de revenir à une version antérieure de la totalité des exigences en cas de besoin.

Notons que certaines fonctionnalités essentielles telles qu'un **glossaire** permettant de définir les termes utilisés dans les exigences et une gestion simultanée de **plusieurs regroupements d'exigences** dans un seul projet établi par *GenSpec*, sont absentes des fonctionnalités. Le support, par une gestion automatique du tableau de conception et de la gestion de la **hiérarchie de composants<sup>3</sup> (sous-systèmes) du système**, est également absent des fonctionnalités offertes. Ces fonctionnalités ont cependant un intérêt grandissant pour l'unité Conception-Automatismes à HQ.

### 2.1.4 Exemple

Cette sous-section illustre les principales fonctionnalités décrites ci-dessus (cf. 2.1.3) au moyen de captures d'écran. Pour obtenir davantage d'informations, se référer à Bujold [Buj04, Buj05]. Notons que le chapitre 6 présente un exemple illustré se focalisant uniquement sur les nouvelles fonctionnalités ajoutées à *GenSpec*.

**Création d'un projet** La création d'un projet *GenSpec* se base sur un autre projet *GenSpec* représentant un modèle<sup>4</sup> pré-défini, correspondant à un canevas d'exigences. *GenSpec* propose quatre modèles de base :

1. IEEE 830 (avec ou sans exigences d'interfaces) ;
2. ISO/IEC 12207 (avec ou sans sauts de page dans les documents d'exigences).

La partie gauche de la figure 2.1 montre la sélection des modèles offerts à l'utilisateur servant de base pour un nouveau projet (Dans ce cas, l'utilisateur choisit le modèle IEEE 830 sans exigences d'interfaces). La partie droite de la figure permet à l'utilisateur de sélectionner l'emplacement du nouveau projet. La BD source, associée au modèle, est intégralement copiée à l'emplacement sélectionné par l'utilisateur pour définir le nouveau projet. L'utilisateur associe un nom au nouveau projet.

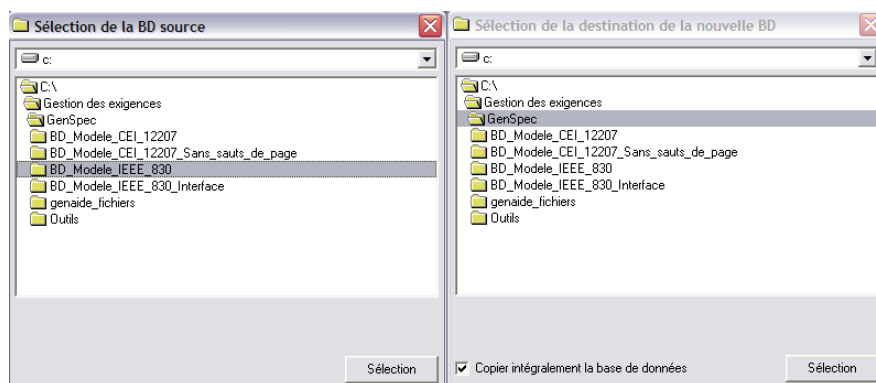


FIG. 2.1 – Création d'un projet *GenSpec*.

<sup>3</sup>Terminologie utilisée par l'unité Conception-Automatismes à HQ.

<sup>4</sup>Terminologie utilisée par HQ correspondant à un canevas d'exigences. Toute création d'un nouveau projet *GenSpec* se fait à partir d'un modèle.

**Ouverture d'un projet** Les exigences servant de base à cet exemple sont issues du projet *GenSpec* créé précédemment. Ce projet *GenSpec* est quelque peu commenté pour aider l'utilisateur à gérer les exigences, les liens entre les exigences, les procédures, etc.

La figure 2.2 présente la fenêtre d'ouverture de projets *GenSpec*. Un projet *GenSpec* correspond à une BD *Microsoft Access* reprenant l'ensemble des informations du projet. La figure 2.2 présente la sélection du projet *GenSpec* "Nouveau projet". La totalité de l'exemple se base sur ce projet *GenSpec*.

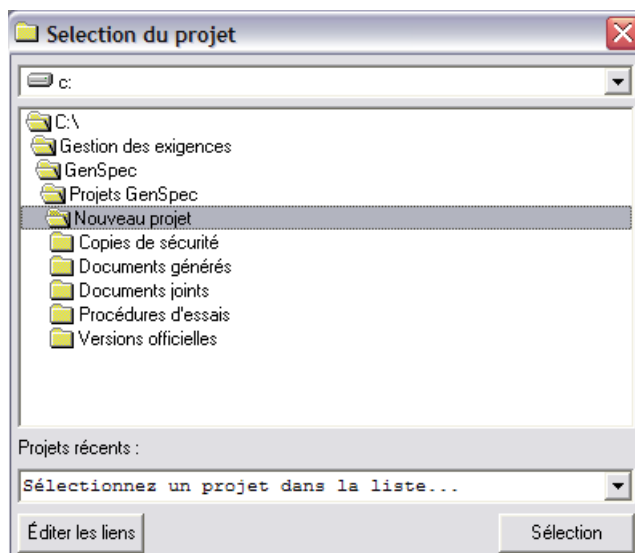


FIG. 2.2 – Ouverture d'un projet *GenSpec*.

**Rédaction d'exigences** La figure 2.3 présente la fenêtre de rédaction d'exigences.

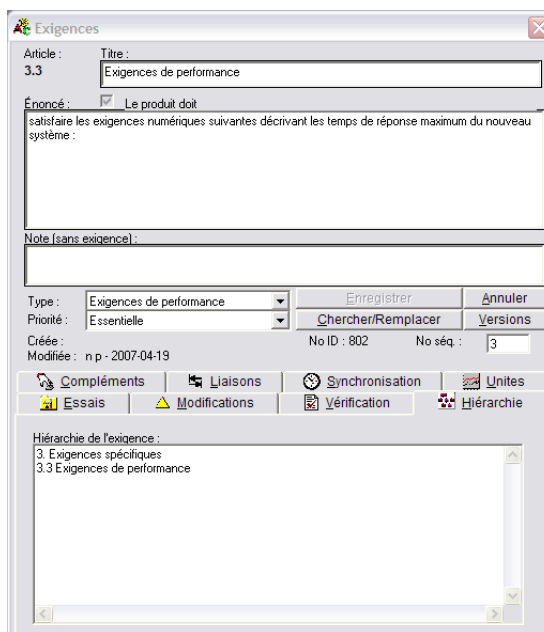


FIG. 2.3 – Fenêtre de rédaction des exigences.

Les informations suivantes y sont représentées :

- le **numéro d'article** “3.3” correspondant à l’emplacement de l’exigence dans la hiérarchie des exigences ;
- le **titre de l’exigence** “*Exigences de performance*” (utilisé notamment pour représenter l’exigence dans l’arbre hiérarchique des exigences) ;
- l’**énoncé de l’exigence** décrivant en quoi consiste l’exigence ;
- un champ destiné à apposer une **note** à l’exigence en cas de nécessité ;
- le **type** d’exigence (dans ce cas, il s’agit du type “*Exigences de performance*”) ;
- la **priorité** de l’exigence (“*Essentielle*” dans l’exemple) ;
- diverses informations telles que le **numéro identifiant**, les informations relatives à la **dernière modification** et le **numéro de séquence** ordonnant l’exigence par rapport à ses exigences soeurs ;
- plusieurs informations associées à l’exigence, notamment sa **hiérarchie** dans le projet en rapport avec les autres exigences qui la généralise ou la spécialise, ses éventuels **liens** avec d’autres exigences, son compte-rendu de **vérification**, etc.

**Liaisons entre exigences** La figure 2.4 montre les liaisons existantes d’une exigence (de type intrant dans cet exemple). L’exigence possédant ces liaisons est **utilisée par** deux autres exigences : “*Macro Fonction 1 (niv. 3 sans liens) : Fonction 1*” et “*Macro Fonction 1 (niv. 3 sans liens) : Fonction 2*”. Il s’agit donc des fonctions utilisant l’exigence comme intrant. De plus, l’exigence **provient** d’une autre exigence dont le titre est “*Interface avec l’exploitant : Intran 1*”, indiquant que l’exigence provient d’un autre projet établi par *GenSpec*. Notons que des **renvois** peuvent être utilisés afin de mettre en relation (sans lien formel) des exigences entre elles. Dans cette situation, il existe un lien avec l’exigence dont le titre est “*Sécurité*”.

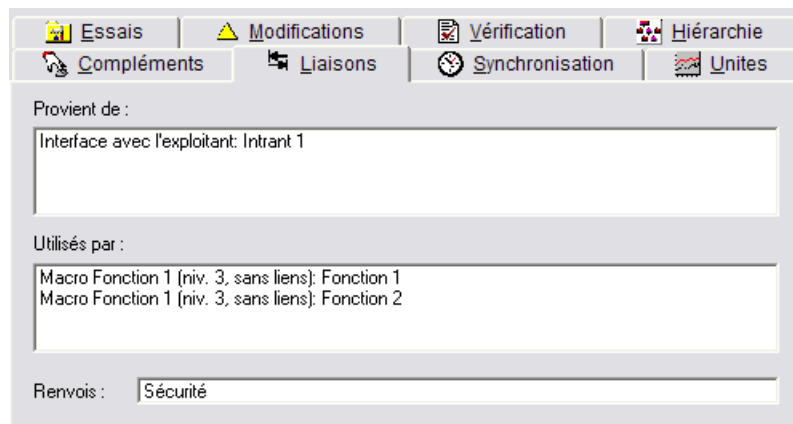


FIG. 2.4 – Les liaisons d’une exigence à d’autres exigences.

**Procédure d’essais (tests)** Un exemple de format de procédure d’essais est présenté dans la figure 2.5. Les procédures permettent de définir les étapes successives pour tester les exigences implémentées. La procédure présentée dans l’exemple préconise un ordre spécifique (“activer”, “mettre”, “réactiver” et “vérifier”) afin de tester l’exigence associée à cette procédure. Un **document joint** peut accompagner la procédure. Un **responsable** est assigné à une procédure et un numéro **ADP** (Avis De Problèmes) peut également accompagner la procédure, notamment pour faire référence à un problème défini, relatif à cette procédure.

FIG. 2.5 – Procédure d’essai relative à une exigence.

**Groupes d’exigences générées** L’utilisateur de *GenSpec* peut sélectionner les exigences à générer dans un document d’exigences par l’utilisation de groupes d’exigences générées. L’utilisateur peut allouer les exigences à générer à différents groupes d’exigences générées selon ses besoins. La figure 2.6 présente deux groupes définis par l’utilisateur : “Exigences de performance” et “Exigence fonctionnelles”. Par la sélection du groupe d’exigences “Exigences fonctionnelles”, tout document généré par *GenSpec* ne contiendra que les exigences allouées à ce groupe.

FIG. 2.6 – Groupes d’exigences à générer.

**Vérifications d'exigences** La figure 2.7 présente les diverses vérifications d'exigences disponibles. Six groupes de vérifications sont mis en évidence. Il s'agit de vérifications relatives :

1. **aux énoncés, notes et priorités** : elles assurent la complétude des informations relatives aux exigences et l'uniformisation des exigences ;
2. **aux liens intrants-extrants-fonctions** : elles assurent la cohérence et l'utilisation correcte des types d'exigences "intrans", "extrants" et "fonction" ;
3. **aux renvois** : elles vérifient que les renvois (références à des exigences ou à des documents d'exigences existants) sont mentionnés et respectent certaines contraintes ;
4. **à la hiérarchie** : elles vérifient la cohérence des liens de parenté entre les exigences ;
5. **aux procédures** : elles assurent la cohérence des procédures d'essais des exigences (qui décrivent la procédure pour vérifier les exigences implémentées) ;
6. **aux étapes de procédure** : elles s'assurent de la cohérence de chacune des étapes des procédures chargées de la vérification des exigences implémentées.

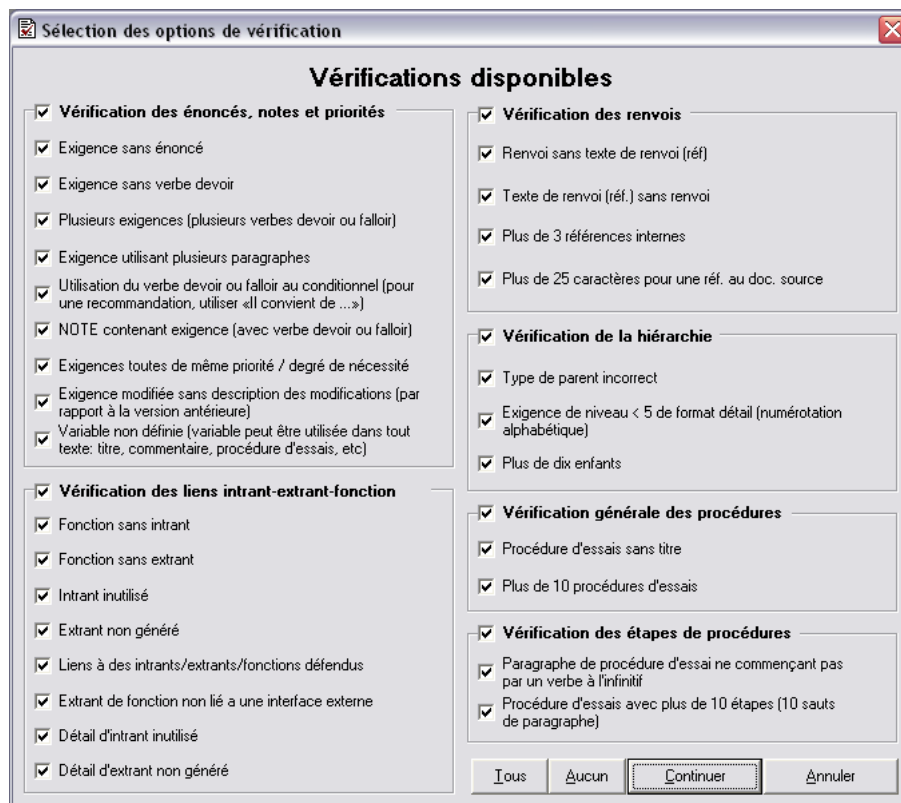


FIG. 2.7 – Ensemble des vérifications applicables aux exigences.

Ces vérifications assurent, avant toute génération de document d'exigences, qu'un certain niveau de cohérence et d'uniformisation est respecté. Un vérificateur d'orthographe et de grammaire est également disponible.



**Génération de documents** Un des avantages majeurs de *GenSpec* est la génération de documents d'exigences paramétrables. Pour chaque type de document généré par *GenSpec* ("Spécifications d'exigences", "Arbre hiérarchique des exigences", "Cahier des essais", etc.), un large panel d'options est proposé aux usagers : choix des éléments à intégrer dans les documents selon les six niveaux hiérarchiques des exigences proposés par l'outil. Se référer aux annexes E et F pour un aperçu du résultat de la génération de documents d'exigences.

La figure 2.8 présente les options de génération pour le document d'exigences de type "Spécifications d'exigences". Le document d'exigences contiendra entre autres, pour les six niveaux, l'énoncé et le degré de nécessité des exigences. Ne seront pas représentés, les commentaires éventuels et le statut de l'exigence (réussi/échoué). L'affichage du numéro d'exigence n'est souhaité qu'à partir du troisième niveau, et un saut de paragraphe avant l'énoncé n'est souhaité que pour les cinq premiers niveaux.

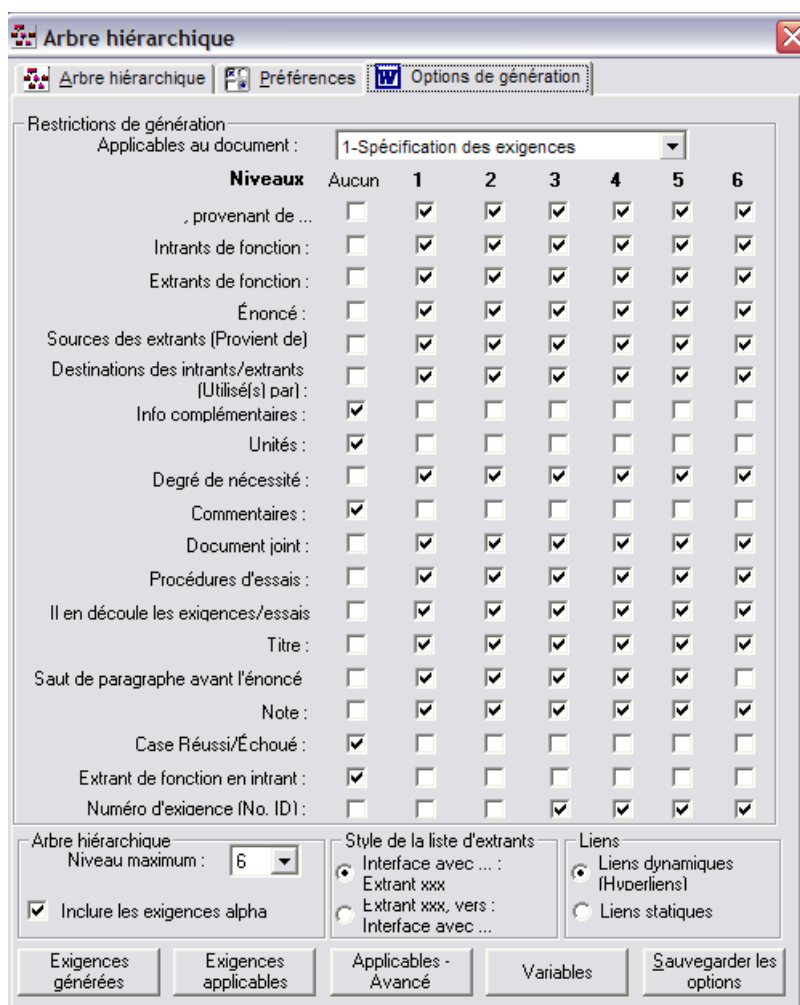


FIG. 2.8 – Options de génération.

## 2.2 Choix et définition du cadre d'évaluation

Dans cette section, nous allons :

1. justifier le choix d'un cadre d'évaluation (2.2.1) ;
2. définir le cadre d'évaluation choisi (2.2.2).

L'ensemble de cette section est largement inspirée de Matulevičius [Mat05].

### 2.2.1 Choix du cadre d'évaluation

Un cadre d'évaluation permet de définir des catégories d'exigences (ou d'activités) relatives aux OSIE. Des métriques sont alors établies sur ces catégories pour évaluer tout OSIE. Un cadre d'évaluation fournit une structure pour évaluer des OSIE et un processus de comparaison entre ces OSIE [Mat05].

Dans la sous-section 1.2.3, nous avons établi une liste non exhaustive d'exigences que devrait satisfaire tout OSIE. Ces exigences proviennent de différents cadres d'évaluation (*Wieger requirements* [Wie99], *Lang and Duggan requirements* [LD04], *INCOSE framework* [Inc07], *Priority-based framework* [HD97] et *Role-based framework* [IEE04]). Par cette démarche, nous souhaitons premièrement faire un état de l'art de plusieurs cadres d'évaluation en regroupant leurs exigences. Et deuxièmement, nous ne souhaitons pas limiter notre approche à un seul cadre d'évaluation pour définir les exigences de tout OSIE et pour évaluer *GenSpec*. C'est pourquoi, nous avons décidé d'appliquer le cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles des OSIE de Matulevičius [Mat05] à *GenSpec*. Le choix de ce cadre d'évaluation a été motivé par trois raisons :

1. Les cadres d'évaluation repris à la sous-section 1.2.3 présentent des faiblesses : cadres d'évaluation inappropriés pour une évaluation détaillée, oubli d'exigences, terminologie non définie entraînant des interprétations différentes des exigences des OSIE, aspects non-fonctionnels ignorés, etc.
2. Le cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles des OSIE est plus clair et plus structuré que les autres cadres d'évaluation et se prête mieux à la réalisation d'une analyse.
3. Le cadre d'évaluation pour les exigences fonctionnelles des OSIE est découpé selon les trois dimensions de l'IE (cf. FIG. 1.9) : “*Agreement*”, “*Representation*” et “*Specification*” définies à la sous-section 1.1.4.

### 2.2.2 Définition du cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles d'un OSIE

L'OSIE *GenSpec* est évalué par une application du cadre d'évaluation (*framework*) pour les exigences fonctionnelles et non-fonctionnelles d'un OSIE. Le but de ce cadre d'évaluation est :

“... to provide a skeleton structure for the RE-tool (Requirements Engineering tool) evaluation and comparison. An intension of the frameworks is not to be detailed, but to provide a list of RE-tool features which should be considered during the RE-tool assessment activities. Features represented in the frameworks describe the RE-tool characteristics which tool user could think it might be nice to have in an RE-tool. (...) They do not characterise how the RE-tool should implement feature in itself. The features are the basis for the RE-tool requirements, which should be specified in a requirements specification for the RE-tool selection” [Mat05].

Notons cependant que :

“When applying the evaluation frameworks, the user needs to consider the features and to decide whether they are important for the RE-tool or not. (...) The importance is recognised according to the user’s practice, experience and the organisational work practice” [Mat05].

L'évaluation de l'OSIE (*RE-tool*) dépend de l'importance relative que l'utilisateur accorde aux différentes fonctionnalités. Le résultat de l'application du cadre d'évaluation sur un OSIE peut entraîner des choix différents selon les utilisateurs.

Le cadre d'évaluation employé présente deux aspects :

1. **Exigences fonctionnelles** (*Functional requirements*) : Les exigences fonctionnelles pour un OSIE décrivent ce que l'OSIE devrait faire et quel comportement il devrait avoir. Ces exigences sont regroupées en trois dimensions reprises dans la figure 2.9, représentant le cadre d'évaluation d'exigences fonctionnelles pour un OSIE.

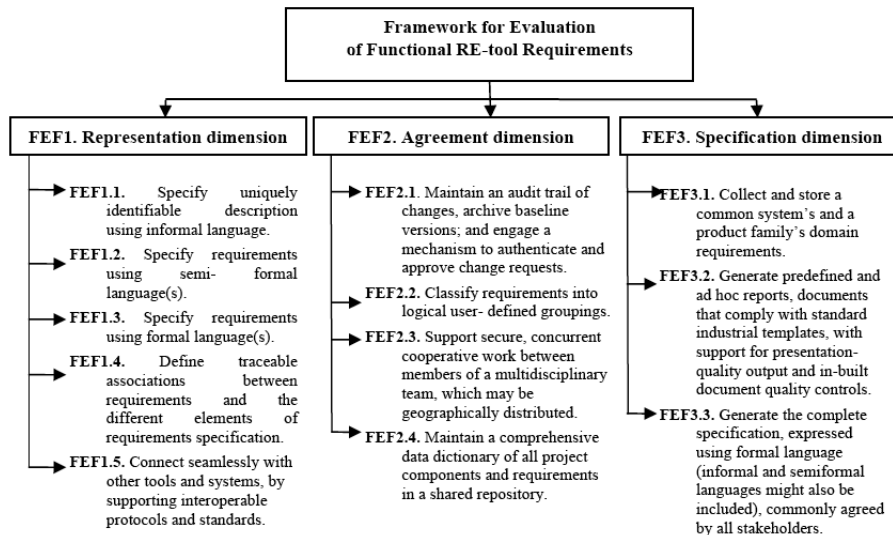


FIG. 2.9 – Cadre d'évaluation d'exigences fonctionnelles pour un OSIE.

Nous relevons 3 dimensions distinctes (cf. sous-section 1.1.4) :

- (a) **Représentation** (*Representation dimension*) : La dimension de la représentation des exigences analyse pour un OSIE :
  - le degré de formalisme ;
  - le mécanisme de traçabilité ;
  - le support des techniques de modélisation ;
  - l'import/export des représentations d'exigences ;
  - les associations avec les autres outils de développement.
- (b) **Accord** (*Agreement dimension*) : La dimension de l'accord, concernant les exigences, décrit pour un OSIE le degré d'accord entre les participants du projet.
- (c) **Spécification** (*Specification dimension*) : La dimension de la spécification des exigences analyse le degré de compréhension des exigences à un moment donné.

2. **Exigences non-fonctionnelles** (*Non-functional requirements*) : Les exigences non-fonctionnelles pour les OSIE décrivent les propriétés que l'OSIE doit avoir (*exemple* : "l'OSIE doit être sûr") plutôt qu'un comportement spécifique. La figure 2.10 présente le cadre d'évaluation d'exigences non-fonctionnelles pour un OSIE. Elle met en évidence les différentes dimensions de ce cadre d'évaluation.

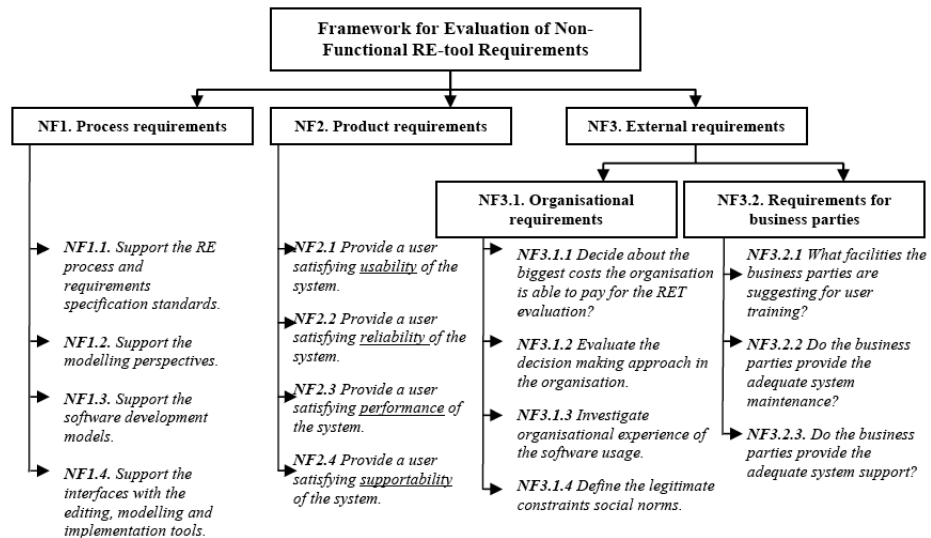


FIG. 2.10 – Cadre d'évaluation d'exigences non-fonctionnelles pour un OSIE.

Les exigences non-fonctionnelles sont subdivisées en trois dimensions distinctes :

- (a) **Exigences du processus** (*Process requirements*) : Les exigences du processus sont les contraintes placées, d'une part sur le processus de développement de l'utilisateur, d'autre part sur les pratiques de travail. Les exigences du processus caractérisent le processus d'IE dans l'organisation où travaille l'utilisateur de l'OSIE.
- (b) **Exigences du produit** (*Product requirements*) : Les exigences du produit sont les traits qui spécifient les caractéristiques désirées chez un OSIE. Nous retrouvons quatre exigences du produit :
- **L'utilisabilité** (*Usability*) : analyse la facilité d'apprentissage de l'OSIE et la facilité d'utilisation de l'OSIE au jour le jour. L'utilisabilité reprend la capacité de compréhension, d'apprentissage, d'utilisation et d'appréciation des utilisateurs quand l'OSIE est utilisé sous certaines conditions spécifiques.
  - **La fiabilité** (*Reliability*) : analyse la fréquence du mauvais fonctionnement de l'OSIE et décrit le pourcentage de temps où l'OSIE est disponible.
  - **La performance** (*Performance*) : définit la rapidité de réponse de l'OSIE, les ressources qu'il utilise et sa capacité à fournir les performances requises selon l'utilisation des ressources sous certaines conditions. Cela concerne deux critères : l'efficacité de stockage et l'efficacité d'exécution.
  - **L'extensibilité** (*Supportability*) : définit comment l'OSIE doit être étendu par de nouvelles fonctionnalités, par l'adoption de standards, par des perspectives de modélisation et par une relation de l'OSIE avec d'autres outils logiciels.

- (c) **Exigences externes** (*External requirements*) : représentent les facteurs sociaux et économiques. Deux dimensions d'exigences sont considérées :
- i. **Exigences organisationnelles** : Les exigences organisationnelles caractérisent les OSIE d'un point de vue coûts et problèmes business.
  - ii. **Exigences pour les groupes business** : Les exigences pour les groupes business sont liées à la performance et à la fiabilité des vendeurs et des consultants de l'OSIE. Cela inclut l'infrastructure et la stabilité, la réputation du vendeur, les autres clients et leur expérience professionnelle.

Pour plus d'informations sur le cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles des OSIE, reportez-vous à Matulevičius [Mat05].

## 2.3 Application du cadre d'évaluation à *GenSpec*

Dans cette section, nous allons :

1. appliquer en détail le cadre d'évaluation d'exigences fonctionnelles à *GenSpec* (2.3.1) ;
2. appliquer en détail le cadre d'évaluation d'exigences non-fonctionnelles à *GenSpec* (2.3.2) ;
3. synthétiser les résultats de l'application des deux cadres d'évaluations à *GenSpec* (2.3.3).

L'annexe L définit les références aux exigences employées (FEF...) et fournit une description détaillée de l'ensemble des exigences à vérifier lors de l'application du cadre d'évaluation.

Rappelons que l'application du cadre d'évaluation s'effectue sur la version précédant le développement de nos fonctionnalités (version 6.4.15 de *GenSpec*).

### 2.3.1 Application du cadre d'évaluation d'exigences fonctionnelles à *GenSpec*

**Dimension de la représentation** Le tableau 2.1 reprend l'ensemble des exigences correspondant à la dimension de la représentation.

TAB. 2.1: Dimension de la représentation.

Caractéristiques	Activités Comment est-ce que l'OSIE...	<i>GenSpec</i>
<b>FEF1.1.</b> Spécifie une description identifiable unique en utilisant un langage informel.	<b>FEF 1.1.1</b> fournit une description en LN.	<b>Oui.</b> <i>GenSpec</i> ne permet la rédaction des exigences qu'en LN. Le langage employé peut être également appelé langage professionnel, certains termes étant associés uniquement au domaine d'application.
	<b>FEF 1.1.2</b> permet la spécification d'un identifiant unique (ID) pour chaque exigence.	<b>Oui.</b> <i>GenSpec</i> associe un identifiant unique à chaque exigence. Cet identifiant est visible par l'interface graphique. L'identifiant est généré automatiquement lors de la création de l'exigence.
	<b>FEF 1.1.3</b> permet l'importation des exigences et de leur description à partir d'un document textuel.	<b>Partiellement.</b> <i>GenSpec</i> ne comporte aucune fonctionnalité d'importation d'exigences à partir d'un document texte. Au mieux, un copier-coller peut être utilisé pour chaque exigence du document original vers <i>GenSpec</i> .
<b>FEF1.2.</b> Spécifie les exigences en utilisant un (des) langage(s) semi-formel(s).	<b>FEF 1.2.1</b> fournit des outils pour la description en langage semi-formel.	<b>Non.</b> <i>GenSpec</i> ne fournit pas de support pour les langages semi-formels.
	<b>FEF 1.2.2</b> fournit la traçabilité en avant/arrière entre les descriptions formelles, semi-formelles et informelles.	<b>Partiellement.</b> <i>GenSpec</i> ne fournit pas de support pour les langages semi-formels. Cependant, un lien peut-être établi entre toute exigence textuelle et un document externe à <i>GenSpec</i> (comportant par exemple l'exigence sous une autre représentation) en joignant ce document comme fichier joint de l'exigence.

TAB. 2.1: Dimension de la représentation.

<b>FEF1.3.</b> Spécifie les exigences en utilisant un (des) langage(s) formel(s).	<b>FEF 1.3.1</b> fournit des outils pour la description en langage formel.	<b>Non.</b> <i>GenSpec</i> ne fournit pas de support pour les langages formels.
	<b>FEF 1.3.2</b> fournit la traçabilité en avant/arrière entre les descriptions formelles, semi-formelles et informelles.	<b>Partiellement.</b> <i>GenSpec</i> ne fournit pas de support pour les langages formels. Cependant, un lien peut-être établi entre toute exigence textuelle et un document externe à <i>GenSpec</i> (comportant par exemple l'exigence sous une autre représentation) en joignant ce document comme fichier joint de l'exigence.
<b>FEF1.4.</b> Définit des associations traçables entre les exigences et différents éléments du document d'exigences.	<b>FEF 1.4.1</b> fournit des fonctions pour tester la traçabilité entre la description informelle, semi-formelle et formelle des exigences.	<b>Non.</b> <i>GenSpec</i> ne fournit pas de fonctions pour tester la traçabilité entre les différentes représentations des exigences. <i>GenSpec</i> ne supporte pas différentes représentations d'une même exigence, excepté la liaison avec un fichier externe qui peut contenir une autre représentation. Cependant, il n'y a pas de méthodes de vérification de traçabilité associé à une autre représentation de l'exigence avec le fichier externe.
	<b>FEF 1.4.2</b> crée des relations traçables parents-enfants entre les exigences.	<b>Oui.</b> <i>GenSpec</i> fournit un mécanisme de traçabilité parent-enfant entre les exigences. Ce mécanisme est le fondement même du fonctionnement de <i>GenSpec</i> , les exigences étant sous forme d'un arbre hiérarchique. Chaque exigence possède une exigence parente, excepté pour l'exigence racine. L'établissement des exigences se fait par une spécialisation de l'exigence parente.
	<b>FEF 1.4.3</b> maintient des relations traçables entre exigences de même niveau <sup>5</sup> .	<b>Oui.</b> <i>GenSpec</i> supporte l'établissement de traçabilité entre exigences de même niveau hiérarchique (niveau d'abstraction). Cependant, aucun support de traçabilité n'est fourni entre deux projets <i>GenSpec</i> distincts. Notons que <i>GenSpec</i> permet l'établissement de liens en fonction du type des exigences impliquées dans la relation indépendamment du niveau hiérarchique de l'exigence.
	<b>FEF 1.4.4</b> maintient une relation traçable entre les diverses informations en relation.	<b>Oui.</b> <i>GenSpec</i> fournit un support de la traçabilité entre les exigences et les différentes informations associées. <i>GenSpec</i> permet de joindre tout document externe à une exigence.

<sup>5</sup>En anglais : "peer-to-peer traceable relation".

TAB. 2.1: Dimension de la représentation.

	<b>FEF 1.4.5</b> maintient la traçabilité avant/arrière entre la source des exigences, les exigences et la conception.	<b>Partiellement.</b> <i>GenSpec</i> permet le maintien de la traçabilité en avant et en arrière en joignant des documents externes à une exigence. Par conséquent, le lien d'une exigence avec sa source et sa conception peut être établi. Cependant, l'établissement et le support de cette traçabilité n'est pas automatisé.
<b>FEF1.5.</b> Connexion de l'OSIE avec les autres outils et systèmes, en supportant des protocoles et standards interopérables.	<b>FEF 1.5.1</b> permet l'importation/l'exportation de la description d'exigences à partir/vers des documents texte.	<b>Partiellement.</b> <i>GenSpec</i> permet de générer de manière automatisée différents documents <i>Microsoft Word</i> : <ul style="list-style-type: none"> <li>– spécification des exigences ;</li> <li>– arbre hiérarchique des exigences ;</li> <li>– références aux exigences sources externes ;</li> <li>– liste des modification courantes ;</li> <li>– historique des modifications ;</li> <li>– cahier des essais ;</li> <li>– tableau exigences avec l'ID de l'exigence.</li> </ul> <i>GenSpec</i> ne supporte pas l'importation d'exigences à partir de documents textuels ou graphiques. Au mieux, un copier-coller peut être fait pour chaque exigence du document original vers <i>GenSpec</i> .
	<b>FEF 1.5.2</b> permet l'importation/l'exportation de la description d'exigences à partir/vers des documents graphiques.	<b>Non.</b> <i>GenSpec</i> ne supporte par l'importation ou l'exportation des exigences sous un format graphique.



**Dimension de l'accord** Le tableau 2.2 reprend l'ensemble des exigences correspondant à la dimension de l'accord.

TAB. 2.2: Dimension de l'accord.

Caractéristiques	Activités Comment est-ce que l'OSIE...	GenSpec
<b>FEF2.1.</b> Maintenir une trace des changements, archiver les versions et établir un mécanisme pour authentifier et approuver les requêtes de changements.	<b>FEF 2.1.1</b> maintient l'authentification des usagers vis-à-vis du système (c'est-à-dire nom d'utilisateur, mot de passe).	<b>Oui.</b> <i>GenSpec</i> requiert l'authentification de l'utilisateur pour son utilisation par un mécanisme de nom d'utilisateur et de mot de passe.
	<b>FEF 2.1.2</b> regroupe les usagers dans différents groupes.	<b>Partiellement.</b> <i>GenSpec</i> ne permet pas de regrouper les différents usagers en groupes distincts. Cependant, chaque usager possède une fonction déterminée dans un projet. Différents groupements d'usagers peuvent alors être considérés en accord avec leur fonction.
	<b>FEF 2.1.3</b> crée différentes vues (selon les documents, exigences, attributs) pour les différents groupes de parties prenantes.	<b>Partiellement.</b> <i>GenSpec</i> ne permet pas de créer différentes vues selon les acteurs du projet. Cependant, <i>GenSpec</i> propose des fonctionnalités différentes selon le type d'usager. <i>Exemple</i> : La gestion des usagers de <i>GenSpec</i> est une fonctionnalité réservée uniquement à l'administrateur du projet.
	<b>FEF 2.1.4</b> enregistre l'accord / la raison / la discussion / la négociation / les changements / l'historique des exigences et comment cela a été réalisé.	<b>Partiellement.</b> <i>GenSpec</i> ne permet pas de définir la raison, l'accord, les discussions, les négociations, etc. concernant une exigence. <i>GenSpec</i> ne permet pas d'exprimer la raison d'être d'une exigence. Cependant, <i>GenSpec</i> permet de commenter tout exigence. Ces commentaires peuvent alors reprendre un ensemble d'informations. <i>GenSpec</i> possède également une gestion simple de l'historique par la sauvegarde générale d'un projet à la demande de l'usager.

TAB. 2.2: Dimension de l'accord.

	<b>FEF 2.1.5</b> appelle les anciennes descriptions / versions des exigences et les enregistre dans un historique.	<b>Partiellement.</b> <i>GenSpec</i> fournit une gestion simple, générale et non-automatisée de l'historique de la totalité des exigences. <i>GenSpec</i> permet la sauvegarde des différentes versions de la totalité des exigences à la demande de l'utilisateur. Cependant, il ne permet pas une gestion individuelle des exigences. Sauf sauvegarde préalable par l'utilisateur, il est impossible de revenir à une ancienne version des exigences.
<b>FEF2.2.</b> Classifier les exigences en regroupements définis par les utilisateurs.	<b>FEF 2.2.1</b> permet la spécification des attributs / propriétés d'une exigences.	<b>Partiellement.</b> <i>GenSpec</i> permet la définition de la priorité des exigences : "Essentielle", "Complémentaire" ou "Optionnelle". <i>GenSpec</i> permet également la définition du type de l'exigence. <i>Exemple</i> : Exigence de performance, Exigence fonctionnelle, Intransit, Extant, etc.
	<b>FEF 2.2.2</b> fournit un tri selon les différents attributs et propriétés.	<b>Non.</b> <i>GenSpec</i> ne permet pas le tri des exigences selon leurs propriétés ou attributs.
	<b>FEF 2.2.3</b> fournit un filtrage selon les différents attributs et propriétés.	<b>Non.</b> <i>GenSpec</i> ne permet pas de filtrer les exigences selon leurs propriétés ou attributs.
<b>FEF2.3.</b> Supporter un travail protégé, concurrent et coopératif entre les différents membres d'une équipe multidisciplinaire qui peut être distribuée géographiquement.	<b>FEF 2.3.1</b> fournit une plate-forme indépendante de l'interface pour les usagers distribués géographiquement.	<b>Non.</b> <i>GenSpec</i> ne fournit pas une plate-forme permettant aux usagers géographiquement distribués d'accéder aux exigences.
	<b>FEF 2.3.2</b> permet de faire une copie pour une modification en différents niveaux d'abstraction (document, exigence) d'une version, déjà approuvée, d'une description des exigences.	<b>Partiellement.</b> <i>GenSpec</i> supporte la copie la totalité des exigences d'une version approuvée. Ces copies peuvent subir les mêmes modifications que tout projet <i>GenSpec</i> standard. Cependant, ce mécanisme de copie concerne la totalité des exigences du projet <i>GenSpec</i> et non pas les exigences prises individuellement.

TAB. 2.2: Dimension de l'accord.

	<b>FEF 2.3.3</b> fournit un cycle de modifications approuvées, pour les négociations de changement et d'accord, avant d'établir ces modifications dans le repository commun.	<b>Non.</b> <i>GenSpec</i> ne supporte pas un cycle de multiples changements des exigences sans modification à la BD (repository) pour arriver à un accord. <i>GenSpec</i> enregistre directement les changements apportés aux exigences dans la BD.
<b>FEF2.4.</b> Maintenir un dictionnaire de données complet pour tous les composants du projet et les exigences dans un repository partagé.	<b>FEF 2.4.1</b> fournit un simple repository ou un dictionnaire de données et de concepts.	<b>Non.</b> <i>GenSpec</i> ne permet pas de définir les termes importants employés dans les exigences. <i>GenSpec</i> ne possède pas de dictionnaire de données ou de concepts.
	<b>FEF 2.4.2</b> fournit un dictionnaire de données séparé pour les utilisateurs techniques et non-techniques.	<b>Non.</b> <i>GenSpec</i> ne possède pas de dictionnaire.
	<b>FEF 2.4.3</b> fournit l'aide du système pour les usagers.	<b>Oui.</b> <i>GenSpec</i> fournit à l'utilisateur de l'aide concernant son fonctionnement par le biais d'un document HTML reprenant une description du fonctionnement de l'OSIE.

**Dimension de spécification.** Le tableau 2.3 reprend l'ensemble des exigences correspondant à la dimension de la spécification.

TAB. 2.3: Dimension de spécification.

Caractéristiques	Activités Comment est-ce que l'OSIE...	GenSpec
<b>FEF3.1.</b> Collecte et stocke des exigences communes au système et au domaine de la famille de produits.	<b>FEF 3.1.1</b> active la sélection et l'extraction d'exigences communes du domaine et les exigences qui différencient le système dans la ligne de produit.	<b>Oui.</b> <i>GenSpec</i> permet de réutiliser les exigences d'un projet existant (par exemple un modèle) pour servir de base à un autre projet. Notons que la fonctionnalité "copier-coller" permet de réutiliser des exigences existantes appartenant à un autre projet <i>GenSpec</i> . Dans ce cas, un projet reprenant les exigences communes du domaine peut être établi pour être réutilisé dans différents projets.
	<b>FEF 3.1.2</b> incorpore les exigences dans un canevas d'exigences.	<b>Partiellement.</b> <i>GenSpec</i> permet l'incorporation d'exigences dans un projet donné par un mécanisme de "copier-coller" entre projets <i>GenSpec</i> . Par la gestion des exigences sous forme d'arbre hiérarchique, il est également possible d'intégrer un ensemble d'exigences hiérarchisées dans un projet (toujours par "copier-coller"). Cependant, <i>GenSpec</i> ne permet pas, par la contrainte d'exigence racine unique pour la hiérarchie d'exigences, de reprendre l'ensemble des exigences dans le cas d'un projet de taille importante. Plusieurs projets établis par <i>GenSpec</i> sont alors nécessaires pour reprendre la totalité des exigences d'un projet.
	<b>FEF 3.1.3</b> adapte / diffuse les changements dans un domaine d'exigences aux projets concrets à l'intérieur du domaine.	<b>Non.</b> <i>GenSpec</i> ne fournit pas un mécanisme de diffusion du changement dans les exigences du domaine lorsqu'une exigence associée est modifiée. Notons cependant qu'une modification apportée à une exigence s'applique automatiquement à toute exigence du même projet <i>GenSpec</i> qui est synchronisée avec cette dernière.
	<b>FEF 3.1.4</b> fournit une comparaison de la faisabilité des exigences du domaine.	<b>Non.</b> <i>GenSpec</i> ne permet pas une comparaison de la faisabilité des exigences du domaine.

TAB. 2.3: Dimension de spécification.

<p><b>FEF3.2.</b> Génère des rapports prédéfinis et <i>ad hoc</i> (documents qui répondent aux modèles standards industriels) avec un support pour la sortie de la qualité de présentation et les documents de contrôle de qualité intégrés.</p>	<p><b>FEF 3.2.1</b> fournit des assistants pour la génération de rapports.</p>	<p><b>Partiellement.</b> <i>GenSpec</i> permet d'établir et d'imprimer des rapports concernant le projet courant. Cependant, ces rapports sont établis à partir de types de documents à sélectionner et d'options s'appliquant à ce type de documents.</p>
	<p><b>FEF 3.2.2</b> fournit la possibilité d'imprimer des rapports selon les vues et le tri.</p>	<p><b>Oui.</b> <i>GenSpec</i> permet de sélectionner les exigences et les informations relatives à ces exigences à intégrer dans les documents d'exigences.</p>
	<p><b>FEF 3.2.3</b> fournit la possibilité d'imprimer les résultats du raisonnement, du brainstorming<sup>6</sup>, etc.</p>	<p><b>Non.</b> <i>GenSpec</i> ne permet pas d'établir et d'imprimer des rapports concernant la raison, les négociations, etc. car il ne gère pas l'ensemble de ces informations.</p>
	<p><b>FEF 3.2.4</b> fournit une comparaison des techniques pour la vérification d'erreurs.</p>	<p><b>Partiellement.</b> <i>GenSpec</i> supporte de nombreux mécanismes de vérifications. Il fournit essentiellement un mécanisme de détection d'erreurs plutôt que de prévention et correction. C'est à l'utilisateur de corriger les erreurs détectées. <i>GenSpec</i> permet de vérifier notamment :</p> <ul style="list-style-type: none"> <li>– la grammaire et l'orthographe des exigences ;</li> <li>– si les exigences possèdent un énoncé ;</li> <li>– si l'énoncé de l'exigence ne contient qu'un seul paragraphe.</li> </ul>
<p><b>FEF3.3.</b> Génère le document d'exigences complet, communément approuvé par l'ensemble des parties prenantes, exprimé en langage formel (langages semi-formels et informels devraient aussi être inclus).</p>	<p><b>FEF 3.3.1</b> correspond aux standards de la documentation de logiciels.</p>	<p><b>Oui.</b> <i>GenSpec</i> permet la génération de plusieurs types de documents en LN. La génération de documents d'exigences par <i>GenSpec</i> peut répondre à différents standards internationaux (correspondant aux canevas d'exigences utilisés lors de l'établissement des exigences).</p>

<sup>6</sup>Terme anglais référant à une technique permettant d'apporter des solutions à un problème donné grâce à un recoupement d'idées effectué par le groupe de travail. [wik07a]

TAB. 2.3: Dimension de spécification.

<b>FEF 3.3.2</b> correspond aux standards définis par l'organisation.	<b>Oui.</b> <i>GenSpec</i> permet la création de modèles d'exigences. Une fois un modèle créé, il peut être réutilisé pour établir les exigences d'un projet, chaque projet <i>GenSpec</i> pouvant servir de base à l'établissement de tout nouveau projet. <i>GenSpec</i> fournit la possibilité de définir les standards organisationnels internes pour l'établissement des exigences.
<b>FEF 3.3.3</b> supporte les langages formels pour un document d'exigences communément approuvé.	<b>Non.</b> <i>GenSpec</i> ne supporte pas les langages formels.

### 2.3.2 Application du cadre d'évaluation d'exigences non-fonctionnelles à GenSpec

**Exigences du processus** Le tableau 2.4 reprend les exigences du processus.

TAB. 2.4: Exigences du processus.

Caractéristiques L'OSIE devrait...	<i>GenSpec</i>
<b>NF1.1.</b> supporter les standards de spécification des exigences et d'IE sélectionnés.	<b>Oui.</b> <i>GenSpec</i> supporte de base les normes IEEE std 830 et ISO/IEC 12207. Cependant, <i>GenSpec</i> peut établir aisément de nouveaux modèles d'exigences qui peuvent être réutilisés pour tout autre projet.
<b>NF1.2.</b> supporter les perspectives de modélisation sélectionnées.	<b>Non.</b> <i>GenSpec</i> ne supporte pas différents types de modélisation des exigences. Les exigences peuvent uniquement être rédigées sous forme de LN.
<b>NF1.3.</b> supporter les types de modèle de développement logiciel.	<b>Partiellement.</b> <i>GenSpec</i> supporte les différents types de modèle de développement logiciel indifféremment. <i>GenSpec</i> ne présente pas de particularités selon le modèle employé pour le développement.
<b>NF1.4.</b> supporter les interfaces avec les outils d'édition, de modélisation et d'implémentation.	<b>Non.</b> <i>GenSpec</i> ne possède aucune interface avec d'autres logiciels supportant les autres phases de développement.

**Exigences du produit** Le tableau 2.5 reprend l'ensemble les exigences du produit.

TAB. 2.5: Exigences du produit.

Caractéristiques L'OSIE devrait...	<i>GenSpec</i>
<b>NF2.1.</b> Fournir une utilisabilité satisfaisante à l'utilisateur du système.	<b>Partiellement.</b> <ul style="list-style-type: none"> <li>– <i>GenSpec</i> est un OSIE simple d'utilisation. Le but de l'OSIE, de par sa simplicité d'utilisation et les fonctionnalités fournies, apparaît clairement aux utilisateurs. Toutefois, l'utilisateur peut rencontrer quelques difficultés lors de l'apprentissage de l'OSIE. La logique d'utilisation de <i>GenSpec</i> est parfois difficile à comprendre. Cependant, l'aide de <i>GenSpec</i> permet de répondre à de nombreuses questions en matière de fonctionnement.</li> <li>– <i>GenSpec</i> est efficace pour établir les exigences d'un nouveau système à développer. Cependant, les utilisateurs rencontrent souvent des difficultés de stabilité du système causées par divers bugs.</li> </ul>
<b>NF2.2.</b> Fournir une fiabilité satisfaisante à l'utilisateur du système.	<b>Partiellement.</b> <ul style="list-style-type: none"> <li>– <i>GenSpec</i> présente des bugs provoquant l'arrêt total de l'OSIE ou empêchant de réaliser certaines tâches de manière efficace (par exemple, la génération de documents).</li> <li>– <i>GenSpec</i> ne fournit pas à l'utilisateur les informations nécessaires concernant l'origine des erreurs système.</li> </ul>
<b>NF2.3.</b> Fournir une performance satisfaisante à l'utilisateur du système.	<b>Oui.</b> Les performances de <i>GenSpec</i> sont satisfaisantes. Les performances dépendent de la localisation de la BD du projet <i>GenSpec</i> . Si la BD est locale, les différentes opérations (et donc accès à la BD) se font plus rapidement que si la BD était sur un serveur (de l'entreprise).
<b>NF2.4.</b> Fournir une extensibilité satisfaisante à l'utilisateur du système.	<b>Oui.</b> <ul style="list-style-type: none"> <li>– <i>GenSpec</i> est étendu par l'ajout de nouvelles fonctionnalités répondant aux nouveaux besoins des utilisateurs, ces nouvelles fonctionnalités étant développées par des stagiaires successifs. Lors de ces ajouts de fonctionnalités, des corrections sont également effectuées pour stabiliser l'OSIE.</li> <li>– L'extensibilité de <i>GenSpec</i> est essentielle pour toujours arriver à l'établissement d'exigences de meilleure qualité.</li> </ul>



**Exigences externes organisationnelles** Le tableau 2.6 reprend les exigences externes organisationnelles.

TAB. 2.6: Exigences externes organisationnelles.

<b>Caractéristiques L'organisation devrait...</b>	<b><i>GenSpec</i></b>
<b>NF3.1.1.</b> Décider les coûts les plus importants que l'organisation est capable de payer pour l'évaluation d'un OSIE.	<ul style="list-style-type: none"> <li>– <i>GenSpec</i> est un OSIE <b>gratuit</b>. Il ne nécessite pas de matériel spécifique pour son fonctionnement.</li> <li>– L'extensibilité de <i>GenSpec</i> est gratuite. Toute nouvelle fonctionnalité développée est incluse dans la nouvelle version gratuite de <i>GenSpec</i>.</li> </ul>
<b>NF3.1.2.</b> Evaluer l'approche de prise de décisions dans l'organisation.	HQ utilise l'OSIE <i>GenSpec</i> qu'ils ont choisi de développer.
<b>NF3.1.3.</b> Examiner l'expérience organisationnelle de l'utilisation du logiciel.	Chez HQ, il n'y a pas de réelle évaluation de <i>GenSpec</i> . L'OSIE est développé pour répondre à leurs besoins. Les usagers de <i>GenSpec</i> permettent d'évaluer son fonctionnement et les fonctionnalités qu'il propose de par leur utilisation. Ils établissent les nouveaux besoins quant aux fonctionnalités à développer.
<b>NF3.1.4.</b> Définir les contraintes de normes sociales légitimes.	L'utilisation exclusive du français fait partie de la culture d'entreprise à HQ. <i>GenSpec</i> est un OSIE totalement en français.

**Exigences externes pour les groupes business** Le tableau 2.7 reprend les exigences externes non-fonctionnelles pour les groupes business.

TAB. 2.7: Exigences externes non-fonctionnelles pour les groupes Business.

Caractéristiques	<i>GenSpec</i>
<b>NF3.2.1.</b> Quels équipements suggèrent les groupes business pour la formation des usagers ?	<ul style="list-style-type: none"> <li>– Il n'existe pas de séminaires d'entraînement pour l'utilisation de <i>GenSpec</i> pour les entreprises autres que HQ. HQ n'assume aucun suivi ni responsabilité vis-à-vis de l'utilisation de <i>GenSpec</i>. Cependant, au sein même d'HQ, des formations sont prévues pour l'utilisation de <i>GenSpec</i>.</li> <li>– HQ ne fournit aucun support quant à l'utilisation de <i>GenSpec</i> par d'autres entreprises. Cependant, le responsable de <i>GenSpec</i> (M. René Bujold) est joignable pour toute question. Un groupe <i>MSN</i> est disponible pour toutes discussions concernant <i>GenSpec</i>.</li> </ul>
<b>NF3.2.2.</b> Est-ce que les groupes business fournissent la maintenance système adéquate ?	<ul style="list-style-type: none"> <li>– La maintenance apportée à <i>GenSpec</i> est très variable. <i>GenSpec</i> évolue par l'ajout de nouvelles fonctionnalités selon l'importance des besoins et par la correction de bugs et autres problèmes.</li> <li>– Si un problème est rencontré avec <i>GenSpec</i>, un site internet ainsi qu'un groupe <i>MSN</i> permet aux usagers de <i>GenSpec</i> de donner un retour d'informations sur les anomalies qu'ils rencontrent lors de son utilisation et peuvent y définir une priorité ainsi qu'un responsable. Ces anomalies sont prises en compte et corrigées. Cependant, il n'y a pas d'échéance concernant la correction des anomalies rencontrées.</li> </ul>
<b>NF3.2.3.</b> Est-ce que les groupes business fournissent le support de système adéquat ?	<ul style="list-style-type: none"> <li>– <i>GenSpec</i> est un OSIE gratuit. Aucun support n'est fourni pour les utilisateurs de <i>GenSpec</i> en dehors d'HQ.</li> <li>– Les nouvelles versions de <i>GenSpec</i> résolvent différents types de problèmes par rapport aux anciennes versions. Parfois, ces nouvelles versions permettent de corriger des anomalies et/ou de répondre à des nouveaux besoins. Ces nouveaux besoins répondant à quelques problèmes des versions précédentes.</li> </ul>

### 2.3.3 Synthèse de l'évaluation

Cette sous-section reprend les points forts et points faibles les plus marquants de l'évaluation de *GenSpec*.

**Points forts** *GenSpec* :

- est **simple** d'utilisation.
- associe un **identifiant unique** à chaque exigence.
- permet de **ré-utiliser les exigences** provenant d'un autre projet *GenSpec* par copier-coller d'exigences.
- permet de **joindre des documents à une exigence**. Les documents associés peuvent être par exemple : une modélisation de l'exigence dans une autre formalisme, un document lié à la traçabilité de l'exigence (*Exemple* : Document établissent son origine), etc.
- possède un **mécanisme de traçabilité entre les exigences**. Notons que *GenSpec* permet l'établissement de liens en fonction du type des exigences impliquées dans la relation quel qu'en soit le niveau hiérarchique de l'exigence.
- fournit un mécanisme permettant de **vérifier un ensemble de propriétés sur les exigences** (*Exemple* : Si l'exigence contient un énoncé, si l'exigence ne contient qu'un seul paragraphe, etc.). Il s'agit plus essentiellement d'un mécanisme de détection que de correction.
- permet de **générer** de manière automatisée différents **documents d'exigences en LN** (format *Microsoft Word*) pouvant répondre à différents standards de canevas d'exigence (IEEE 830, ISO/IEC 12207).
- permet l'**utilisation et la définition de canevas d'exigences**, ceux-ci pouvant répondre aux besoins précis de l'organisation.
- n'est **pas associé à un modèle de développement particulier** (*Exemple* : cascade, itératif, etc.).

**Points faibles** *GenSpec* :

- ne permet la rédaction des exigences qu'en LN **amenant de l'ambiguïté ainsi qu'une non-uniformisation des termes employés** par l'utilisation de synonymes dans les exigences. L'ambiguïté et ce manque d'uniformisation entraînent des interprétations différentes d'une même exigence.
- ne possède **pas de dictionnaire de données ni de concepts** permettant de définir les termes employés dans les exigences.
- ne permet pas, par la **contrainte d'exigence racine unique pour la hiérarchie d'exigences**, de reprendre l'ensemble des exigences dans le cas d'un projet de taille importante. Plusieurs projets établis par *GenSpec* sont alors nécessaires pour reprendre l'ensemble des exigences. De nombreux problèmes de traçabilité apparaissent alors dus aux liens non automatisés entre les exigences contenues dans différents projets établis par *GenSpec*.
- ne permet pas de **filtrer les exigences** selon certains critères et de fournir **différentes vues** selon le type d'utilisateur.
- ne possède **aucune interface avec d'autres logiciels** supportant les autres phases de développement.

Notons qu'à ces points faibles s'ajoute le besoin d'un **support pour la conception architecturale** d'un projet, besoin soulevé par les usagers de *GenSpec* à HQ.

**Tableaux récapitulatifs** Les tableaux 2.8 et 2.9 récapitulent respectivement l'évaluation de *GenSpec* par le cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles. Notons que les exigences non-fonctionnelles *NF3.1.1* à *NF3.2.3* ne sont pas reprises dans ces tableaux car leur évaluation ne peut se réduire à "Oui", "Non" ou "Partiellement".

TAB. 2.8: Tableau récapitulatif relatif à l'évaluation par les exigences fonctionnelles.

Caractéristiques	Activités	<i>GenSpec</i>		
		Oui	Partiellement	Non
<b>FEF 1.1.</b>	<b>FEF 1.1.1.</b>	X		
	<b>FEF 1.1.2.</b>	X		
	<b>FEF 1.1.3.</b>		X	
<b>FEF 1.2.</b>	<b>FEF 1.2.1.</b>			X
	<b>FEF 1.2.2.</b>		X	
<b>FEF 1.3.</b>	<b>FEF 1.3.1.</b>			X
	<b>FEF 1.3.2.</b>		X	
<b>FEF 1.4.</b>	<b>FEF 1.4.1.</b>			X
	<b>FEF 1.4.2.</b>	X		
	<b>FEF 1.4.3.</b>	X		
	<b>FEF 1.4.4.</b>	X		
	<b>FEF 1.4.5.</b>	X		
<b>FEF 1.5.</b>	<b>FEF 1.5.1.</b>		X	
	<b>FEF 1.5.2.</b>			X
<b>FEF 2.1.</b>	<b>FEF 2.1.1.</b>	X		
	<b>FEF 2.1.2.</b>		X	
	<b>FEF 2.1.3.</b>		X	
	<b>FEF 2.1.4.</b>		X	
	<b>FEF 2.1.5.</b>		X	
<b>FEF 2.2.</b>	<b>FEF 2.2.1.</b>		X	
	<b>FEF 2.2.2.</b>			X
	<b>FEF 2.2.3.</b>			X
<b>FEF 2.3.</b>	<b>FEF 2.3.1.</b>			X
	<b>FEF 2.3.2.</b>		X	
	<b>FEF 2.3.3.</b>			X
<b>FEF 2.4.</b>	<b>FEF 2.4.1.</b>			X
	<b>FEF 2.4.2.</b>			X
	<b>FEF 2.4.3.</b>	X		
<b>FEF 3.1.</b>	<b>FEF 3.1.1.</b>	X		
	<b>FEF 3.1.2.</b>		X	
	<b>FEF 3.1.3.</b>			X.
	<b>FEF 3.1.4.</b>		X	
<b>FEF 3.2.</b>	<b>FEF 3.2.1.</b>		X	
	<b>FEF 3.2.2.</b>	X		
	<b>FEF 3.2.3.</b>			X
	<b>FEF 3.2.4.</b>		X	
<b>FEF 3.3.</b>	<b>FEF 3.3.1.</b>	X		
	<b>FEF 3.3.2.</b>	X		
	<b>FEF 3.3.3.</b>			X

TAB. 2.9: Tableau récapitulatif relatif à l'évaluation par les exigences non-fonctionnelles.

Caractéristiques	<i>GenSpec</i>		
	Oui	Partiellement	Non
<b>NF1.1.</b>	X		
<b>NF1.2.</b>			X
<b>NF1.3.</b>			X
<b>NF1.4.</b>			X
<b>NF2.1.</b>		X	
<b>NF2.2.</b>		X	
<b>NF2.3.</b>	X		
<b>NF2.4.</b>	X		

## 2.4 Résumé du chapitre

À l'issue de ce chapitre, nous avons établi une description générale et une évaluation de l'OSIE *GenSpec*.

L'OSIE *GenSpec* est développé à HQ en réponse à la complexité des OSIE commerciaux complexes et difficiles à utiliser. Cet OSIE est basé sur des normes internationales et présente diverses fonctionnalités (rédaction des exigences, caractérisation des exigences, structuration et liaison des exigences, etc.).

*GenSpec* présente cinq types d'utilisateurs, chacun possédant différents privilèges vis-à-vis de l'OSIE correspondant aux tâches du type d'utilisateur dans un projet.

L'évaluation de *GenSpec* a été réalisée par l'application du cadre d'évaluation pour les exigences fonctionnelles et non-fonctionnelles des OSIE de Matulevičius [Mat05]. Cette évaluation met en évidence les avantages et limitations de *GenSpec*.

# Chapitre 3

## Problématique

Dans ce chapitre, nous discutons des problèmes majeurs rencontrés par l'unité Conception-Automatismes à HQ avec l'utilisation de *GenSpec*. L'analyse se limite aux problèmes liés à l'ambiguïté et à la gestion multi-documents d'exigences (GMDE).

Ce chapitre est divisé en 2 sections :

1. **Ambiguïté** : description des conséquences de l'ambiguïté sur les exigences et l'établissement des différentes formes d'ambiguïté (3.1).
2. **GMDE et regroupements d'exigences** : description de la problématique de découpage et d'allocation d'exigences en sous-systèmes (3.2).

### 3.1 Ambiguïté

Dans de cette section, nous allons :

1. fournir une description des problèmes liés à l'ambiguïté (3.1.1) ;
2. définir les différents types d'ambiguïté (3.1.2) ;
3. établir plusieurs origines à l'ambiguïté (3.1.3).

#### 3.1.1 Généralités

Berry et Kamsties [BKK03] déclarent

*“Software requirements specifications (SRSs) (...) [have] many properties, including the need to be precise and accurate, to be self-consistent, and to anticipate all possible contingencies.”*

Les documents d'exigences (*SRSs*) sont généralement écrits en LN, accompagnés d'autres notations tel que des diagrammes UML. Occasionnellement, un document d'exigences est complètement formalisé, utilisant peu de LN excepté pour les commentaires. Pratiquement, tout document conceptuel initial pour un système est réalisé en LN. Une étude menée en 2003 par l'Università di Trento, repris par Berry et Kamsties [BKK03] montre que la majorité des documents (71,8 %) pour l'analyse des exigences sont écrits en LN [BKK03].

Le LN ne nécessite aucune formation ni connaissance d'un quelconque formalisme pour comprendre ou rédiger un document d'exigences. Cependant, l'ambiguïté qui lui est associée est un problème majeur. En IE, l'ambiguïté peut être définie comme introduisant plusieurs interprétations possibles d'une même exigence malgré la connaissance du contexte de l'IE [Cha06].

Notons que nous limitons ici la discussion au LN bien qu'il soit clair que des descriptions semi-formelles ou formelles sont également susceptibles de présenter de l'ambiguïté, bien

qu'en nombre plus limité [BKK03].

L'ambiguïté est le talon d'Achille des documents d'exigences. Elle conduit à des attentes divergentes et des implémentations inadéquates ou indésirables. Un document d'exigences ambigu peut engendrer des implémentations différentes malgré la certitude des programmeurs d'avoir réalisé le comportement attendu [BKK03].

Comme nous l'avons exposé dans le chapitre 2, *GenSpec* permet la rédaction des exigences en LN. Les exigences produites par le biais de cet OSIE souffrent également des problèmes liés à l'ambiguïté.

### 3.1.2 Types d'ambiguïtés

Chantree [Cha06] établit quatre types d'ambiguïtés :

1. **ambiguïté lexicale** : L'ambiguïté lexicale survient lorsqu'un terme possède plusieurs sens. Ce type d'ambiguïté peut être subdivisé en homonymie et polysémie (*cf. sous-section 3.1.3*).  
*Exemple* : “*Canon*” (d'une armée) et “*Canon*” (un chant de messe) sont homonymes [Wik06].
2. **ambiguïté syntaxique** (ou structurelle) : L'ambiguïté structurelle réfère à une séquence de mots pouvant être structurée grammaticalement de différentes manières, amenant généralement plusieurs interprétations.  
*Exemple* : “*Le professeur d'art tibétain*” peut être lu comme “*Le professeur (d'art tibétain)*” ou “*Le professeur d'art qui est tibétain*” [BKK03].
3. **ambiguïté sémantique** : L'ambiguïté sémantique peut être l'ambiguïté lexicale ou peut référer à une combinaison de mots ayant plusieurs sens.  
*Exemple* : “*Personne n'a vu un cochon avec des ailes*” peut être lu de deux manières différentes. Premièrement, on peut le comprendre comme il n'existe pas de cochons avec des ailes ou deuxièmement, il existe un cochon mythique avec des ailes et personne ne l'a jamais vu.
4. **ambiguïté pragmatique** : L'ambiguïté pragmatique survient lorsque le contexte d'une expression a de l'influence, fournissant plusieurs alternatives sémantiques de l'expression.  
*Exemple* : “*Les camions devraient traiter les chemins avant qu'ils ne gèlent*”. “ils” peut référer aussi bien aux camions qu'aux chemins.

### 3.1.3 Origines de l'ambiguïté

Les différents types d'ambiguïté posent problème en IE par le processus de désambiguïsation inconsciente ou non reconnue. Le lecteur, totalement inconscient des autres alternatives sémantiques du texte, comprend le premier sens qui lui vient à l'esprit et le considère comme l'unique sens associé à ce texte. Or, le sens que le lecteur a associé à ce texte peut être erroné [BKK03].

Le problème d'ambiguïté peut être amplifié lorsque l'auteur du document d'exigences est indisponible lors de la lecture du document et ce, pour pouvoir lui soumettre des questions. Souvent, l'auteur du document d'exigences travaille sur d'autres projets et devient indisponible pour les programmeurs, laissant alors de nombreuses ambiguïtés en suspens [BKK03].

Toutes ces ambiguïtés ont différentes sources. Ces ambiguïtés sont engendrées par des erreurs communes de reconnaissances de termes techniques par l'auteur ou le lecteur. Il est courant que des idées fausses sur le sens d'un mot ou d'une phrase, contribuent à l'ambiguïté,

spécialement lorsqu'un terme, mot ou phrase possède un sens technique et un sens commun [BKK03].

Nous nous limiterons, dans cette analyse, à donner une description de l'ambiguïté lexicale car la gestion des autres types d'ambiguïté nécessite des mécanismes trop avancés pour *GenSpec*. Pour plus d'informations concernant les autres types d'ambiguïté, veuillez consulter Berry et Kamsties [BKK03] et l'annexe I reprenant un résumé des différents types d'ambiguïtés établis par ces mêmes auteurs.

La compréhension d'un document d'exigences par l'ensemble des parties prenantes d'un projet est essentiel pour son bon déroulement. À HQ, de nombreuses discussions peuvent survenir entre les différents acteurs d'un projet en raison d'une utilisation ou d'une sémantique non uniformisée de termes ou d'une sémantique différente des termes employés.

Ci-dessous, nous passons en revue les deux types d'ambiguïté lexicale : homonymie et polysémie. A cela, s'ajoutera deux autres sources possibles d'ambiguïté : la synonymie et la subjectivité.

**Homonymie et polysémie** L'homonymie apparaît lorsque deux mots possèdent la même représentation écrite et phonétique mais sont de sémantique et d'étymologies différentes, c'est-à-dire de différentes histoire de développement [BKK03]. “*Les homonymes sont des mots dont les différents sens ne sont pas reliés entre eux*” [Dev07]. Par exemple, le mot *avocat* renvoie à la définition d'un fruit ou d'une profession. Cependant, il n'existe aucun lien sémantique apparent entre ces deux mots [Dev07].

La polysémie survient lorsque deux mots possèdent, tout comme les homonymes, la même représentation écrite et phonétique, mais ceux-ci ont des sens apparentés ainsi qu'une étymologie commune [BKK03, Dev07]. Par exemple, le mot *saumon* renvoie aussi bien à la définition de l'animal vivant qu'à la chair issue de l'animal [Dev07]. Ces deux définitions possèdent des significations différentes mais elles ont un lien sémantique évident.

**Synonymie** “La synonymie est un rapport de proximité sémantique entre des mots d'une même langue. Des termes liés par synonymie sont des synonymes.” [Syn07]

Par exemple, *manger* et *se nourrir* sont synonymes [Syn06].

L'emploi de synonymes est très fréquent dans les exigences rédigées en LN. Il est courant que différents acteurs, voire un même acteur, emploie(nt) des termes distincts afin de référer à une même sémantique. Par exemple, dans un projet chez HQ, les termes “bassins” et “liste des groupes de charge par bassin” sont synonymes. Cette utilisation non uniformisée peut alors entraîner confusions et erreurs lors de la lecture des exigences.

**Subjectivité** Un terme subjectif ou une expression subjective est un terme ou une expression dont la compréhension repose sur l'affectivité personnelle.

Par exemple, *correct*, *efficace*, *rapide*, *complet* sont des termes subjectifs. Ces termes peuvent être interprétés de différentes manières selon l'affectivité du lecteur ou de l'auteur. En effet, la valeur que peut associer un acteur X à l'efficacité par rapport à un acteur Y peut entraîner une interprétation différente d'une exigence en termes d'efficacité.



## 3.2 Gestion multi-documents d'exigences et regroupements d'exigences

Comme nous l'avons développé dans la sous-section 1.1.2, l'IE possède différentes activités. Celles-ci entraînent un découpage du système général en sous-systèmes, l'allocation et la spécification (flowdown) de l'ensemble des exigences à ces sous-systèmes.

### Particularités du processus d'IE dans l'unité Conception-Automatismes à HQ

Le processus d'IE suivi par l'unité Conception-Automatismes à HQ présente certaines particularités :

1. Un premier découpage du système est effectué par l'analyste avant la rédaction des exigences. Le système est décomposé en **regroupements d'exigences**<sup>1</sup> et non en sous-systèmes. Un regroupement d'exigences est un ensemble hiérarchisé d'exigences établi par un analyste.

*Exemple* : Dans un projet réalisé par l'unité Conception-Automatismes, le système est décomposé en plusieurs regroupements d'exigences : SBC (Surveillance des bassins de charges), TC (Télé-délèstage de charges), etc.

2. L'analyste rédige les exigences dans le regroupement d'exigences établi précédemment. Un processus de flowdown est alors réalisé, permettant d'établir une hiérarchie d'exigences. Une exigence de niveau inférieur spécialise une exigence de niveau supérieur. Notons qu'une même exigence peut apparaître dans différents regroupements d'exigences.
3. Un second découpage du système est effectué par le concepteur. Le système est décomposé en une **hiérarchie de composants**<sup>2</sup> (cf. FIG. 3.1), correspondant aux sous-systèmes et en une **liste de fonctions**, correspondant aux comportements que le système devra posséder une fois développé. L'ensemble des exigences établies sont allouées à ces composants et fonctions par un processus d'allocation.

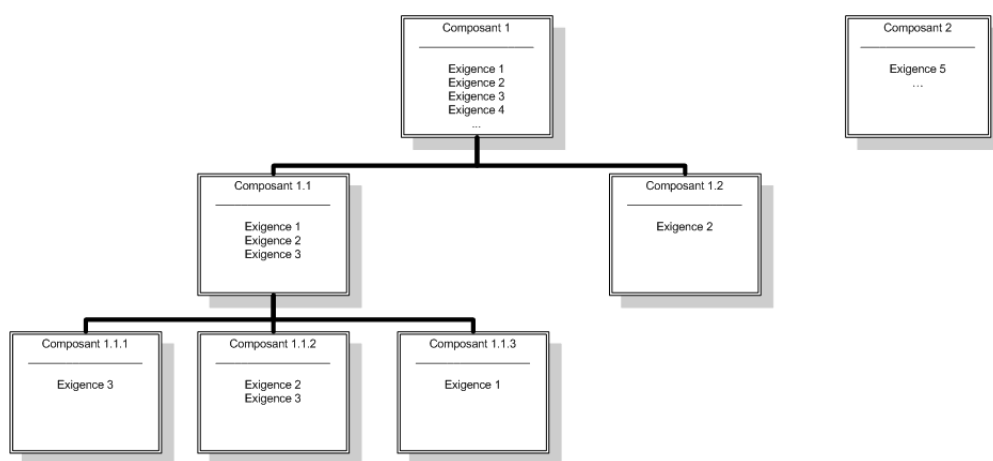


FIG. 3.1 – Hiérarchie de composants établie par le concepteur.

Cependant, l'OSIE *GenSpec* présente certaines limitations par rapport à un tel processus.

<sup>1</sup>Terminologie utilisée par l'unité Conception-Automatismes à HQ.

<sup>2</sup>Terminologie utilisée par l'unité Conception-Automatismes à HQ.

### 3.2.1 Gestion multi-documents d'exigences

Le problème majeur rencontré par *GenSpec* est le manque de support pour un découpage en regroupements d'exigences. Tout nouveau regroupement d'exigences d'un projet constitue un nouveau "projet *GenSpec*", bien que l'ensemble des projets *GenSpec* réfère à un projet en cours de développement. Un projet *GenSpec* correspond à un seul regroupement d'exigences. Un projet *GenSpec* peut convenir à un projet HQ dans le cas où la taille (le nombre d'exigences) de ce dernier est relativement faible.

La figure 3.2 montre que les exigences d'un projet à réaliser peuvent se répartir sur de multiples projets *GenSpec* (un projet *GenSpec* représente un ensemble d'exigences). Certaines exigences identiques peuvent être dupliquées d'un projet *GenSpec* à l'autre (représenté par l'intersection des ensembles "Projet *GenSpec* 1" et "Projet *GenSpec* 2"). Notons qu'une exigence d'un projet ne peut être contenue en dehors d'un projet *GenSpec*.

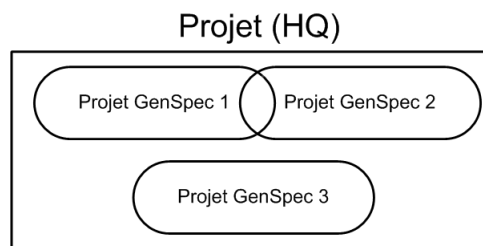


FIG. 3.2 – Relation entre un projet et ses projets *GenSpec*.

Cela entraîne diverses difficultés :

- **Gestion manuelle de la traçabilité** : Tout lien entre exigences dans des projets différents doit être géré manuellement. Une telle gestion est susceptible de provoquer des incohérences lors du développement d'un projet.

Par exemple dans la figure 3.2, une exigence E3 (appartenant au "Projet *GenSpec* 3") dépend de l'existence de l'exigence E1 (appartenant au "Projet *GenSpec* 1"). Lors de la suppression de l'exigence E1 dans le regroupement d'exigences "Projet *GenSpec* 1", l'exigence E3 doit être supprimée du regroupement d'exigences "Projet *GenSpec* 3". Sans gestion automatique, une incohérence peut survenir si l'utilisateur oublie de supprimer l'exigence E3.

- **Dédoublage des exigences** : Une même exigence peut apparaître dans plusieurs regroupements d'exigences et donc dans plusieurs projets *GenSpec*. Une seule occurrence, ou tout au moins une gestion automatique des occurrences d'une exigence dans un projet, permet de limiter la redondance d'informations et de répercuter toute modification.

Prenons par exemple (cf. FIG. 3.2) l'exigence E1 du regroupement d'exigences "Projet *GenSpec* 1". Celle-ci possède  $n$  occurrences dans le regroupement d'exigences "Projet *GenSpec* 2". Toute modification d'une des  $n$  occurrences, doit être répercutée sur les  $n-1$  autres occurrences. Une même exigence dans des états différents dans des regroupements d'exigences différents est problématique.


- **Documents d'exigences distincts** : La problématique liée aux multi-projets *GenSpec* entraîne la génération de multiples documents d'exigences distincts, chaque regroupement d'exigences établissant un document d'exigences indépendant des autres. Encore une fois, nous retrouvons une redondance d'informations entre les différents documents. Sauf par traitement manuel (par exemple à l'aide d'un traitement de texte), il est impossible d'établir un document d'exigences général reprenant l'ensemble des exigences du projet en cours de développement.

### 3.2.2 Regroupements distincts entre l'analyste et le concepteur

Le découpage d'un système en sous-systèmes peut être réalisé de différentes façons selon les aspects jugés importants. À ce titre, on peut affirmer que le découpage d'un système en regroupements d'exigences effectué par l'analyste est arbitraire. L'analyste privilégiant certains aspects à d'autres. Chez HQ, le concepteur réalise un second découpage du système en une hiérarchie de composants pour établir une découpe architecturale. Le découpage effectué par l'analyste ne répondant pas à ces besoins.

Le concepteur établit alors manuellement (via un tableur) un second découpage des exigences en composants (cf. FIG. 3.3). Pour chaque exigence, un ensemble d'information est repris :

- **SES** (*Spécification des exigences du système*) : le nom du projet *GenSpec* dans lequel se trouve l'exigence ;
- **NoWord** : le numéro de l'exigence dans le document d'exigences *Microsoft Word* généré ;
- **Titre** : le titre de l'exigence ;
- **Fonction** : la fonction que l'exigence a dans le projet ;
- **Application** : le nom du composant (utilisé lors la découpe du système) dans lequel se trouve l'exigence.



Conception d'architecture système

SES	NoWord	Titre	Fonction	Application
TDST-TC	3.3.3	Attributs	Conception	Automatisme TDST
TDST-TC	3.4	Autres exigences	Automatisme TDST	Automatisme TDST
CADRE	3.3.1	Performance	Conception	CADRE
CADRE	3.3.1.1	Statiques	Conception	CADRE

FIG. 3.3 – Exemple de tableau d'allocation (*Application*) à des composants et fonctions établis manuellement (HQ).

La hiérarchie de composants reprend l'ensemble des exigences du système. Outre ces composants, le concepteur alloue également les exigences du système aux fonctions.

L'allocation des exigences à une hiérarchie de composants et fonctions est une problématique majeure à HQ. Elle peut entraîner des problèmes dans le développement d'un projet si des modifications apportées aux exigences ne sont pas répercutées manuellement.

### 3.3 Résumé du chapitre

À l'issue de ce chapitre, les problèmes liés à l'ambiguïté et au découpage du système en sous-systèmes ont été passés en revue vis-à-vis de *GenSpec*.

L'**ambiguïté** est un phénomène dont il faut tenir compte en IE. De nombreux documents d'exigences sont rédigés en LN. L'ambiguïté est inhérente au LN. Cependant, l'utilisation du LN reste répandue pour sa facilité d'accès.

L'ambiguïté en IE possède plusieurs origines :

- les fausses idées sur la sémantique d'un terme ;
- les homonymes, polysèmes, synonymes et termes subjectifs ;
- la limitation du lecteur à la première interprétation de sa lecture ;
- l'indisponibilité de l'auteur du document d'exigences à répondre aux interrogations.

Le non support d'une **gestion multi-documents d'exigences** est problématique pour *GenSpec*. Elle engendre :

- une gestion manuelle de la traçabilité ;
- un dédoublement des exigences ;
- la génération de documents d'exigences distincts.

Des **regroupements distincts entre l'analyste et le concepteur** sont établis dans l'unité Conception-Automatismes à HQ. Le découpage en regroupements d'exigences effectué par l'analyste ne convient pas au concepteur pour l'allocation des exigences aux composants. Le concepteur doit alors établir et maintenir manuellement l'ensemble des allocations d'exigences.



## Chapitre 4

# État de l'art d'outils de support à l'ingénierie des exigences

Ce chapitre évalue les fonctionnalités trouvées dans plusieurs OSIE contribuant à résoudre les problèmes identifiés dans le chapitre 3. Il se décompose en deux sections :

1. Une description des fonctionnalités fournies par cinq OSIE commerciaux (4.1) ;
2. Une synthèse des fonctionnalités recensées dans ces OSIE et un résumé des idées provenant de la recherche (4.2).

Il ne s'agit donc pas de réaliser une analyse complète qui couvrirait toutes les fonctionnalités disponibles de chacun des OSIE. L'analyse porte sur les fonctionnalités qui traitent l'ambiguïté et gèrent les regroupements des exigences. Notons que chaque OSIE dispose de sa propre terminologie. Les analyses présentées utilisent la terminologie spécifique aux OSIE. Par exemple, certains OSIE utilisent "*mot*" ou "*concept*" pour référer à ce que nous appelons "*terme*".

### 4.1 Outils

Afin de réaliser l'évaluation, plusieurs sources ont été prises en compte :

- Le site internet <http://www.incose.org> [Inc07] met à disposition des intervenants des informations relatives aux OSIE. Ces informations sont utiles pour les personnes impliquées dans la gestion des exigences d'un projet. Les informations, classées dans une BD et présentées sous forme matricielles, sont fournies directement par les distributeurs de ces OSIE.
- Le site internet des distributeurs présente leurs produits en mettant en évidence leur utilité et leurs points forts.
- Les manuels d'aide proposés par ces OSIE décrivent précisément leurs possibilités ainsi que la mise en oeuvre de ces possibilités.
- L'utilisation des OSIE à des fins d'expérimentation pour tester les fonctionnalités disponibles. Cette utilisation permet de porter un jugement (autre que celui fourni par le vendeur de l'OSIE) sur des critères de qualité tels que la convivialité, l'utilisabilité ou la facilité de compréhension des fonctionnalités.

Les OSIE analysés dans ce chapitre sont les suivants :

1. Telelogic Doors 8.1 (4.1.1) ;
2. IRqA (Integral Requisite Analyzer) 3.5.0 (4.1.2) ;
3. REM (REquirements Management) 1.2.2 (4.1.3) ;
4. IBM Rational Requisite Pro 7.0.0.0 (4.1.4) ;
5. SysML pour Artisan Studio 6.1.21 (4.1.5).

L'analyse présente pour chacun des outils :

- les techniques proposées pour répondre à la problématique de l'ambiguïté ;
- les possibilités d'organiser les exigences en regroupements d'exigences.

Notons que des remarques préliminaires accompagnent certains OSIE afin de mettre en évidence une terminologie ou des caractéristiques spécifiques à l'OSIE.

### 4.1.1 Telelogic Doors 8.1

**Traitement de l'ambiguïté** L'OSIE propose deux types de dictionnaires, chacun se différenciant par son emplacement. Le dictionnaire client réside sur la machine sur laquelle la partie cliente de Doors est installé, tandis que le dictionnaire de la BD est localisé sur la partie serveur. Les deux types de dictionnaire se distinguent par le contrôle d'accès et les privilèges accordés aux utilisateurs. Ces dictionnaires permettent de gérer les mots enregistrés par les utilisateurs de l'OSIE.

La gestion des dictionnaires propose à l'utilisateur les trois types d'actions suivantes :

1. ajouter ou supprimer un terme dans la liste des termes contenus dans le dictionnaire choisi ;
2. définir ou supprimer des mots alternatifs, de sorte que lors de la vérification orthographique, les occurrences des mots alternatifs définis préalablement puissent être remplacés par le terme qui leur correspond ;
3. modifier les droits d'accès selon les privilèges accordés à l'utilisateur s'il s'agit du dictionnaire de la base de données.

La figure 4.1 présente la fenêtre relative aux dictionnaires. Le dictionnaire client y est représenté, notamment avec les alternatives de quelques mots appartenant à ce dictionnaire.

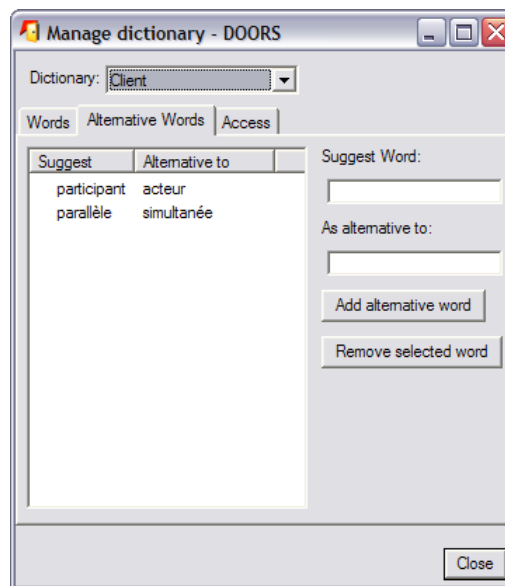


FIG. 4.1 – Liste des mots alternatifs.

La vérification orthographique utilise les dictionnaires. Lorsqu'un mot alternatif est rencontré, le vérificateur orthographique propose de le remplacer par le mot qui lui est associé (c'est ce dernier qu'il est recommandé d'utiliser dans les exigences). La détection de mots alternatifs se fait lorsque la vérification orthographique est demandée, une fois les exigences définies et enregistrées. Ce scénario est représenté dans la figure 4.2.

L'OSIE ne propose pas de moyens permettant d'associer une définition à un mot. L'unique fonctionnalité relative à la définition des mots alternatifs ne permet que très basiquement et de manière incomplète de répondre au problème de l'ambiguïté lexicale.



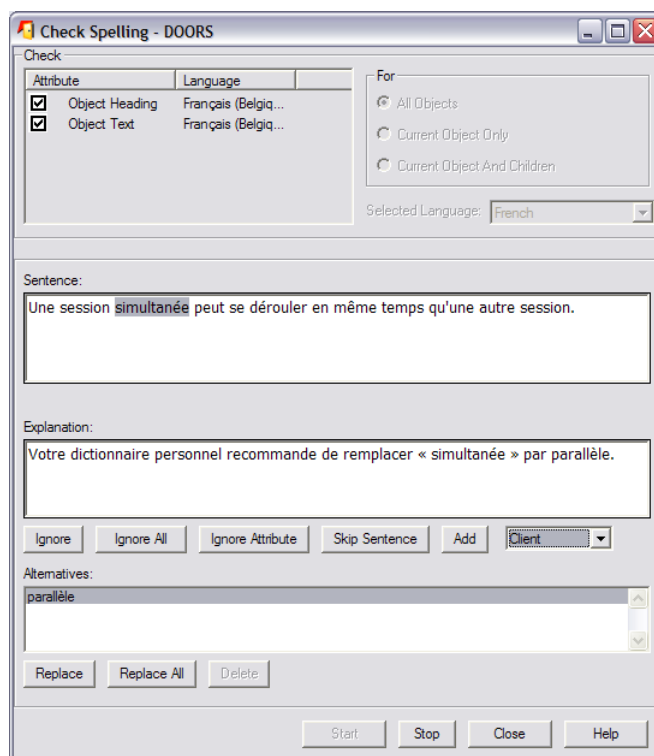


FIG. 4.2 – Proposition de remplacement d'un terme.

**Regroupement des exigences** Toutes les informations d'un projet sont contenues dans des modules. Les modules sont organisés dans une hiérarchie de dossiers permettant de les structurer. Les informations d'un projet peuvent donc être réparties entre plusieurs modules en fonction par exemple du type d'exigences que contient chacun de ces modules. La figure 4.3 montre un exemple de classification. Le projet (un type particulier de dossier) "*Sports utility vehicles 4x2*" contient (et est contenu dans) une hiérarchie de dossiers. L'un des dossiers de ce projet ("*Requirements*") contient deux modules ("*Functional Requirements*" et "*User requirements*"), chacun structurant une partie des exigences de ce projet.

Prenons l'exemple d'un constructeur de voitures [Tel]. La construction de voitures faisant l'objet d'un projet peut inclure les modules suivants :

1. un module pour les exigences du client décrivant les caractéristiques qu'il souhaite pour la voiture ;
2. un module de design décrivant les exigences fonctionnelles que la voiture doit satisfaire ;
3. un module de test décrivant les tests qui doivent être réalisés sur la voiture.

Concernant la traçabilité, des liens entre exigences appartenant à des modules différents peuvent être créés. Nous pouvons lier une exigence du client avec une caractéristique de conception qui la satisfait, ou encore un lien entre les caractéristiques de conception et les tests réalisés en vue d'une vérification [Tel]. Notons que l'OSIE fournit un type particulier de module, appelé "*Module de liens*" qui rassemble tous les liens existants entre les exigences. Ces liens peuvent être visualisés sous un format graphique et peuvent être établis entre modules appartenant à des projets distincts.

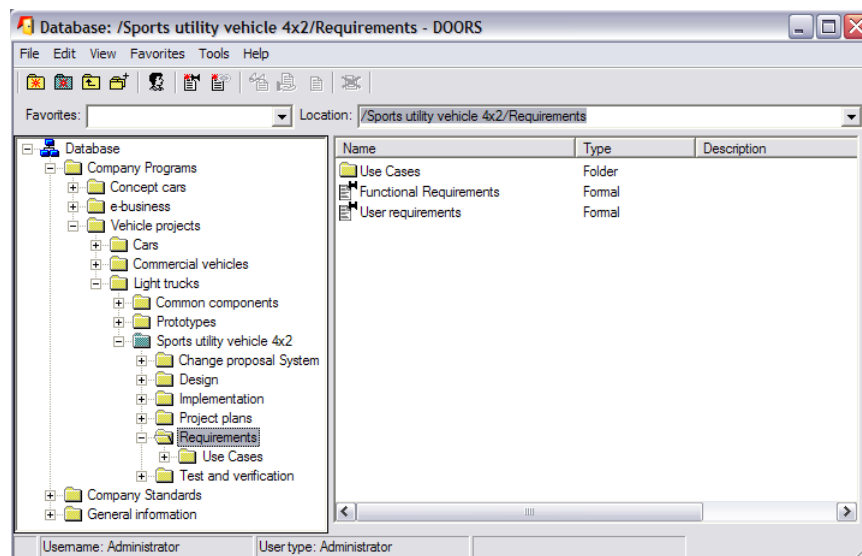


FIG. 4.3 – Organisation des dossiers, projets et modules [Tel].

### 4.1.2 IRqA 3.5.0

**Traitement de l’ambiguïté** Chaque élément du projet possède un ensemble de propriétés. Ces éléments peuvent être une exigence (décrit le besoin d’un utilisateur final), un service (ce que le système doit faire pour satisfaire une exigence), un cas d’utilisation, un acteur etc. Il existe un type particulier d’éléments permettant de définir les termes nommés “*concepts*”, représentant une entité pertinente du domaine d’application, englobant sa structure et son comportement [IRq].

La liste ci-dessous dresse un portrait des différentes informations relatives aux “*concepts*”. Notons que toutes ces informations ne concernent pas directement le traitement de l’ambiguïté. Elles sont néanmoins décrites afin de montrer les possibilités qu’offre l’OSIE en rapport avec sa gestion des “*concepts*”. Nous y retrouvons :

- **La description** du concept correspondant à sa définition dans son domaine d’application.
- **Les détails** décrivant la date de création du concept, l’auteur du concept et la partition d’accès. Les informations de l’auteur sont le prénom, le nom, le numéro de téléphone, l’adresse e-mail, le login et son groupe. Les partitions d’accès sont des parties de la spécification qui se distinguent par les privilèges qu’elles accordent aux utilisateurs. Chaque élément d’un projet IRqA (exigence, service, attribut, rapport, etc.) n’appartient qu’à une seule partition d’accès [IRq].
- **Les éléments associés** au concept. L’OSIE propose à l’utilisateur de définir une liste d’éléments qui sont en relation avec le concept. Ces éléments peuvent être par exemple des services ou des exigences.
- **Les facettes** en relation avec le concept. Les facettes sont différents points de vues permettant d’organiser les spécifications. A chaque facette correspondent différentes valeurs. Ces facettes peuvent représenter par exemple le type d’accès, la priorité, le type de documents, les valeurs de ces facettes pourraient être respectivement un type d’accès confidentiel, une priorité essentielle et le type de documents “note interne au projet”.
- **Les attributs et les opérations.** Il s’agit entre autres des relations existantes entre les différents concepts, ou des relations entre ces concepts et les cas d’utilisation.
- **La liste des synonymes** du concept.
- **Les diagrammes** ayant un rapport avec le terme. Ces diagrammes peuvent être des cas d’utilisation, des diagrammes de classes ou des diagrammes de contexte.

Une fonctionnalité permet de mettre en relation automatiquement un concept avec un élément du système tel qu’une exigence. L’OSIE fournit un “analyseur automatique” qui recherche, une fois les exigences définies, sur base du nom et de la description des exigences les occurrences des concepts ou des synonymes des concepts définis dans ces exigences. A titre d’exemple, la figure 4.4 a détecté une occurrence du terme “*Meeting minutes*” dans le nom de l’exigence “*RQ015SW*”. L’utilisateur peut ensuite accepter la liaison automatique entre le concept et l’exigence dans laquelle ce dernier apparaît. Il est possible de consulter les propriétés de l’exigence avant d’établir une éventuelle liaison.

L’OSIE propose de représenter ces concepts dans des diagrammes, présentant graphiquement les concepts et leurs relations. La figure 4.5 illustre les différents concepts définis (*Document, Request, Person, Department*) ainsi que les relations entre ces concepts. Ce type de diagramme s’avère utile pour représenter la structure des données de haut niveau d’un système [IRq].

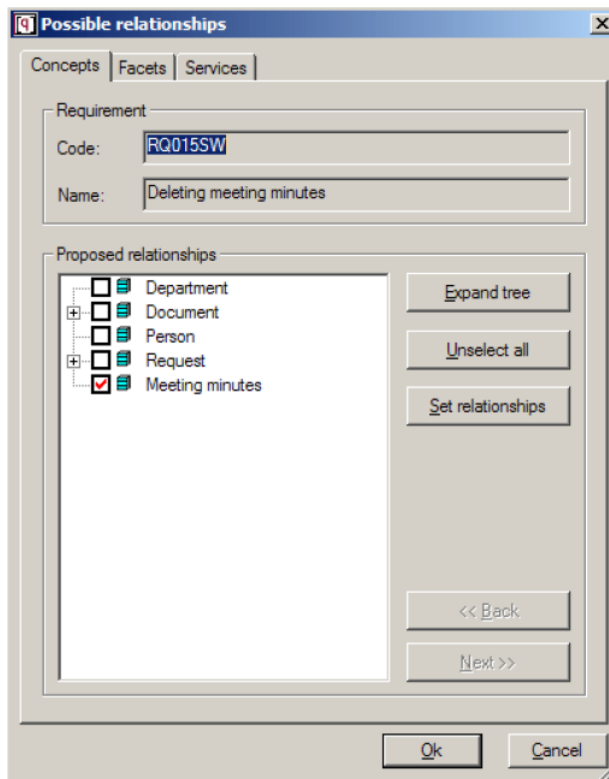


FIG. 4.4 – Proposition de relations [IRq].

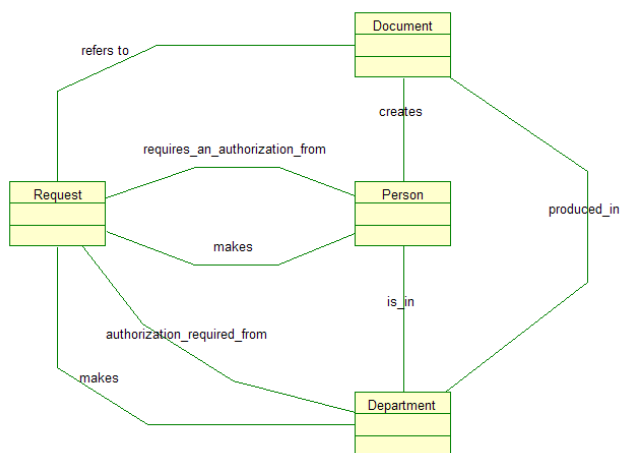


FIG. 4.5 – Diagramme des concepts [IRq].

**Regroupement des exigences** L'OSIE permet de présenter les exigences du projet sous plusieurs formats, rendant l'organisation de ces dernières très flexible. Ces formats classent les exigences :

- en hiérarchie ;
- en liste permettant d'établir des relations autres que hiérarchiques entre les exigences (par exemple des relations définies par l'utilisateur) ;
- en facettes représentant différents points de vues d'organisation des exigences.

### 4.1.3 REM 1.2.2

**Remarques préliminaires** REM (REquirements Management) est un OSIE expérimental de gestion des exigences développé par l'université de Séville, Espagne. La représentation des exigences est basée sur XML. L'usage du langage XSLT permet non seulement la génération automatique de documents, mais permet également la vérification des propriétés de qualité désirées, et le calcul automatique des quelques métriques [DBRT01].

Un projet REM est composé de trois documents [GM03] :

- un document présentant les exigences du client en LN (C-requirements) ;
- un document présentant les exigences du développeur (D-requirements) ;
- un registre pour détecter les conflits et un support de négociation.

La caractéristique majeure du programme est de présenter le résultat des documents à tout moment, au fur et à mesure des modifications apportées. La figure 4.6 fournit un exemple de document d'exigences. La partie gauche de la fenêtre présente les éléments à intégrer dans ce document, à savoir, dans l'exemple, deux organisations, une exigence fonctionnelle et une partie prenante. La fenêtre de droite présente le résultat du document d'exigences et, sous un format tabulaire, toutes les informations disponibles pour chaque objet défini dans la fenêtre de gauche.

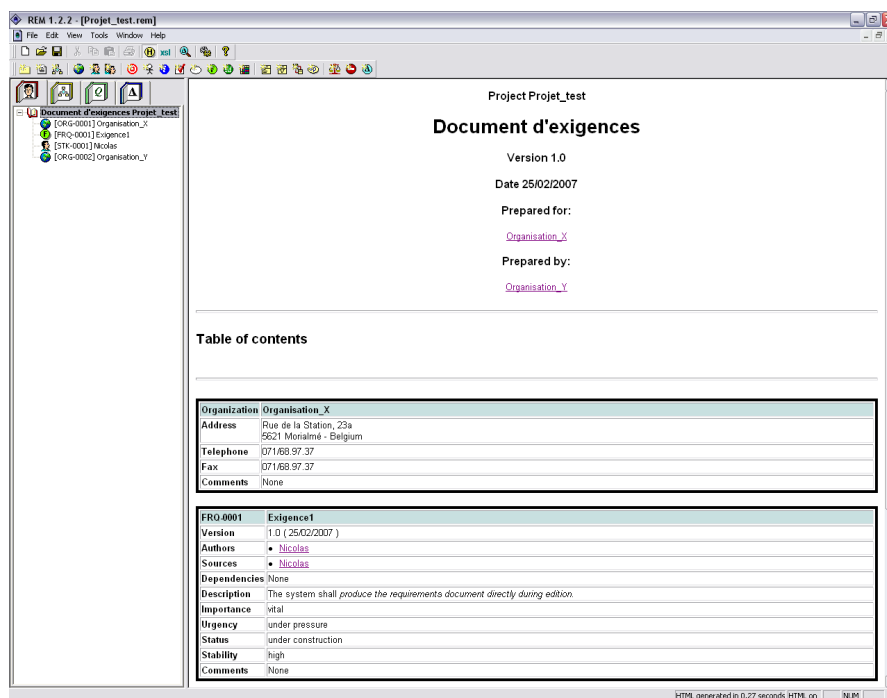


FIG. 4.6 – Document d'exigences dans REM.

**Traitement de l'ambiguïté** L'OSIE ne fournit pas de fonctionnalités de traitement de l'ambiguïté telles qu'un glossaire pour décrivant la terminologie utilisée dans les exigences.

**Regroupement des exigences** L'OSIE propose de définir des hiérarchies de dossiers présentant les différents éléments du projet. Chaque hiérarchie se rapporte à un document qui présente, de manière structurée, les éléments présents dans cette même hiérarchie. Des liens peuvent être établis entre les différents éléments du système contenus dans ces regroupements. Vis-à-vis de la problématique de la distinction entre les regroupements de l'analyste et du concepteur, aucune fonctionnalité n'est disponible.

#### 4.1.4 IBM Rational RequisitePro 7.0.0.0

**Traitement de l'ambiguïté** Cet OSIE présente un type particulier d'exigence associé aux termes. L'OSIE fournit la possibilité de définir des termes dans un glossaire simplifié et un niveau d'ambiguïté qui peut être associé à un terme afin de déterminer si son intitulé ou sa définition risque d'amener de l'ambiguïté.

L'OSIE propose de visualiser les termes définis sous forme de fichier *Microsoft Word*. Ce document sert de glossaire regroupant les termes et leur définition respective.

Une fonctionnalité intéressante est la possibilité d'établir une discussion au sujet de n'importe quel objet d'un projet, y compris pour les termes définis. Les différents acteurs peuvent apposer des commentaires en rapport avec un objet. Par exemple, pour qu'une discussion puisse avoir lieu afin de s'accorder sur une définition non ambiguë d'un terme.

Les figures suivantes présentent les fonctionnalités décrites ci-dessus. La figure 4.7 illustre un exemple de discussion en rapport avec un terme défini dans le glossaire du projet.

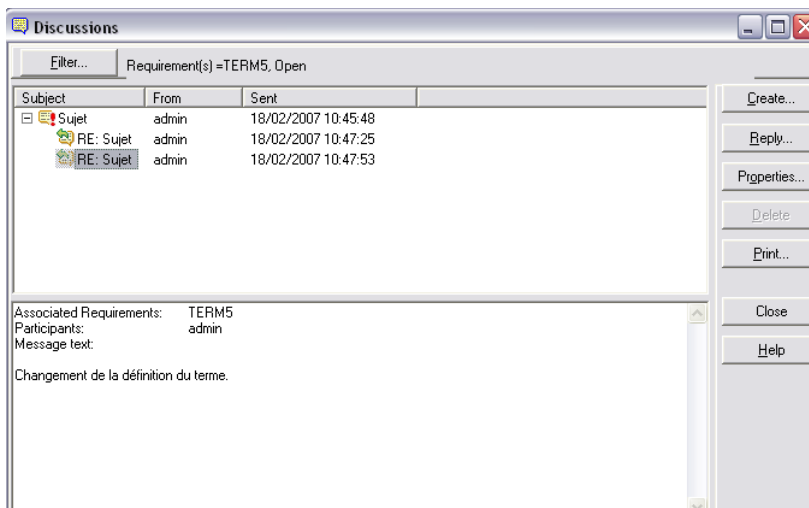


FIG. 4.7 – Discussions relatives à un terme dans RequisitePro.

La figure 4.8 présente les termes ainsi que leurs caractéristiques sous un format tabulaire. Parmi les caractéristiques, nous retrouvons entre autres le niveau d'ambiguïté du terme, son numéro identifiant, sa date de modification, etc. La définition du terme est consultable dans la partie inférieure de la fenêtre.

**Regroupement des exigences** L'OSIE propose d'assembler des éléments (de type exigences ou définitions) dans des "packages". Ces éléments peuvent être déplacés d'un package à un autre en fonction des besoins. Notons en particulier que les termes du glossaire appartiennent à un package prédéfini nommé "glossary".

Les exigences sont organisées en fonctions des besoins des utilisateurs par l'intermédiaire de type d'exigences. L'utilisateur peut ainsi définir des exigences système, des exigences de marketing, des exigences de tests, etc. Chacun de ces types d'exigences possède son propre ensemble d'attributs. Ces attributs peuvent être par exemple, la priorité, le coût, le risque, etc. [Inc07]

The screenshot shows the Rational RequisitePro interface with a table of requirements. The table has the following columns: Requirements, Ambiguity, Unique ID, Location, Package, Author, Revision, Date, Reason, Traced from, Traced to, and Root Tag#. The data rows are as follows:

Requirements	Ambiguity	Unique ID	Location	Package	Author	Revision	Date	Reason	Traced from	Traced to	Root Tag#
TERM1: Administrator	Low	244	Glossary	Glossary	admin	1.0005	07/11/2002 16:46	editorial review			1
TERM2: Shopper	High	245	Glossary	Glossary	admin	1.0008	18/02/2007 10:44	Created trace relationship		TERM5(s)	2
TERM3: Catalog	Medium	246	Glossary	Glossary	admin	1.0004	15/10/2002 20:14	AMBIGUITY: <no entry>			3
TERM4: Customer ID	Low	247	Glossary	Glossary	admin	1.0006	18/02/2007 10:44	Created trace relationship		TERM5(s)	4
TERMS: ADMIN1	Low	351	Database	Glossary	admin	1.0003	18/02/2007 10:53	AMBIGUITY: High - Low	TERM2(s)	TERM4(s)	5
* <Click here to create a...	High	empty	Database	None	admin	1.0000	18/02/2007 10:55				pending

At the bottom of the window, the status bar shows "Ready" and "5 requirements".

FIG. 4.8 – Exemple de caractéristiques des termes.



### 4.1.5 SysML pour Artisan Studio 6.1.21

**Remarques préliminaires** Artisan Studio offre un ensemble d’extensions. “*SysML pour Artisan Studio*” est une extension de l’OSIE permettant de modéliser les besoins selon le formalisme SysML décrit ci-dessous. Cette extension offre de nouvelles fonctionnalités à l’utilisateur de l’OSIE.

La présentation ci-dessous relative à SysML provient d’un article de l’AFIS (L’association Française d’Ingénierie Système) [sit06] :

*“L’essor d’UML dans le domaine du logiciel et l’effort industriel de développement d’outils qui l’accompagne ont conduit à envisager son utilisation en ingénierie système<sup>1</sup>. Cependant, du fait de sa conception fortement guidée par les besoins du passage à la programmation par objets, le langage était, tout au moins dans ses premières versions, peu adapté à la modélisation des systèmes complexes et donc au support de l’ingénierie système.”*

*“D’où l’effort de normalisation en cours à l’OMG d’un langage de modélisation de systèmes, SysML (System Modeling Language). Formalisé sous forme d’un profil UML 2.0, il cherche à adapter au mieux les apports conceptuels et de formalisme de la pensée UML avec les besoins de l’ingénierie système : formalisation des exigences, décomposition, vision fonctionnelle, expression du continu, vérification et validation.”*

**Traitement de l’ambiguïté** Nous ne retrouvons pas de glossaire qui permette de donner une définition à un terme ou une expression. Par contre, l’OSIE propose un dictionnaire qui sert à référencer l’ensemble des objets existants dans un projet en présentant chacun des types d’objets différents sous forme de dossiers. Les caractéristiques de chacun de ces objets sont consultables à partir de ce dictionnaire. Les objets peuvent être de plusieurs types, notamment, acteurs, dépendances, cas d’utilisation, exigences etc.

La figure 4.9 illustre le dictionnaire des données. Le dossier présent dans la partie gauche de la fenêtre contient les deux cas d’utilisation du projet. La fenêtre de droite présente le diagramme de cas d’utilisation. Les propriétés relatives au cas d’utilisation “*Allow Pedestrian Crossing*” sont affichées notamment sous l’onglet général qui présente le nom et le type de l’objet en cours.

L’OSIE permet de remplacer automatiquement l’occurrence d’un terme par un autre dans les objets d’un projet. La description ci-dessous présente la procédure.

Il est possible de créer un objet à partir d’un mot ou d’un groupe de mots présents dans le champ description d’un objet “exigence”. Cet objet peut être une constante *typeDef*, par exemple une chaîne de caractères. Si cette constante a plusieurs occurrences dans les exigences, une modification de la chaîne de caractères de cette constante entraîne une modification de toutes les exigences utilisant cette constante. Par exemple, supposons, d’une part l’exigence “Le système doit donner accès au projet en lecture à l’usager de type lecteur”, d’autre part la constante “lecteur”. Si la chaîne de caractères “lecteur” est modifiée en “visiteur”, l’exigence sera “Le système doit donner accès au projet en lecture à l’usager de type visiteur”. Notons enfin que l’emploi d’une constante de type *TypeDef* est fortement lié à l’implémentation et n’a peut-être pas été développée dans une optique de traitement de l’ambiguïté.

---

<sup>1</sup>L’Ingénierie Système (ou ingénierie de systèmes) est une démarche méthodologique générale qui englobe l’ensemble des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d’un client tout en satisfaisant l’ensemble des parties prenantes [def04].

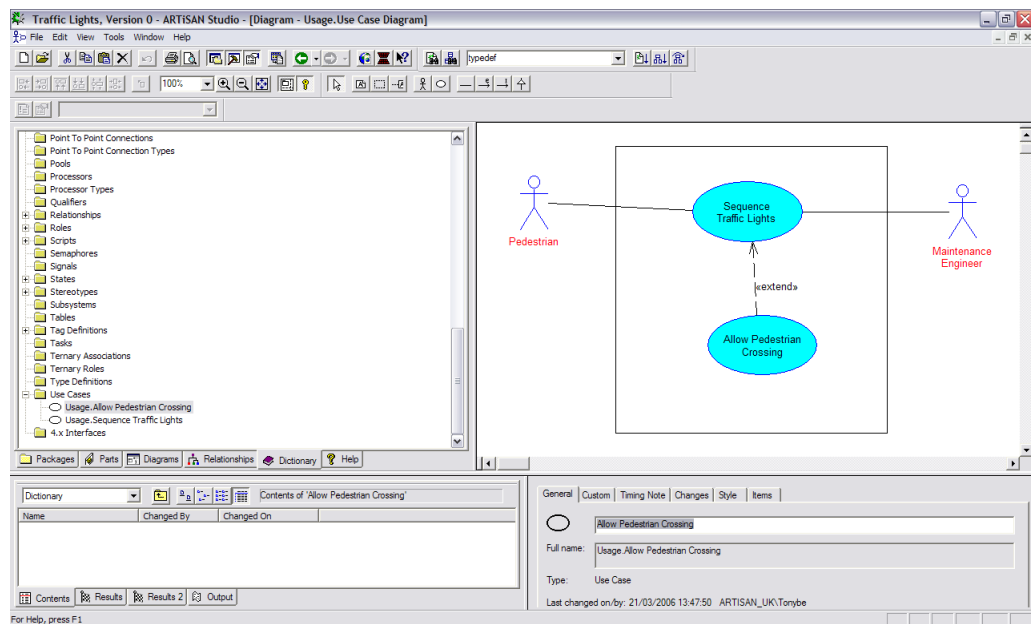


FIG. 4.9 – Dictionnaire d'objets.

**Regroupement des exigences** SysML propose de créer trois types d'objets distincts [Art] :

- **Exigence** : Il est possible de relier les exigences entre elles ou à des diagrammes par l'intermédiaire de différents types de liens.
- **Diagramme d'exigences** : Modélise et montre les relations entre les exigences et présente les liens de traçabilité entre les exigences du diagramme. Ces derniers peuvent représenter des cas d'utilisation, diagrammes de classes, d'activités, d'états etc.
- **Tableau d'exigences** : Il présente les informations de traçabilité dans un tableau *Microsoft Excel*. Les colonnes de ce tableau sont paramétrables afin de présenter les informations pertinentes pour l'utilisateur.

Ces types d'objets sont structurés selon leur type dans le “*Dictionary*” (présenté dans la partie gauche de la figure 4.9).

## 4.2 Synthèse

Cette section expose nos constats concernant les lacunes existantes et les fonctionnalités intéressantes des OSIE s'intégrant dans le cadre de la problématique décrite au chapitre 3. Nous ne résumons que les éléments significatifs ressortant de l'étude des OSIE. Des propositions de solutions à intégrer dans les OSIE sont mentionnées. La possibilité d'intégrer ces solutions à *GenSpec* est également discutée.

### 4.2.1 Tableau récapitulatif concernant le traitement de l'ambiguïté

Le tableau 4.1 présente une évaluation des OSIE concernant la problématique du traitement de l'ambiguïté. Telelogic Doors permet de lister et de détecter des synonymes dans les exigences afin de proposer un terme ("*mot*") conseillé à utiliser dans les exigences. IRqA permet de définir et détecter les termes ("*concepts*") et ses synonymes dans les exigences. L'outil REM ne propose pas de techniques visant à traiter l'ambiguïté. IBM Rational Requisite Pro permet la définition de termes par un type particulier d'exigence et permet d'établir des discussions relatives à ces termes. SysML pour Artisan Studio permet de remplacer un terme défini dans la totalité des exigences.

Les OSIE possèdent généralement un dictionnaire de données ou un glossaire. Certains disposent d'une fonctionnalité assez avancée (IRqA), mais aucun ne fournit un mécanisme avancé permettant de traiter l'ambiguïté.

TAB. 4.1: Tableau récapitulatif relatif à la problématique du traitement de l'ambiguïté.

OSIE	<i>Traitement de l'ambiguïté</i>		
	Oui	Partiellement	Non
Telelogic Doors 8.1		X	
IRqA 3.5.0		X	
REM 1.2.2			X
IBM Rational Requisite Pro 7.0.0.0		X	
SysML pour Artisan Studio 6.1.21		X	

Après avoir analysé les OSIE en matière de traitement de l'ambiguïté, nous en proposons également une synthèse issue de la recherche. Nous retiendrons le guide de Berry, Kamsties et Krieger qui propose des techniques pour améliorer l'écriture des exigences [BKK03] et la thèse de Chantree [Cha06] qui propose une méthode avancée permettant de détecter les ambiguïtés à l'aide d'un corpus.

**Brève description de la thèse de Chantree** La thèse de Chantree [Cha06] propose une méthode d'identification des ambiguïtés susceptibles de provoquer des malentendus, et par conséquent, des implémentations inexactes. Utiliser cette méthode pendant l'analyse des exigences semble être une solution rentable dans la mesure où, comme nous l'avons vu, résoudre les problèmes tardivement dans le processus de développement peut s'avérer très coûteux. Cependant, localiser les erreurs n'est pas une tâche aisée, et même les exigences vérifiées à plusieurs reprises peuvent toujours présenter des défauts. Enfin, Chantree offre une technique visant à assister les ingénieurs à réaliser ce processus.

Sa méthode consiste à distinguer si les ambiguïtés qui provoquant des malentendus peuvent être distinguées des autres. En effet, Chantree résume sa méthode de la manière suivante [Cha06] :

*"We test firstly whether this can be witnessed in the perceptions that humans have of ambiguities. Secondly we test whether such perceptions can be predicted automatically by the use of heuristics."*

L'implémentation basée sur cette méthode nécessite la construction d'un corpus spécialisé à partir duquel les ambiguïtés sont extraites. La nature des jugements relatifs à ces ambiguïtés est obtenue à partir des opinions d'un groupe de personnes soigneusement sélectionné. Enfin, différentes méthodes ont été utilisées pour différencier les ambiguïtés jugées inoffensives des autres.

Notons que la solution de désambiguïsation apportée à *GenSpec* n'a pas utilisé la méthode de Chantree. Sa méthode reste trop éloignée par rapport à la problématique qui nous concerne dans le sens où elle nécessite des ressources humaines et techniques très spécifiques (experts du LN et utilisation d'un corpus de connaissance). De plus, HQ souhaite que son OSIE *GenSpec* reste simple et utilise des techniques intuitives. C'est pour ces raisons que des techniques, bien que moins étendues que celles de Chantree, ont été appliquées à *GenSpec*.

#### 4.2.2 Tableau récapitulatif concernant le regroupement des exigences

Le tableau 4.2 présente une évaluation des OSIE concernant la problématique du regroupement des exigences. Telelogic Doors organise ses projets grâce à une hiérarchie de dossiers et permet la création de différents modules d'exigences. IRqA propose de créer de multiples vues différentes des exigences (en liste, hiérarchie et en facettes). REM permet de créer une hiérarchie d'objets mais est très lié à la présentation des documents d'exigences résultant de ces objets. IBM Rational Requisite Pro propose de définir des “*packages*” en fonction du type d'exigences et permet de déplacer des exigences d'un package à l'autre. SysML pour Artisan Studio propose de nombreuses manières d'organiser et de représenter les exigences (notamment avec les tableaux et diagrammes d'exigences).

TAB. 4.2: Tableau récapitulatif relatif à la problématique des regroupements d'exigences.

OSIE	Regroupement des exigences		
	Oui	Partiellement	Non
<b>Telelogic Doors 8.1</b>	X		
<b>IRqA 3.5.0</b>	X		
<b>REM 1.2.2</b>		X	
<b>IBM Rational Requisite Pro 7.0.0.0</b>	X		
<b>SysML pour Artisan Studio 6.1.21</b>	X		

Les OSIE commerciaux proposent de nombreuses fonctionnalités permettant la gestion et la visualisation des exigences. L'ensemble des OSIE analysés dans la section précédente classe les exigences ou autres objets dans une hiérarchie de dossiers, principe largement répandu. Ainsi, l'utilisateur d'un OSIE organise selon ses besoins et ses contraintes les exigences en groupes qu'il a préalablement définis.

En ce qui concerne les relations entre les exigences, le principe de **matrice de traçabilité** est souvent présent dans les OSIE. La figure 4.10 présente un exemple de matrice de traçabilité d'un projet *Doors* [Tel]. Une telle matrice donne une représentation des exigences qui facilite la visualisation des relations entre elles. Les OSIE fournissent généralement la possibilité de relier des exigences appartenant à des regroupements différents. Les relations peuvent être mises en évidence par l'intermédiaire d'une représentation graphique, matricielle ou par la présence de liens hypertextes. La matrice de la figure 4.10 met en évidence, par l'intermédiaire d'un carré bleu foncé, qu'il existe un lien entre les deux objets à l'intersection de ce carré.

Dans *GenSpec*, une même exigence peut appartenir à plusieurs regroupements d'exigences. Chaque occurrence de cette exigence consiste en une nouvelle exigence synchronisée (uniquement dans un projet *GenSpec*) liant chaque occurrence entre elles. Les autres OSIE (par exemple *Doors*) permettent de “copier-coller” des exigences mais sans les synchroni-

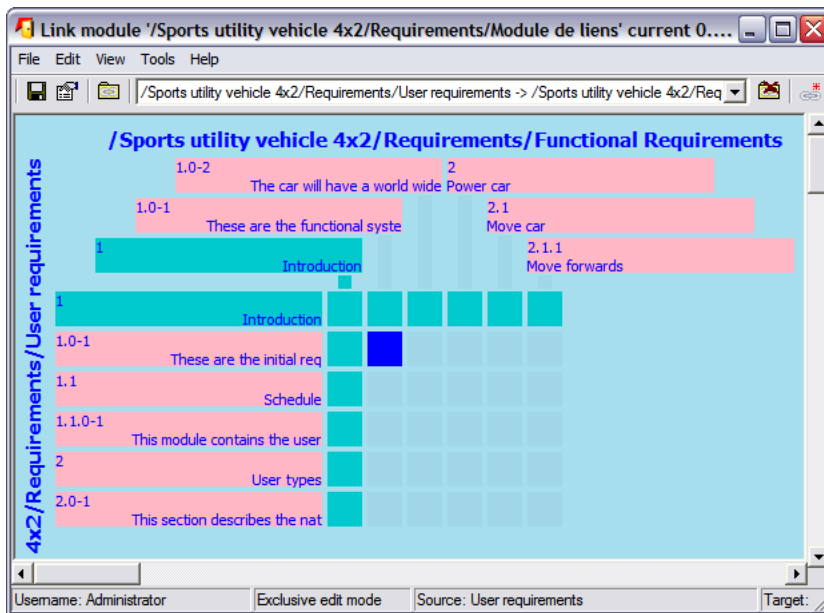


FIG. 4.10 – *Matrice de traçabilité [Tel]*.

ser, en autorisant cependant de créer des liens entre celles-ci. L'accent est alors mis sur les exigences et leurs liaisons.

### 4.3 Résumé du chapitre

Ce chapitre a mis en évidence les moyens fournis par quelques OSIE de support à l'IE concernant les problèmes identifiés dans le chapitre 3.

Concernant la notion d'ambiguïté, nous remarquons qu'une partie seulement de ces OSIE fournit un dictionnaire de données ou de concepts. Les dictionnaires que nous avons trouvés dans les OSIE servent généralement à lister les objets existant dans le projet. Nous retrouvons peu de fonctionnalités relatives à un glossaire qui permettent de définir la terminologie utilisée dans les exigences ou dans le projet. Les OSIE ne proposent pas de mécanismes performants de désambiguïsation. Hormis la détection de synonymes ou de mots alternatifs, aucune fonctionnalité avancée n'a été recensée.

Concernant la gestion des exigences en regroupements d'exigences, tous les OSIE analysés permettent d'organiser les exigences et les objets d'un projet. La possibilité de travailler à partir de différentes vues paramétrables est également disponible.

## Chapitre 5

# Solutions apportées à *GenSpec*

Ce chapitre contient une description de notre solution en réponse à la problématique définie au chapitre 3. Cette solution a été appliquée à l'OSIE *GenSpec*. Ce chapitre comporte :

1. une description générale du processus de réalisation suivi (5.1) ;
2. une description détaillée des activités réalisées (5.2) ;
3. un relevé des problèmes que nous avons rencontrés (5.3).

### Remarques préliminaires :

1. La solution que nous avons développée est subdivisée en deux fonctionnalités :
  - (a) Un **glossaire**, pour le traitement de l'ambiguïté.
  - (b) Une **gestion multi-documents d'exigences (GMDE)**, pour rassembler l'ensemble des exigences et un mécanisme d'**allocation d'exigences à une hiérarchie de composants et une liste de fonctions (AECF)** fournissant un support à l'activité de conception.
2. Différentes parties prenantes ont pris part au développement des nouvelles fonctionnalités. Deux acteurs majeurs sont identifiés dans ce chapitre :
  - le **client 1**, correspondant à *M. René Bujold*, chargé de l'analyse.
  - le **client 2**, correspondant à *M. Alain Sicard*, chargé de la conception.

Le client 1 et le client 2 sont les deux acteurs directement concernés par le développement des nouvelles fonctionnalités. Notons que le client 1 est également le responsable du développement des nouvelles fonctionnalités de *GenSpec*.

3. Dans ce chapitre, toute utilisation des termes “analyste” et “concepteur” renvoie respectivement à la notion générale du rôle d'un analyste et d'un concepteur dans le développement d'un projet.

## 5.1 Description du processus suivi

Un ensemble d'activités (représenté par un diagramme d'activités) a été réalisé pour établir notre solution. La réalisation d'une activité implique généralement un livrable qui sert de base à l'activité suivante dans le processus. Une activité supplémentaire de maintenance des exigences et de la conception ont été réalisées parallèlement aux autres activités. Une description détaillée de l'ensemble de ces activités peut être trouvée dans la section 5.2. Notons aussi que les différentes validations sont réalisées par le client 1.

La figure 5.1 présente les activités et les livrables en résultant. Les deux colonnes (“*swim lanes*”) réfèrent respectivement :

- aux stagiaires réalisant les activités du processus (*Nicolas Pirmez* et *Olivier Pire*);
- aux clients à qui s'adressent les nouvelles fonctionnalités (le client 1 et/ou le client 2).

La succession des activités et des livrables produits sont identiques pour les deux stagiaires. Ils ont réalisé les mêmes activités en parallèle mais sur différentes fonctionnalités :

- *Olivier Pire* a réalisé l'analyse et la conception du glossaire ainsi que l'implémentation de la GMDE et l'AECF;
- *Nicolas Pirmez* a réalisé l'analyse et la conception de la GMDE et l'AECF ainsi que l'implémentation du glossaire.

Avant l'implémentation, un échange des documents d'exigences et de conception a été réalisé entre les stagiaires. L'implémentation d'une fonctionnalité a été réalisée par un stagiaire sur base des documents réalisés par l'autre stagiaire. Par exemple, *Nicolas Pirmez* a implémenté le glossaire sur bases des exigences et de la conception du glossaire réalisée par *Olivier Pire*. Cette démarche a permis d'évaluer la qualité des exigences et de la conception réalisée par chaque stagiaire. De plus, elle a permis d'étendre la connaissance mutuelle de la problématique des deux stagiaires.

Suite à des modifications (notamment concernant les besoins des clients) apportées tout au long du processus de développement, la **revue des exigences et de la conception** a été effectuée tout au long de l'implémentation. Notons que le diagramme d'activités n'explique ces revues que lors d'un début de développement d'un nouvel incrément ou itération.

Le client consulté dépend de la fonctionnalité qui le concerne. Le client 2 était uniquement consulté pour valider les prototypes d'interface graphique de l'AECF et valider les itérations relatives à l'AECF. Toutes les autres validations étaient réalisées par le client 1 (y compris en ce qui concerne les découpages en itérations et incréments du processus de développement).

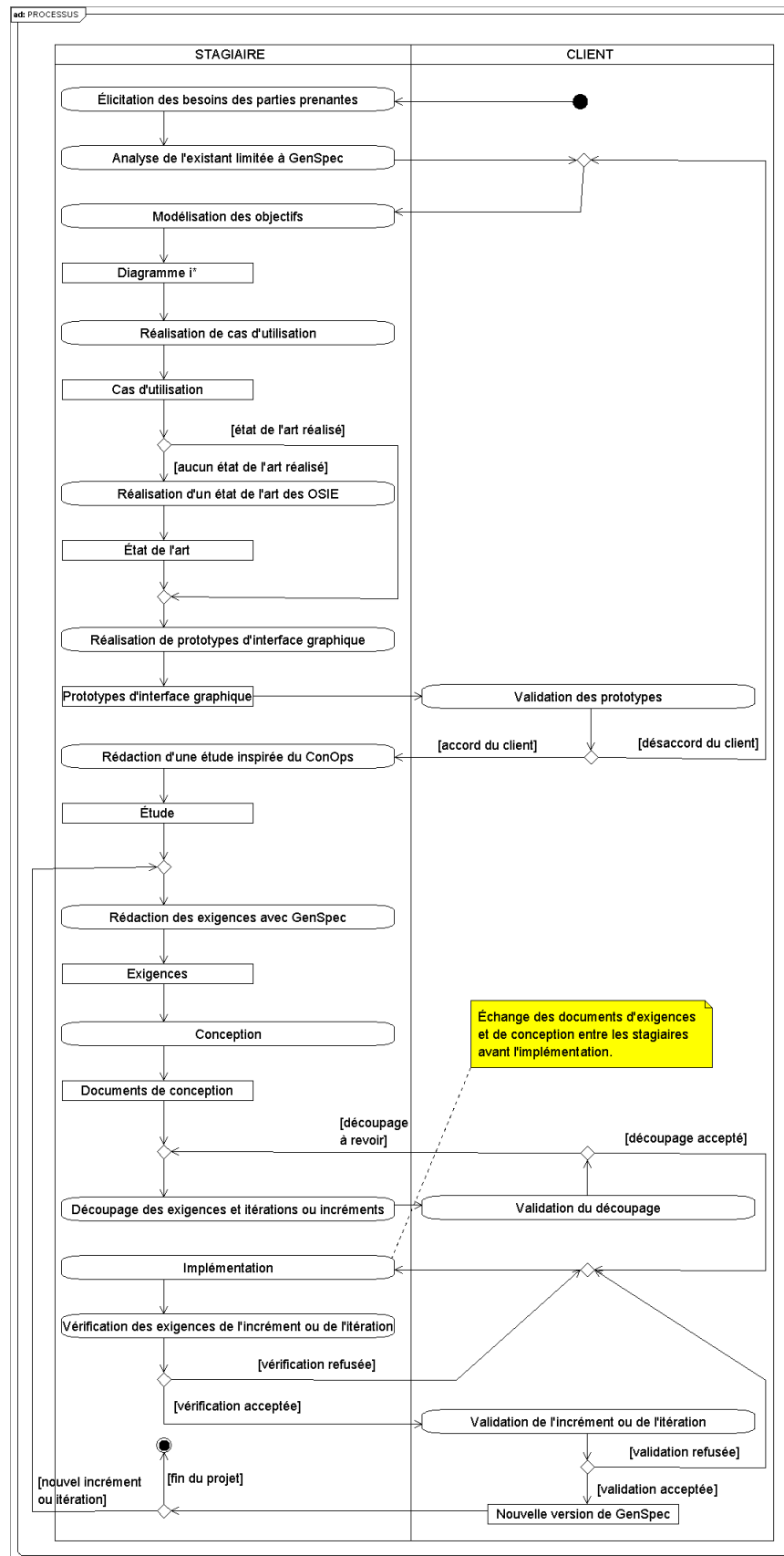


FIG. 5.1 – Processus suivi pour établir la solution.



## 5.2 Description détaillée des activités du processus suivi

Dans cette section, nous allons, pour chaque activité du processus suivi :

1. définir l'**objectif** de cette activité ;
2. expliquer la **procédure** employée, c'est-à-dire comment l'activité a été réalisée ;
3. fournir éventuellement une **description** du contenu développé lors de cette activité.

### 5.2.1 Élicitation des besoins

**Objectif** L'objectif de l'élicitation est de collecter assez d'informations pour identifier le problème et devenir un expert du domaine (cf. sous-section 1.1.2).

**Procédure** L'élicitation des exigences est réalisée en plusieurs étapes successives :

- Réalisation d'**interviews** avec l'ensemble des parties prenantes concernées ;
- **Synthèse et confrontation** des informations élicitées avec les parties prenantes. Plusieurs réunions sont effectuées dans le but de gérer les éventuels conflits relatifs aux besoins des parties prenantes. Les parties prenantes ont une vision différente des besoins et ne s'accordent pas toujours sur la terminologie à utiliser (cf. section 5.3). Un accord a été établi quant à la définition des besoins et la terminologie à utiliser.
- Subdivision des besoins en deux fonctionnalités distinctes :
  - un **glossaire** avancé permettant de réduire l'ambiguïté des exigences ;
  - une **GMDE** et **AECF** permettant, d'une part une gestion plus efficace de l'ensemble des exigences d'un projet, d'autre part un support à l'activité de conception.

L'ensemble de ces activités a été réalisé en parallèle avec la réalisation de prototypes d'interfaces graphiques, diagrammes d'objectifs et cas d'utilisation fournissant un support efficace aux activités d'élicitation.

### 5.2.2 Analyse de l'existant

**Objectif** L'analyse de l'existant a permis de mettre en évidence, d'une part les grands principes de fonctionnement de *GenSpec* et d'autre part, les limitations suscitant un intérêt pour le développement des nouvelles fonctionnalités.

**Procédure** L'analyse de l'existant a été réalisée en deux étapes :

1. la collecte et la lecture de documents et articles concernant *GenSpec* ainsi que l'analyse de documents produits par l'intermédiaire de *GenSpec* pour affiner la compréhension du fonctionnement de l'OSIE ;
2. l'utilisation de *GenSpec* pour la rédaction des exigences des deux nouvelles fonctionnalités à développer.

### 5.2.3 Modélisation des objectifs

**Objectif** Les modèles d'objectifs permettent la définition des objectifs d'un projet. Ils définissent la motivation des choix d'analyse et de conception et les aspects intentionnels des acteurs d'un SI. Ils représentent également les aspects non-fonctionnels (efficacité, sécurité, confidentialité, etc.) du projet [Hey05].

**Procédure** Nous avons représenté ces objectifs par l'intermédiaire de deux diagrammes d'objectifs  $i^*$  correspondant aux deux fonctionnalités (cf. FIG. 5.2 et 5.3).

**Description** Les acteurs considérés sont les types d'utilisateurs *GenSpec* (cf. sous-section 2.1.2). Notons que le concepteur est un nouveau type d'utilisateur. Il réalise la conception architecturale du système à développer. Les deux modèles d'objectifs présentent des caractéristiques propres :

1. **Glossaire** : La figure 5.2 reprend le diagramme d'objectifs de la fonctionnalité de glossaire.

L'*utilisateur - GenSpec* rédige des exigences en LN. Ses objectifs sont :

- établir des exigences non ambiguës et traçables le plus rapidement et de façon la plus complète possible. L'utilisateur utilise les termes définis précédemment ou définit les nouveaux termes employés lors de la rédaction des exigences.
- consulter ou modifier les termes existants dans le glossaire en vue d'améliorer la compréhension des exigences.

L'*administrateur - GenSpec* réalise le document d'exigences d'un projet. L'administrateur établit un glossaire reprenant l'ensemble des informations sur les termes employés. Ses objectifs sont :

- uniformiser les termes utilisés ;
- diminuer les délais dans la rédaction du document d'exigences ;
- augmenter la traçabilité entre les exigences afin de connaître l'ensemble des exigences où le terme est utilisé ;
- diminuer les coûts, en terme d'argent ou d'autres ressources, associés à la rédaction du document d'exigences ;
- améliorer la compréhension du document d'exigences ;
- diminuer l'ambiguïté du document d'exigences.

Notons que l'administrateur possède entre autres les privilèges de l'utilisateur.

Les *autres acteurs - GenSpec* sont le concepteur, le commentateur, le visiteur et le testeur de *GenSpec*<sup>1</sup>. Leur objectif est de consulter l'ensemble des informations du glossaire afin d'améliorer leur compréhension des exigences.

2. **GMDE et AECF** : La figure 5.3 correspond au diagramme d'objectifs de la fonctionnalité de GMDE et AECF.

L'*administrateur - GenSpec* réalise le document d'exigences. Ses objectifs sont :

- réduire les erreurs dans les documents d'exigences ;
- diminuer les coûts ;
- diminuer les délais nécessaires à la rédaction des documents d'exigences ;
- augmenter la traçabilité entre les exigences d'un projet.

L'*utilisateur - GenSpec* rédige les exigences d'un projet. Ses objectifs sont :

- disposer d'un document d'exigences complet ;
- utiliser une même exigence dans des regroupements d'exigences (cf. section 3.2) distincts de manière à éviter la redondance d'exigences et d'améliorer la traçabilité dans un même projet.

---

<sup>1</sup>Ces différents usagers sont représentés par un seul acteur afin d'éviter la redondance.

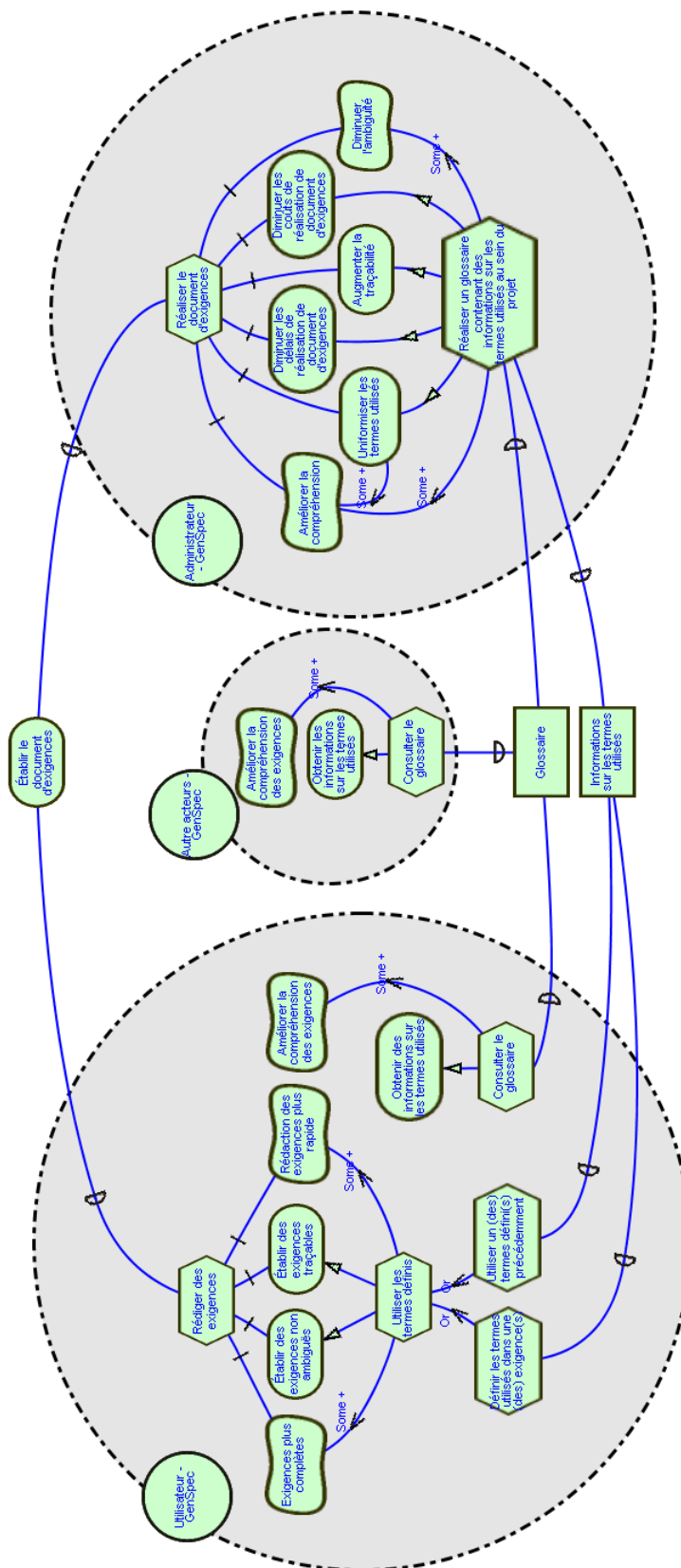


FIG. 5.2 – Diagramme d'objectifs  $i^*$  de la fonctionnalité de glossaire.

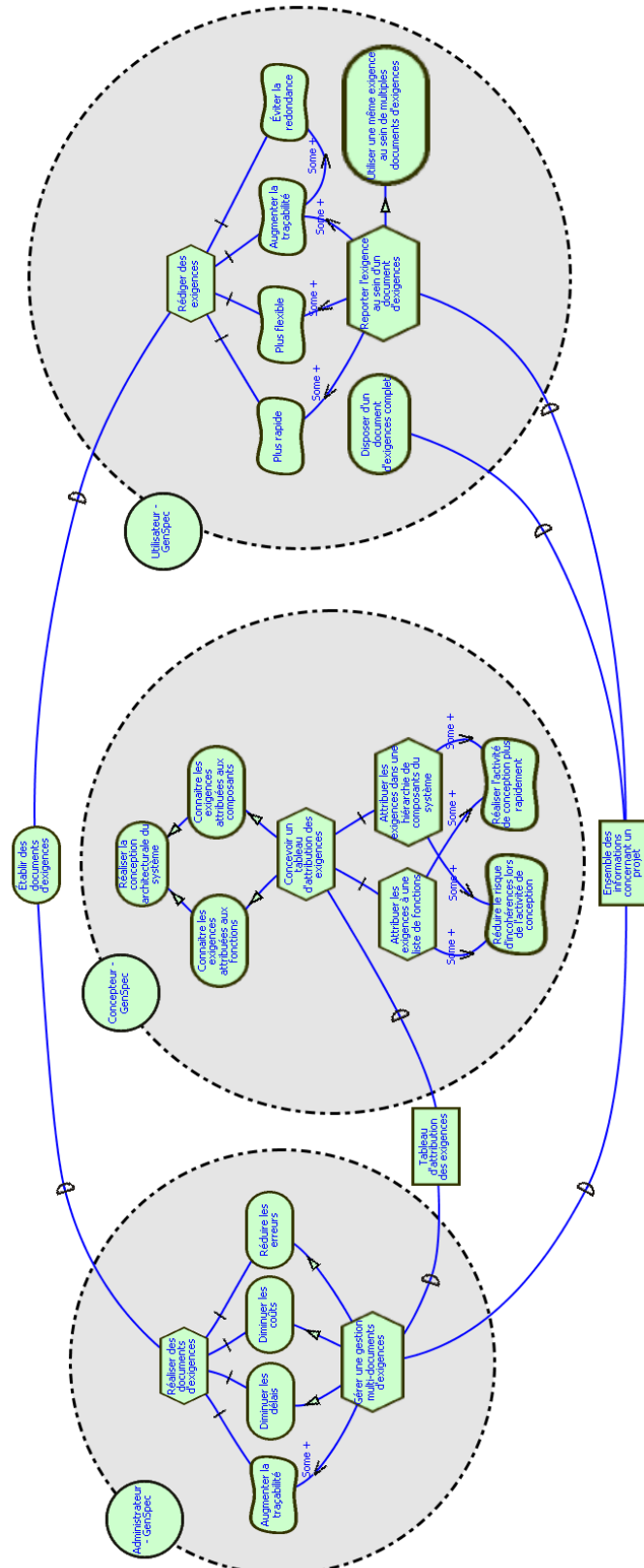


FIG. 5.3 – Diagramme  $i^*$  associé à la fonctionnalité de GMDE et à l'AECF.

Le *concepteur - GenSpec* réalise la conception architecturale du système à développer. Ses objectifs sont :

- réaliser la conception architecturale du système ;
- établir un tableau reprenant les allocations des exigences aux composants et fonctions ;
- réaliser l'AECF plus rapidement ;
- réduire les incohérences possibles lors de la conception ;
- connaître les exigences allouées à un composant (cf. section 3.2) ;
- connaître les exigences allouées à une fonction (cf. section 3.2).

#### 5.2.4 Cas d'utilisation

**Objectif** Les cas d'utilisation sont utilisés pour [Hey05] :

- présenter un type de **diagramme compréhensible** par l'ensemble des parties prenantes (les cas d'utilisation sont rédigés en LN, se basent sur des scénarios et sont concrets) ;
- stimuler la **découverte d'autres types d'informations** telles que la description du domaine d'application ou d'éventuelles exigences, qu'elles soient fonctionnelles ou non ;
- réaliser une première **estimation** de l'ampleur, de la complexité et du coût du système.

**Procédure** Les scénarios de cas d'utilisation importants et pertinents ont été réalisés. Ceux-ci sont repris dans trois *diagrammes de cas d'utilisation*<sup>2</sup> reprenant respectivement les cas d'utilisation associé au glossaire, à la GMDE et à l'AECF. A titre d'exemple, la figure 5.4 reprend le *diagramme de cas d'utilisation* du glossaire. Pour plus d'informations concernant les différents diagrammes et scénarios associés, veuillez consulter l'annexe A pour le glossaire et l'annexe B pour la GMDE et l'AECF.

#### 5.2.5 État de l'art des OSIE

**Objectif** Un état de l'art concernant la problématique définie au chapitre 3 a été réalisé pour s'informer sur l'existence de fonctionnalités analogues au glossaire, à la GMDE et à l'AECF dans le domaine des OSIE.

**Procédure** Deux activités ont permis l'établissement de cet état de l'art :

1. l'analyse de cinq OSIE autres que *GenSpec* ;
2. l'analyse d'articles de recherche se rapportant à cette problématique.

Le chapitre 4 présente en détail l'ensemble de cette démarche.

#### 5.2.6 Prototypes d'interfaces graphiques

**Objectif** Différents prototypes d'interfaces graphiques ont été réalisés comme support aux activités d'élicitation. Ces prototypes permettent aux parties prenantes de valider leurs idées et leurs besoins afin qu'ils soient correctement interprétés [Bra02].

**Procédure** Les prototypes d'interfaces ont été établis selon la compréhension et l'interprétation des besoins et attentes des parties prenantes en matière de solution. Ces prototypes ont été présentés plusieurs fois et ont été modifiés conformément aux remarques des parties prenantes. Le prototype d'interface final correspond globalement à l'interface implémentée. La figure 5.5 présente un prototype d'interface pour l'AECF.

---

<sup>2</sup>UC diagram.

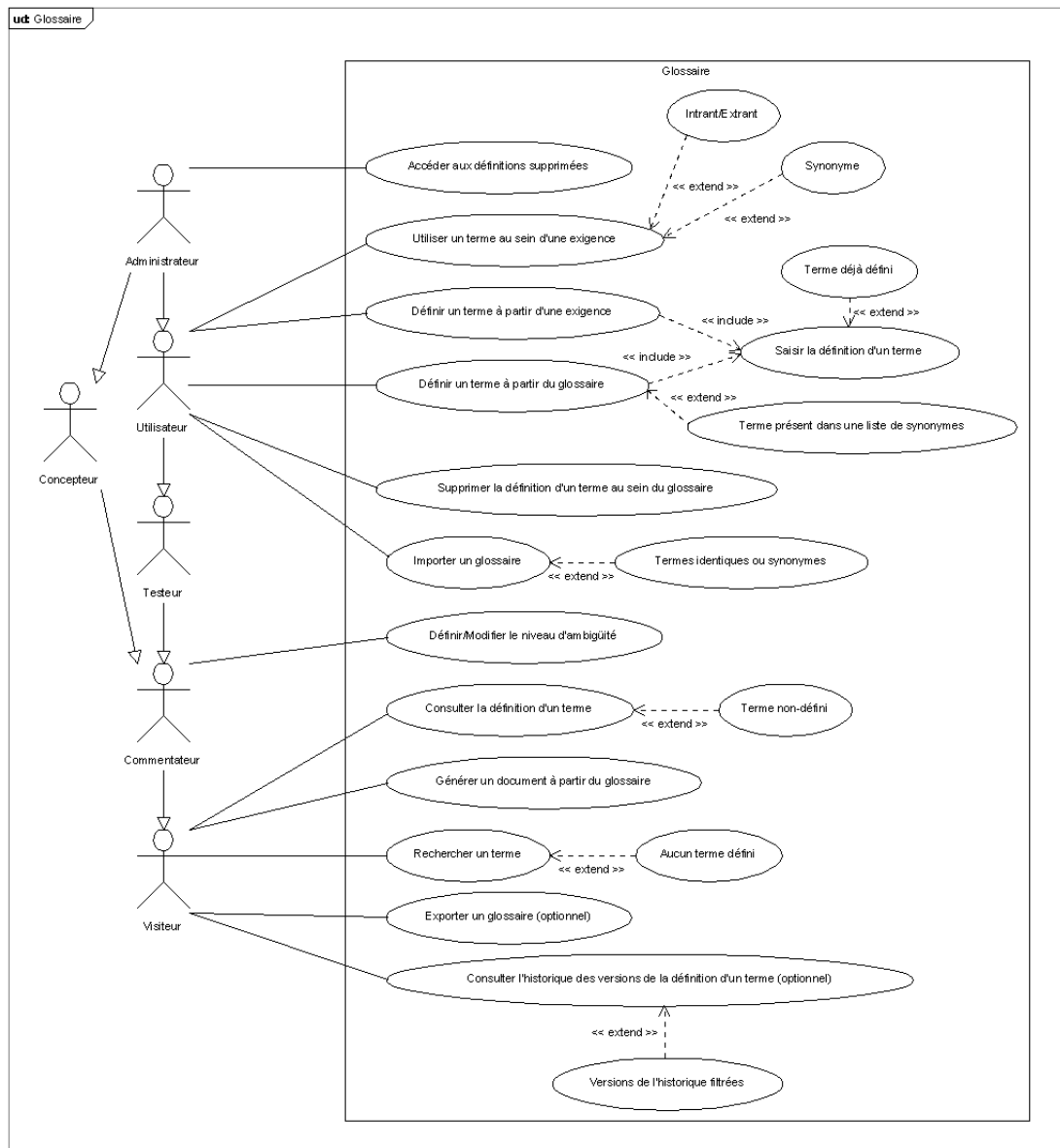


FIG. 5.4 – Diagramme de cas d'utilisation associé au glossaire.

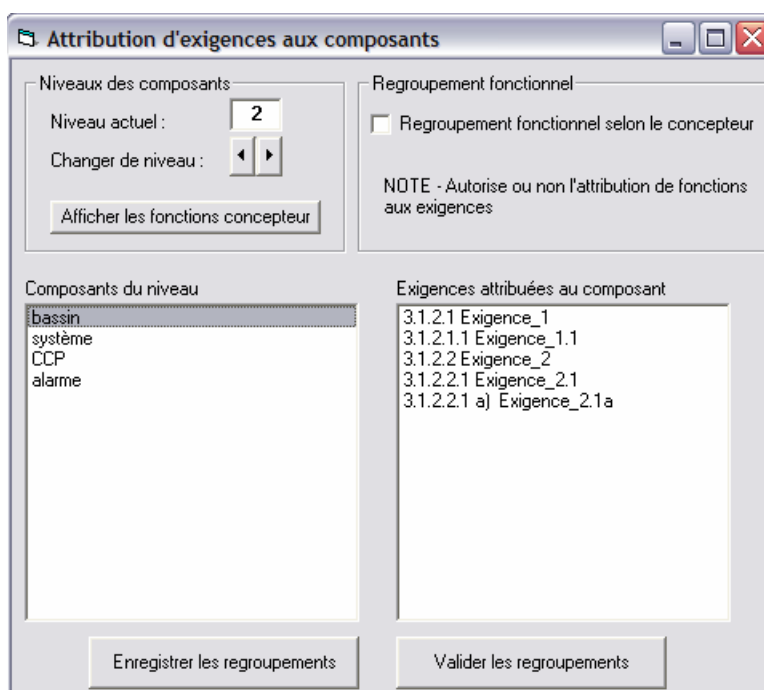


FIG. 5.5 – Prototype d'interface pour AECF (cf. annexe D).

### 5.2.7 Étude

**Objectif** Une étude narrative présentant l'environnement et les fonctionnalités du système [Dor99] est requise avant tout projet de développement à l'unité Conception-Automatismes à HQ. L'étude se base sur le standard IEEE 1362 de documents de conception opérationnel (*ConOps*). Elle fournit une analyse générale des activités et effectue le lien entre les besoins des utilisateurs et leur vision, ainsi que les exigences techniques du développeur [IEE98].

**Procédure** L'étude est un document récapitulatif. Dans notre cas, elle reprend, de manière synthétique, les besoins découverts lors de l'élicitation. L'étude est structurée de la manière suivante :

1. la **situation actuelle** : la situation actuelle décrit le système avant le développement des nouvelles fonctionnalités. Le contexte, l'objectif et l'étendue, les classes d'utilisateurs et leur profil ainsi que la structure organisationnelle correspondant au système y sont décrits.
2. les **solutions proposées** : les solutions proposées décrivent les alternatives de solutions en réponse aux besoins. Les solutions proposées comprennent la justification, la nature des changements et la description des différents éléments de solution présents dans des OSIE concurrents.
3. les **recommandations** : les recommandations décrivent plus en détail la solution la plus adaptée (par présentation du contexte, de l'objectif et de l'étendue du nouveau système).
4. l'**analyse du système proposé** : l'analyse décrit les améliorations (les désavantages) et limites du futur système.

L'annexe C correspond à l'étude du glossaire et l'annexe D correspond à l'étude de la GMDE et l'AECF.

### 5.2.8 Rédaction des exigences

**Objectif** Suivant la méthodologie de l'unité Conception-Automatismes à HQ, nous avons rédigé l'ensemble des exigences de manière détaillée à l'aide de *GenSpec*.

**Procédure** *GenSpec* a été utilisé afin de rédiger en LN les exigences relatives aux fonctionnalités de glossaire et de GMDE. Les annexes E et F présentent respectivement les documents d'exigences du glossaire et celui de la GMDE. Ces documents d'exigences sont composés de l'arbre hiérarchique des exigences et de la spécification des exigences (cette terminologie est spécifique à *GenSpec* [Hyd07b]). Cette démarche a permis de :

1. mettre en évidence les besoins des parties prenantes vis-à-vis de *GenSpec*.
2. se familiariser avec les fonctionnalités existantes de *GenSpec* avant la conception et l'implémentation des nouvelles fonctionnalités.

### 5.2.9 Conception

**Objectif** La conception est une description précise des fonctionnalités à développer moyennant différentes notations techniques : modélisation de la BD, interactions entre les composants, etc.

Ces notations fournissent ensemble une description du système à développer.

**Procédure** Pour chacune des fonctionnalités, différents diagrammes ont été établis.

1. **Schéma entités-associations conceptuel et logique**

**La structure de la BD originale est très redondante, mal structurée et certains éléments restent injustifiés<sup>3</sup>.** La BD n'a pas été modifiée (excepté pour le support du glossaire et de la GMDE et AECF) pour ne pas rendre obsolète l'ensemble des anciens projets *GenSpec*. De plus, une telle ré-ingénierie de l'OSIE dépassait le cadre de notre stage. Par conséquent, **aucun type d'entité (TE) ou attribut n'a été renommé<sup>4</sup> ou supprimé.** La figure 5.6 présente le schéma entités-associations conceptuel sans les données associées au glossaire à la GMDE et l'AECF. Nous distinguons quatre parties dans cette figure :

- les usagers de *GenSpec* (appelés “*Utilisateurs*” dans le schéma) avec leurs connexions et leurs privilèges ;
- les exigences, les groupes d'exigences (notamment le groupe d'exigences à générer) et les essais utilisés dans *GenSpec* ;
- les types d'exigences et les relations entre ces types ;
- diverses informations complémentaires (numéro de version, paramètres, etc.) décrivant notamment les options de génération des documents d'exigences, les variables établies, etc.

Un schéma entité-association conceptuel reprenant les concepts essentiels de *GenSpec* a été réalisé et proposé dans le cadre d'une ré-ingénierie de *GenSpec* (cf. annexe J). Notons également qu'une critique des schémas de BD réalisés et implémentés pour notre solution est décrite dans le chapitre 7. Elle a pour but de proposer une alternative au schéma de BD relatif au glossaire, notamment dans la manière de gérer les relations entre les termes, les termes subjectifs et les synonymes.

---

<sup>3</sup>Par exemple, les identifiants facultatifs.

<sup>4</sup>Par exemple, le TE “*Exigences*” a été laissé au pluriel, le TE “*Utilisateurs*” n'a pas été renommé en “*Usager*”. Or, un “utilisateur” est un type d'usager de *GenSpec* et par conséquent une occurrence de la table “*Utilisateurs*”. La table “*Utilisateurs*” devrait être renommée en “*Usager*”



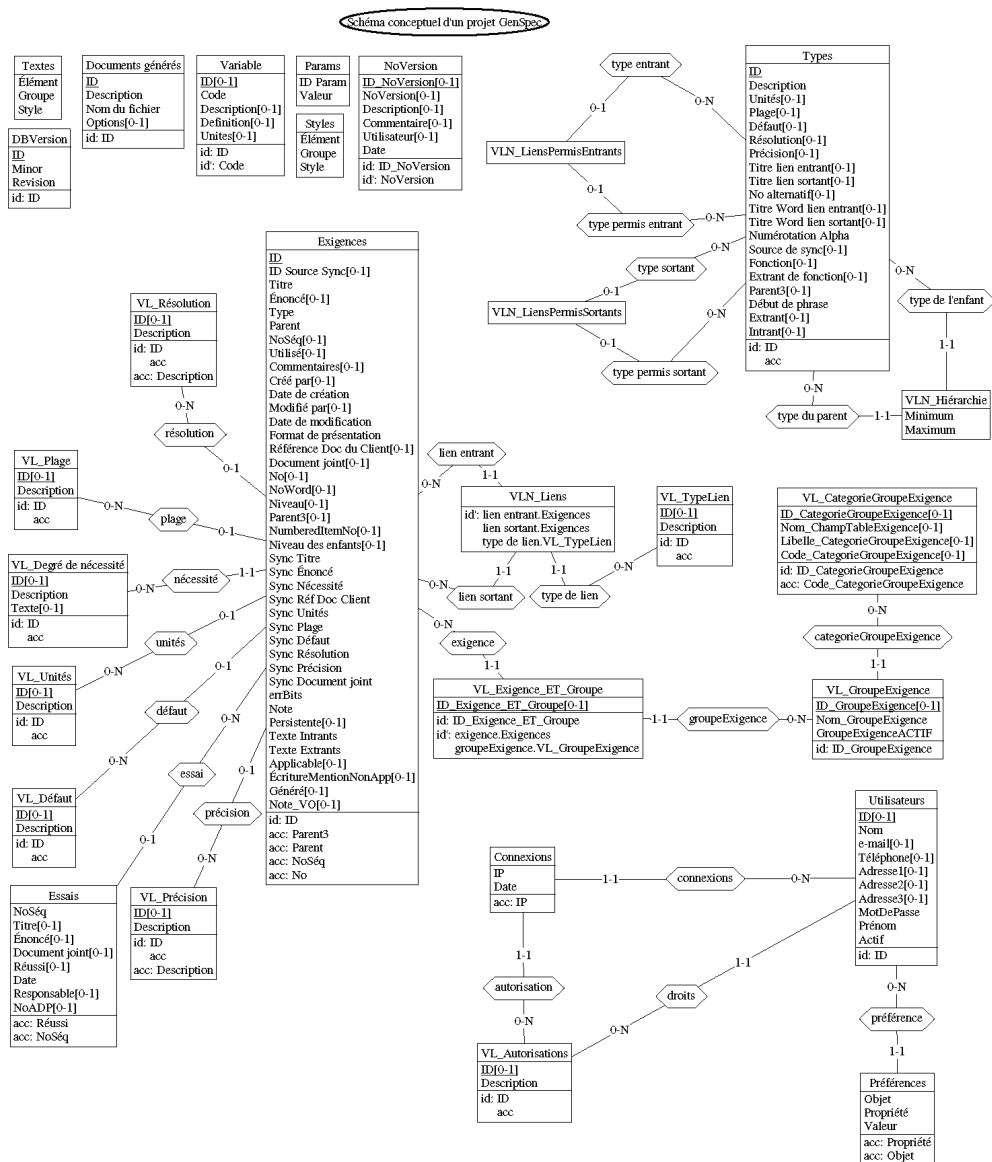


FIG. 5.6 – Schéma conceptuel entités-associations avant l'intégration des nouvelles fonctionnalités.

Par souci de clarté, deux types de schémas relatifs à la base de données de *GenSpec* ont été réalisés :

- (a) les schémas réduits décrivant uniquement les données relatives au glossaire et à la GMDE (et l'AEFC) (cf. FIG. 5.7) ;
- (b) le schéma global reprenant la totalité de la BD de *GenSpec* modifiée pour tenir compte des deux fonctionnalités supplémentaires (cf. FIG. 5.8).

La figure 5.7 présente le **schéma conceptuel réduit associé au glossaire**. Les TE “*Exigences*” et “*Utilisateurs*” appartiennent à la structure de données originales de la BD. Le TE *Exigences* ne reprend que l’attribut *ID* par concision. Le TE *Utilisateurs*<sup>5</sup> correspond aux types d’usagers de *GenSpec*.

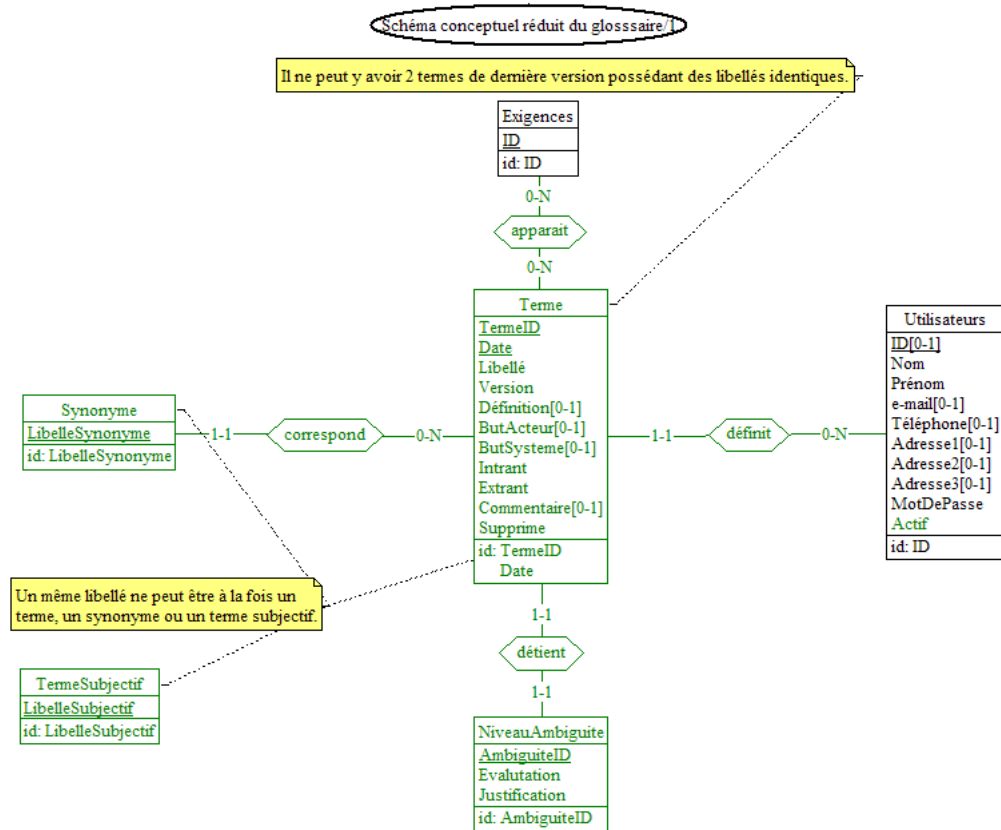


FIG. 5.7 – Schéma conceptuel entités-associations présentant la fonctionnalité du glossaire.

Le TE *Terme* reprend les informations sur un terme :

- **Date** : correspond à la date de création ou de modification de l’ensemble des informations associées à un terme.
- **Libellé** : correspond à la syntaxe du terme.
- **Version** : correspond au numéro de version évoluant selon les différentes modifications réalisées aux informations du terme.
- **Définition** : correspond à la sémantique du terme.
- **ButActeur** : décrit l’utilité du terme pour tout acteur lié au développement du projet. Un acteur est un humain ou un autre système.
- **ButSysteme** : décrit l’utilité du terme pour le système en développement.
- **Intransant** : détermine si le terme est un intransant ou non.
- **Extransant** : détermine si le terme est un extransant ou non.
- **Commentaire** : reprend les commentaires associés à un terme.
- **Supprime** : détermine si le terme est supprimé ou non. S’il est supprimé, le terme n’apparaît plus dans le glossaire.

Notons que le but pour l’acteur et le système est utilisé lorsque l’usager de *GenSpec* utilise le terme en tant qu’exigence de type intransant ou extransant.

<sup>5</sup>Le terme *Utilisateurs* pour le TE est inadéquat. Un utilisateur est un type d’usager et par conséquent une entité de la table *Utilisateurs*.

Un terme peut *apparaître* plusieurs fois dans plusieurs exigences et **apparaît obligatoirement dans le glossaire**. Un utilisateur peut définir plusieurs termes mais un terme n'est défini que par un seul utilisateur.

Un niveau d'ambiguïté (*NiveauAmbiguïté*) est défini. Il permet à un usager d'évaluer (et de justifier) les informations sur un terme. Une évaluation d'ambiguïté élevée marque la nécessité de fournir des informations plus appropriées.

Plusieurs *synonymes* (cf. sous-section 3.1.3) peuvent être associés à un terme. Un synonyme n'est associé qu'à un seul terme sinon deux termes du glossaire seraient synonymes indirectement. **Le glossaire associe une seule sémantique à un seul terme et inversement**. Les synonymes d'un terme sont listés afin de ne plus les rencontrer lors de la rédaction des exigences.

Des termes subjectifs (*TermeSubjectif*) (cf. sous-section 3.1.3) peuvent être définis pour éviter leur utilisation dans les exigences.

La figure 5.8 présente le **schéma conceptuel réduit à la GMDE et l'AECF**. Encore une fois, le TE "Exigences" appartient à la structure de données originale de la BD. Le TE *Exigences* ne reprend que l'attribut *ID* par concision.

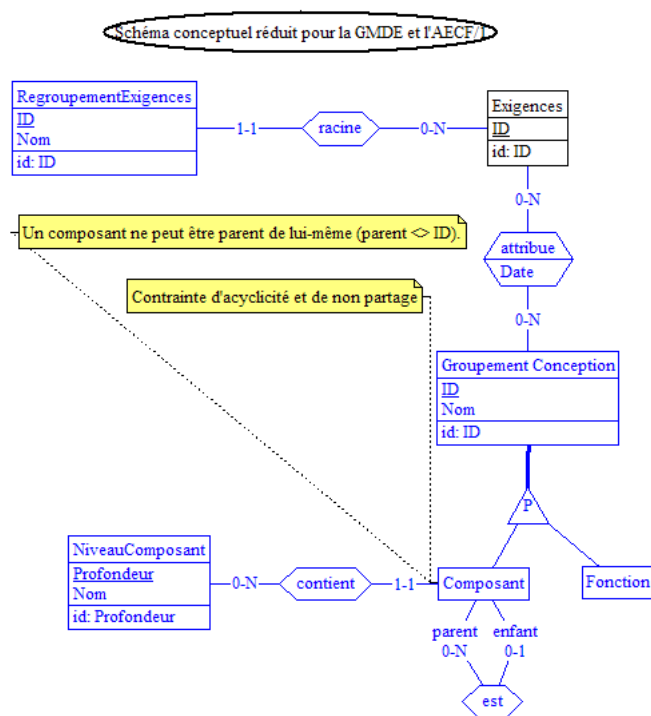


FIG. 5.8 – Schéma conceptuel entités-associations présentant la GMDE.

**GMDE** Un regroupement d'exigences (*RegroupementExigence*) possède une seule exigence comme racine. Pour rappel, un regroupement d'exigences est un ensemble hiérarchisé d'exigences (cf. section 3.2).

**AECF** Plusieurs exigences peuvent être allouées à plusieurs *Composants* et *Fonctions*. Pour rappel, un composant est un ensemble d'allocations d'exigences correspondant à un sous-système. Les composants sont structurés hiérarchiquement (cf. section 3.2). Une fonction est un ensemble d'allocations d'exigences déterminant une des fonctions du système à développer (cf. section 3.2).

Les exigences sont allouées pour établir le tableau de conception.

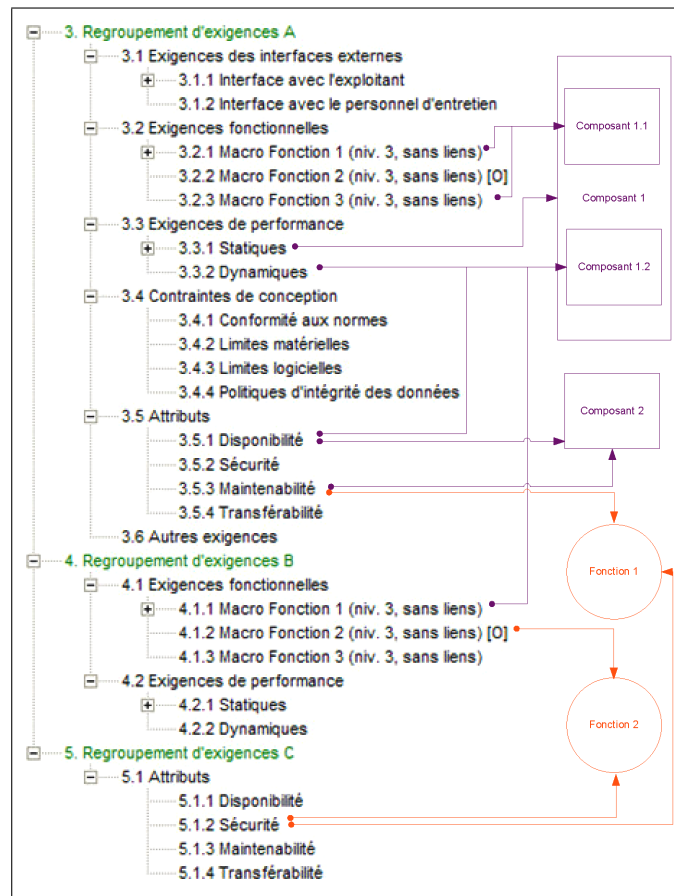


FIG. 5.9 – Schématisation des regroupement d'exigences, des composants et fonctions.

La figure 5.9 exprime le caractère transversal d'un regroupement d'exigences, d'un composant et d'une fonction. Une même exigence peut être allouée à plusieurs composants et fonctions différents. S'il s'agit de composants, une exigence peut être allouée à des composants de niveaux différents. A titre d'exemple (l'exemple est issu d'un des modèle de base de projet *GenSpec*), cette figure présente :

- **trois regroupements d'exigences** dont l'exigence racine est, pour chacun d'eux, colorée en vert ;
- **deux fonctions** colorées en orange incluant des allocations d'exigences de regroupements différents ;
- **plusieurs composants** colorés en mauve se partageant des allocations d'exigences provenant de regroupements d'exigences distincts.

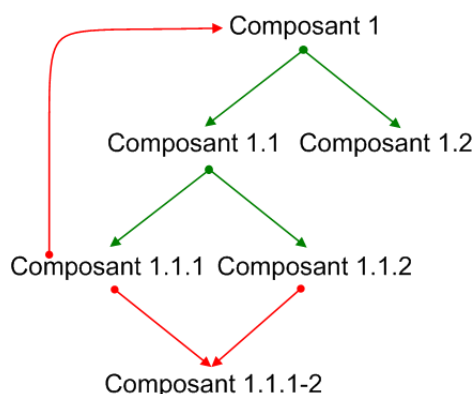


FIG. 5.10 – Structure autorisée et interdite pour les composants.

Un niveau (*NiveauComposant*) détermine la profondeur d'un composant dans la hiérarchie de composants. Un niveau est nommé et comprend plusieurs composants.

Les deux contraintes relatives aux composants (et non aux allocations) sont l'acyclicité entre les composants, leur imposant une structure d'arbre hiérarchique ainsi que le non partage entre les composants (flèches vertes). La figure 5.10 illustre les deux situations interdites (flèches rouges) découlant de ces deux contraintes. Notons que les allocations d'exigences à un composant appartiennent en conséquence aux composants parents.

La figure 5.11 présente le schéma logique de la GMDE et de l'AECF.

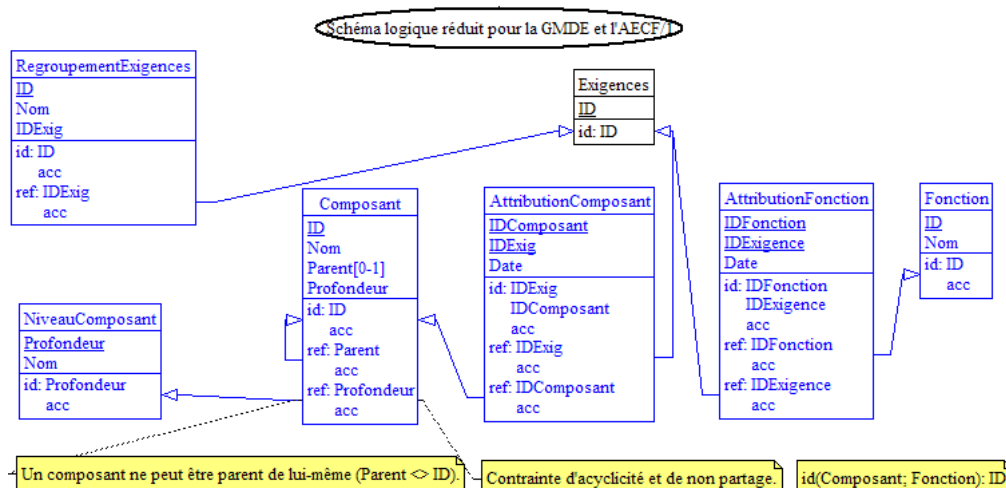


FIG. 5.11 – Schéma logique présentant la GMDE.

La transformation du schéma conceptuel en schéma relationnel peut, par la présence d'une relation "is-a", être réalisée de plusieurs manières [Hai02] :

- par une transformation ascendante ;
- par une transformation descendante ;
- par matérialisation.

Une transformation descendante a été réalisée. Le TE “*Groupement Conception*” n’apparaît alors pas dans le schéma logique. La transformation descendante a été préférée aux autres par la volonté de distinguer clairement les concepts de composants et de fonctions. Par conséquent, les attributs du TE *Groupement Conception* sont dans les TE “*Composant*” et “*Fonction*”. Signalons qu’une contrainte relative à l’unicité de l’identifiant entre la relation *Composant* et *Fonction* découle de la transformation.

La figure 5.12 présente le **schéma conceptuel entités-associations de *GenSpec*** après l’intégration des nouvelles fonctionnalités. Les TE et relations bleus font référence à la fonctionnalité de GMDE et d’AECF, les verts font référence au glossaire. Les TE noirs représentent l’état du schéma de BD restant inchangé avec les nouvelles fonctionnalités. Ce schéma conceptuel reprend un et un seul projet dans la BD. **Un projet *GenSpec* correspond donc à une seule BD.**

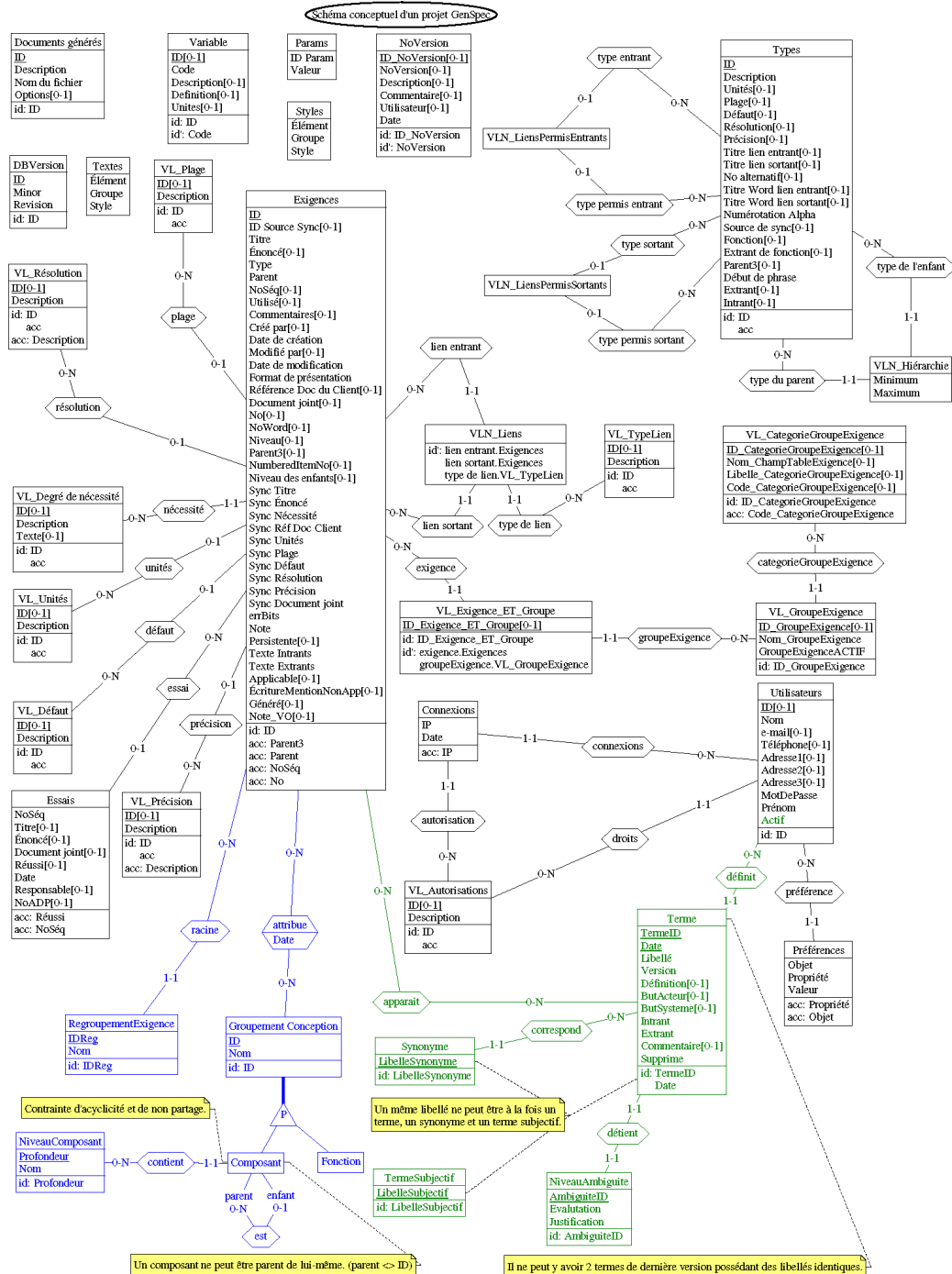


FIG. 5.12 – Schéma conceptuel entités-associations de GenSpec après l'intégration des nouvelles fonctionnalités.

## 2. Diagramme de composants

La figure 5.13 présente le diagramme des composants. Il est divisé en trois packages<sup>6</sup> : *Glossaire*, *Gestion multi-documents* (reprenant la GMDE et AECF) et *Commun*.

Les composants du package **Glossaire** sont :

- **Gestionnaire de recherche** : gère toutes les recherches relatives aux termes définis dans le glossaire de *GenSpec*.
- **Gestionnaire d'importation** : gère les importations de glossaires existants, ainsi que les incohérences pouvant découler de ces importations, notamment les conflits entre termes.
- **Gestionnaire des termes** : gère les opérations de création, de modification, de consultation et de suppression de termes.
- **Gestionnaire de détection** : gère les opérations de détection de synonymes et de termes subjectifs dans les exigences.

Les composants du package **Gestion multi-documents** sont :

- **Gestionnaire de vérification** : gère les vérifications liées aux regroupements d'exigences et aux composants et fonctions.
- **Gestionnaire des composants** : gère la hiérarchie des composants. Il permet de créer, renommer, supprimer des composants et permet l'allocation des exigences à des composants.
- **Gestionnaire des fonctions** : gère la liste des fonctions. Il permet de créer, renommer, supprimer des fonctions et permet l'allocation des exigences à des fonctions.
- **Gestionnaire des regroupements** : gère les regroupements d'exigences, notamment en donnant la possibilité d'ajouter, de renommer et de supprimer des regroupements d'exigences.

Les composants du package **Commun** sont :

- **Gestionnaire de la BD** : s'occupe des accès avec la base de données de *GenSpec*. Il est responsable de l'enregistrement permanent des informations ainsi que de la consultation des données enregistrées.
- **Gestionnaire des exigences** : s'occupe de la gestion des exigences (ajout, suppression, modification, etc.).
- **Gestionnaire de génération de documents** : gère tout ce qui est relatif à la génération de documents *Microsoft Word* ou *Microsoft Excel*. De plus, il fournit la possibilité de paramétrer les options de génération.
- **Gestionnaire intrant/extrant** : gère les interactions avec les utilisateurs. Il invoque les opérations sur les autres composants par l'intermédiaire de l'interface homme-machine.

## 3. Diagrammes de séquences

Différents diagrammes de séquences ont été réalisés pour représenter les interactions entre les différents composants. Le lecteur est prié de se référer aux annexes G et H pour les diagrammes de séquences réalisés respectivement pour le glossaire et la GMDE et AECF.

---

<sup>6</sup>Terme anglais signifiant "regroupement" référant, d'un part aux fonctionnalités de glossaire et de GMDE et AECF, d'autre part aux éléments communs aux deux fonctionnalités.



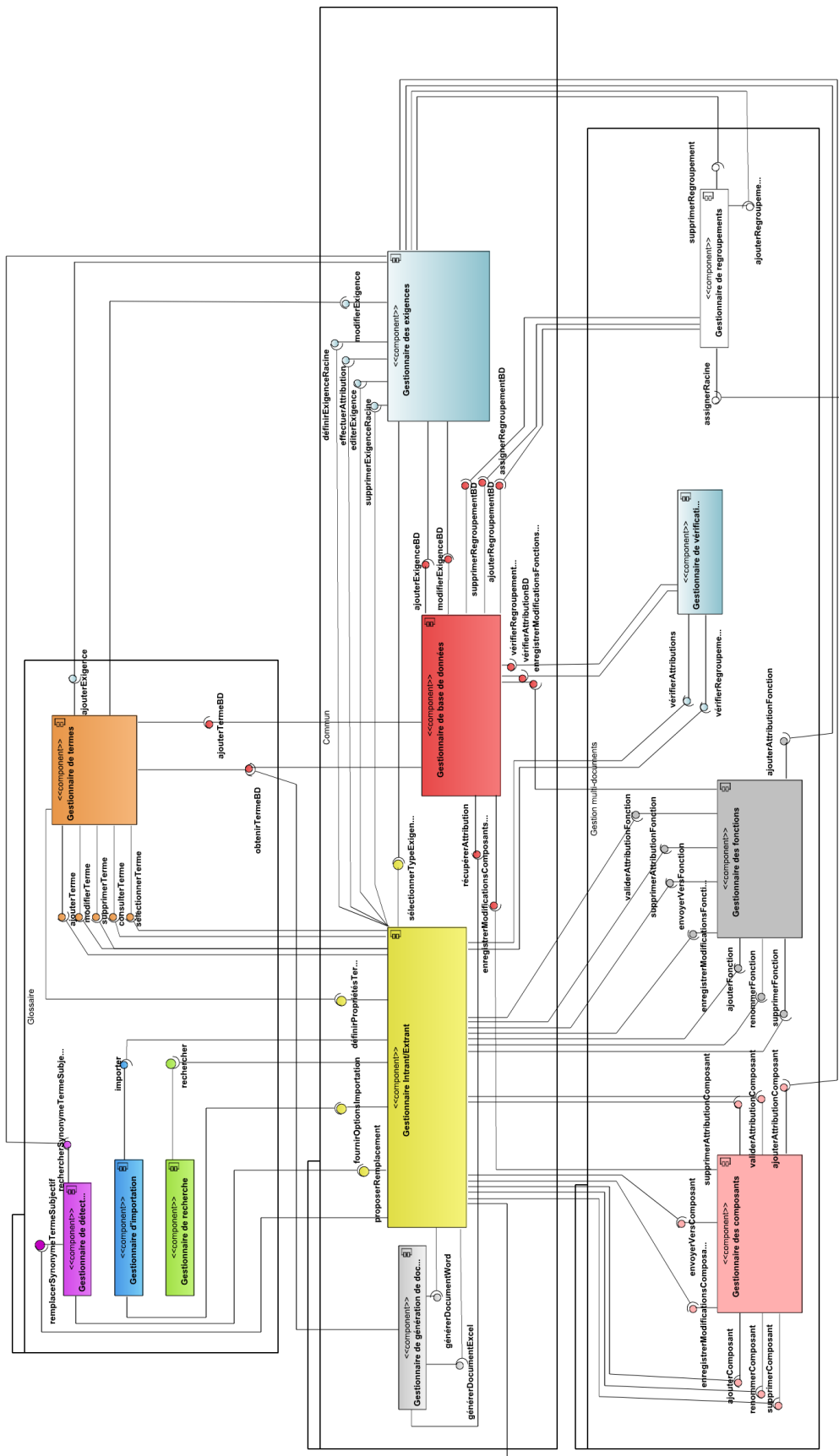


FIG. 5.13 – Diagramme de composants associé aux fonctionnalités de glossaire et de GMDE.

## 4. Diagramme de classes

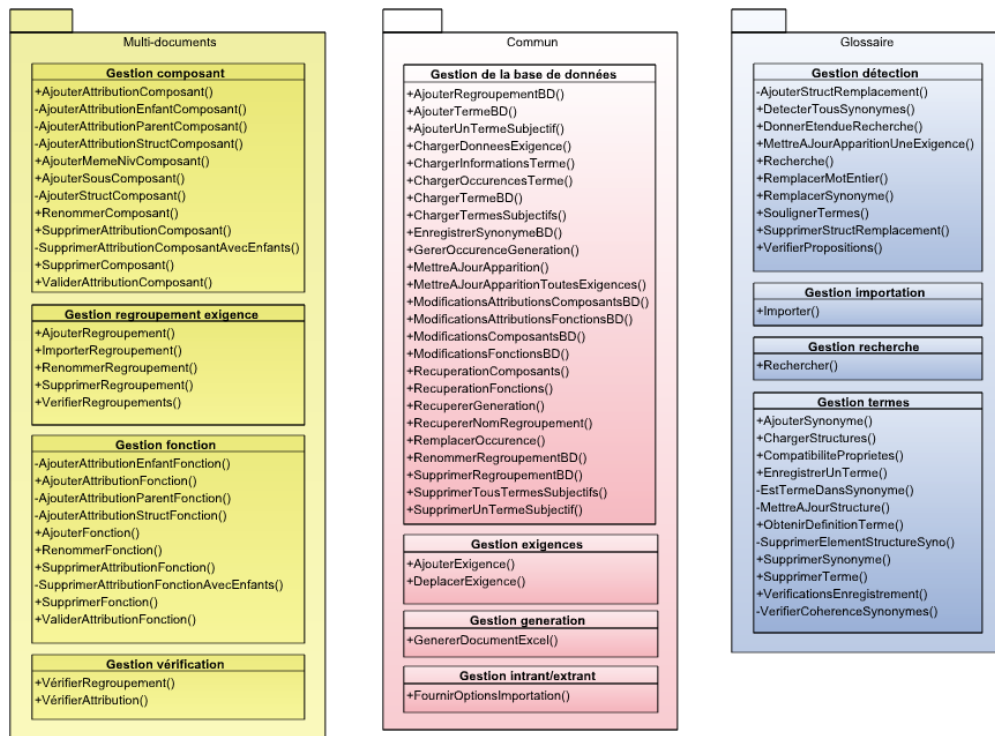


FIG. 5.14 – Diagramme de classes du glossaire et de la GMDE et l'AECE.

La figure 5.14 présente le diagramme de classes. Il est réparti sur les trois mêmes packages que le diagramme de composants.

Le diagramme de classes rassemble les méthodes relatives au glossaire, à la GMDE et de l'AECE. L'ensemble des méthodes existantes appartenant à *GenSpec* n'apparaît pas dans ce diagramme pour deux raisons :

- La concision ;**
- La structure du code existant :** la structure du code de *GenSpec* étant trop anarchique, il était impossible, à priori, de recenser les méthodes qui auraient subi des modifications en rapport avec les nouvelles fonctionnalités. Ce n'est que lors de l'implémentation que nous avons pu découvrir les méthodes nécessitant des modifications.

## 5.2.10 Découpage en incréments et itérations

**Objectifs** Le découpage des exigences en incréments et itérations a permis de développer les nouvelles fonctionnalités dans *GenSpec* de manière évolutive. À chaque incrément ou itération, une nouvelle version stable de *GenSpec* était réalisée.

**Procédure** L'ensemble des exigences a été subdivisé :

- en incréments pour le glossaire, chaque incrément ajoutant de nouvelles fonctionnalités [Hab07] ;
- en itérations pour la GMDE et l'AECE, chaque itération améliorant les fonctionnalités de la GMDE et l'AECE [Hab07].

Le découpage en itérations et incréments a été réalisé selon les priorités établies par le client 1. Les exigences, considérées comme essentielles, étaient dans les premiers incréments et itérations. Tandis que les exigences, considérées comme moins importantes ou optionnelles (tel que l'historique des termes), étaient dans les derniers incréments et itérations.

### 5.2.11 Implémentation

**Objectifs** Implémenter les exigences spécifiées de l'itération ou de l'incrément.

**Procédure** L'implémentation du glossaire, de la GMDE et l'AECF s'est basée sur les exigences établies à l'aide de *GenSpec* et sur le document de conception. Les nouvelles fonctionnalités ont été implémentées en *Visual Basic 6* comme le reste de *GenSpec*.

L'implémentation de la fonctionnalité de glossaire a été réalisée par le responsable de l'analyse et de la conception GMDE. La procédure inverse a été employée pour l'implémentation de la GMDE et AECF. La figure 5.1 représente cette dynamique. Cet échange a permis :

- de mettre en évidence les insuffisances et incohérences établies lors des phases d'élicitation et de conception par une lecture plus critique de l'analyse et de la conception (car le lecteur est différent de l'auteur) ;
- de comprendre de manière détaillée l'entièreté de la problématique liée au glossaire, à la GMDE et à l'AECF.

### 5.2.12 Vérification des exigences (tests)

**Objectifs** La vérification a pour objectif de vérifier si l'implémentation correspond aux exigences établies précédemment. On fait alors l'hypothèse que les exigences sont correctes [Eas05].

**Procédure** Après chaque incrément ou itération, l'analyste/concepteur (i.e. les stagiaires) vérifie le résultat de l'implémentation. Chaque exigence est testée pour mettre à jour les exigences non réalisées ou incomprises. Cette vérification permet également d'identifier des bugs éventuels dans l'implémentation réalisée.

### 5.2.13 Validation de l'incrément ou de l'itération (tests d'acceptation)

**Objectifs** La validation par le client 1 (car responsable du développement) de l'incrément ou de l'itération implémentée permet de :

1. vérifier la cohérence entre l'implémentation et les besoins réels des parties prenantes ;
2. mettre en évidence les besoins incompris.

La figure 1.1 à la section 1.1.2 présente la distinction entre la vérification et la validation.

**Procédure** La validation était effectuée par le client 1 après l'implémentation de chaque incrément ou itération. Une fois l'incrément ou l'itération testé(e) et validé(e) par le client, une nouvelle version finale de *GenSpec* est mise en place. Cependant, si l'incrément ou l'itération implémenté(e) ne répond pas aux besoins réels, il/elle est revu(e) et modifié(e) jusqu'à validation positive.

### 5.2.14 Maintenance des exigences et de la conception

**Objectifs** La maintenance des exigences et de la conception est essentielle dans le développement d'un projet. Elle permet de tenir à jour les besoins des différentes parties prenantes, les exigences étant toujours en constante évolution. Le but est de maintenir l'implémentation à jour.

**Procédure** Des exigences sont ajoutées et modifiées tout au long du développement d'un projet. Ces modifications proviennent de nouvelles demandes des parties prenantes, d'exigences floues, etc.

## 5.3 Problèmes rencontrés

Dans cette section, nous allons décrire les problèmes rencontrés lors de l'élaboration de la solution :

- les incohérences entre les acteurs du projet (5.3.1) ;
- la connaissance tacite (5.3.2) ;
- l'absence de documentation de l'outil (5.3.3).

### 5.3.1 Incohérence entre les acteurs du projet

Il existait plusieurs incohérences entre les différentes parties prenantes, notamment vis-à-vis de l'AECF :

- **Visions distinctes des besoins** : les parties prenantes avaient des besoins différents pour la même problématique. Le client 2 n'étant pas familiarisé avec *GenSpec* exprimait des besoins déjà solutionnés par l'OSIE à titre de nouveaux besoins. Les parties prenantes rencontraient alors des problèmes dans l'expression commune des besoins.
- **Vocabulaire identique se portant à des concepts différents** : l'ensemble des parties prenantes employait les mêmes termes avec une sémantique différente. Il était impossible pour les parties prenantes de définir clairement les concepts de composants et de fonctions utilisés. Ces termes étaient pourtant utilisés de manière courante. Après diverses réunions et interviews que nous avons menées, les parties prenantes sont arrivées à un accord sur la terminologie à utiliser ainsi que sur leur sémantique.

Ces incohérences ont entraîné certaines difficultés dans la compréhension de la problématique et dans la vision de la solution à réaliser.

### 5.3.2 Connaissance tacite

Le problème de la connaissance tacite est un phénomène connu en IE (cf. sous-section 1.1.2). Des informations importantes mais considérées comme évidentes par les parties prenantes restent inexprimées. A plusieurs reprises, nous avons été confrontés à ce problème.

### 5.3.3 Documentation de la base de données

Aucune documentation de la BD n'était disponible. Il ne nous était plus possible de communiquer avec la personne qui avait créé le schéma de BD car ce dernier avait quitté l'entreprise plusieurs années auparavant. Notre unique solution était de prendre connaissance nous-mêmes de ce schéma de BD en récoltant de l'information auprès d'utilisateurs de *GenSpec*. Notons également qu'aucune description des entités, relations et attributs n'était proposée.

Lors des phases de validation d'un incrément ou d'une itération par le client 1, il est arrivé que notre solution implémentée ne fonctionne pas avec de vrais projets car les bases de données avaient été modifiées par rapport au modèle original fourni, etc.

## 5.4 Résumé du chapitre

Ce chapitre a décrit le processus utilisé pour fournir une solution à la problématique définie au chapitre 3. Il décrit chacune des activités de ce processus de manière détaillée, ainsi que les livrables qui en résultent. Les différentes activités effectuées sont :

- élicitation des besoins ;
- analyse de l'existant ;
- modélisation des objectifs ;
- réalisation de cas d'utilisation ;
- réalisation d'un état de l'art des OSIE ;
- réalisation de prototypes d'interfaces graphiques ;
- réalisation d'une étude (basé sur le *ConOps*) ;
- rédaction des exigences ;
- conception de notre solution ;
- découpage de l'implémentation en incréments et itérations ;
- implémentation de notre solution ;
- vérification des exigences (tests) ;
- validation de l'incrément ou de l'itération implémenté (tests d'acceptation) ;
- maintenance des exigences et de la conception.

Les problèmes rencontrés décrivent les obstacles observés lors de la réalisation des activités du processus. Ces obstacles peuvent provenir des parties prenantes du projet ou d'un manque d'informations disponibles.

## Chapitre 6

# Description technique des solutions apportées à *GenSpec* par un exemple illustré

Ce chapitre présente la description technique des nouvelles fonctionnalités intégrées à *GenSpec*. Il présente :

1. un exemple illustrant les nouvelles fonctionnalités (6.1) ;
2. l'impact de l'intégration des nouvelles fonctionnalités dans *GenSpec* (6.2) ;

### 6.1 Exemple illustré

Cette section présente un exemple illustré des nouvelles fonctionnalités intégrées à *GenSpec*. Il se décompose en trois parties :

1. la définition de termes dans le glossaire (6.1.1) ;
2. la GMDE (6.1.2) ;
3. la conception du système par l'AECF (6.1.3).

**Remarques préliminaires** L'exemple présenté dans ce chapitre a pour objectif de montrer et de décrire les nouvelles fonctionnalités intégrées à *GenSpec*. L'exemple concerne un projet de gestion de conférences<sup>1</sup>. Nous supposons que plusieurs usagers sont chargés de rédiger les exigences, les organiser et de générer des documents d'exigences pour mettre en évidence les problèmes pouvant survenir lorsque plusieurs usagers travaillent simultanément sur un même projet.

Initialement, nous souhaitions établir un exemple concret de projet de l'unité Conception-Automatismes d'HQ. Cependant, le caractère confidentiel de leurs projets et l'utilisation de termes spécifiques au domaine d'application ne nous ont pas permis de proposer un tel exemple.

---

<sup>1</sup>Cet exemple est issu d'une étude de cas. Se référer à [FP04] pour l'énoncé complet de l'étude de cas.

### 6.1.1 Définition de termes dans le glossaire

**Définition de la terminologie** Lors de la rédaction des exigences, l'utilisateur peut choisir le vocabulaire à utiliser dans le projet. De cette manière :

- l'utilisateur peut associer une définition à un terme ;
- un autre usager peut se conformer au vocabulaire utilisé avant de rédiger les exigences.

La figure 6.1 présente la fenêtre du glossaire de *GenSpec*. Le glossaire contient une liste de termes. À chaque terme sont associées les informations suivantes :

FIG. 6.1 – Fenêtre du glossaire illustrant le terme “rapporteur”.

1. Le **libellé** du terme (ici “Rapporteur”).
2. La **version** de l'ensemble des informations. Elles renseignent :
  - l'auteur de la version des informations du terme (*Nicolas Pirmez*) et son type d'utilisateur (*administrateur*) ;
  - la date de création (ou de la dernière modification) du terme (*3/04/2006 14 :07 :29*).
3. Les **propriétés du terme** qui définissent si un terme peut être de type intrant et/ou extrant. Par exemple, dans le cas où le titre d'une exigence est identique au libellé d'un terme du glossaire, l'utilisateur de *GenSpec* peut créer de nouvelles exigences de type intrant ou extrant directement en glissant le terme du glossaire vers l'arbre hiérarchique des exigences. Inversement, il peut créer une nouvelle entrée dans le glossaire en glissant une exigence de l'arbre hiérarchique des exigences vers le glossaire (il lui reste alors à compléter les informations relatives au terme). Dans l'exemple, le terme n'est ni intrant ni extrant.
4. L'**identifiant** du terme lui associant un numéro unique dans le projet (*Numéro 1*).

5. La **définition** du terme : “*Chercheur ayant pour charge de lire les soumissions et élaborer un rapport pour chaque soumission.*”
6. Le **but pour l’acteur** : Aucun.
7. Le **but pour le système** : Aucun.
8. La **liste des synonymes** : Un seul synonyme est recensé pour le terme “*Rapporteur*” (“*Réviseur*”).
9. Le **niveau d’ambiguïté** permet d’évaluer l’ambiguïté d’un terme (selon l’ensemble de ses informations). Le niveau d’ambiguïté peut être élevé, moyen, faible selon le jugement de l’usager. Une justification peut accompagner le niveau d’ambiguïté. S’il n’y a aucune ambiguïté, le niveau d’ambiguïté est “aucun”.
10. La **liste des occurrences du terme** dans les exigences permet à l’usager de le renseigner sur la fréquence et l’endroit d’utilisation du terme dans les exigences. Pour chaque occurrence du terme, l’identifiant de l’exigence ainsi que son titre sont listés.
11. Les **commentaires** permettent l’ajout d’informations complémentaires.

**Gestion des synonymes** La figure 6.1 présente “*réviseur*” comme étant synonyme du terme “*rapporteur*”. Un (ou plusieurs) usager(s) peuvent utiliser le synonyme “*réviseur*” lors de la rédaction des exigences, amenant incohérence et ambiguïté dans le document d’exigences. Grâce au glossaire de *GenSpec*, les usagers peuvent pallier à ce problème de la manière suivante :

- Lorsque le terme “*rapporteur*” est utilisé lors de la rédaction d’une exigence pour la première fois, l’usager introduit toutes les informations dont il dispose au sujet de ce terme (définition, buts pour l’acteur, buts pour le systèmes, synonymes, etc.). L’usager fournit également d’éventuels synonymes au terme dont l’emploi est à éviter (par exemple “*réviseur*”).
- Si un usager utilise le synonyme “*réviseur*” au lieu du terme “*rapporteur*”, il est informé que le terme “*rapporteur*” est en vigueur dans ce projet. L’usager a deux alternatives :
  1. soit il se réfère à la définition de “*rapporteur*” établie par le premier usager et utilise ce terme dans les exigences ;
  2. soit il établit un niveau d’ambiguïté au terme “*rapporteur*” existant dans le glossaire ainsi qu’une justification car le terme “*réviseur*” lui paraît plus adéquat. Cette justification peut être à l’origine d’une discussion entre les usagers afin de s’accorder sur le vocabulaire à utiliser dans le projet.

**Niveau d’ambiguïté** La figure 6.2 présente l’évaluation du niveau d’ambiguïté du terme “*session*”. Un usager a associé à ce terme un **niveau d’ambiguïté** “*moyen*” car il juge que la définition (en accord avec sa justification) du terme est inadaptée. Les usagers peuvent par la suite discuter et s’accorder sur la définition à utiliser.

FIG. 6.2 – Niveau d’ambiguïté du terme “*session*”.



**Répercussion automatique d'une modification de toutes les occurrences d'un terme** Le glossaire permet à l'utilisateur de remplacer toutes les occurrences d'un terme par un de ses synonymes. La figure 6.3 expose le remplacement du terme “*session parallèle*” par son synonyme “*session simultanée*” dans les exigences et le choix de répercuter ce remplacement dans la totalité des exigences du projet. La décision d'utiliser un terme à la place d'un autre peut survenir suite à un accord entre plusieurs usagers.

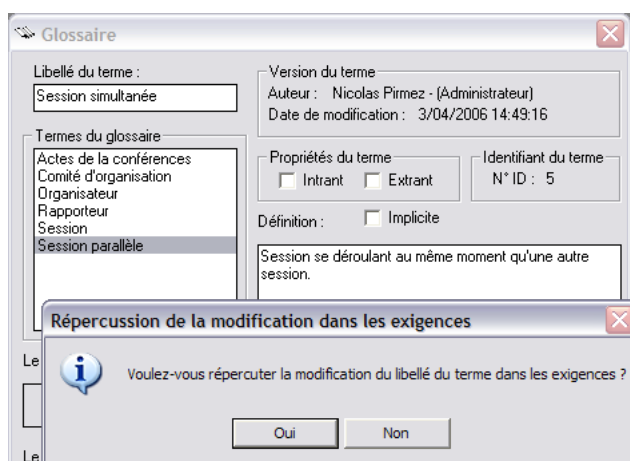


FIG. 6.3 – Répercussion du remplacement du terme “*session parallèle*” par son synonyme “*session simultanée*” dans les exigences.

**Détection de synonymes et termes subjectifs** Si un synonyme est détecté lors de la frappe, *GenSpec* suggère à l'utilisateur de le remplacer par le terme auquel il est associé. La figure 6.4 montre la reconnaissance automatique du synonyme “*session simultanée*” (dans la situation où le terme “*session parallèle*” lui est préféré).

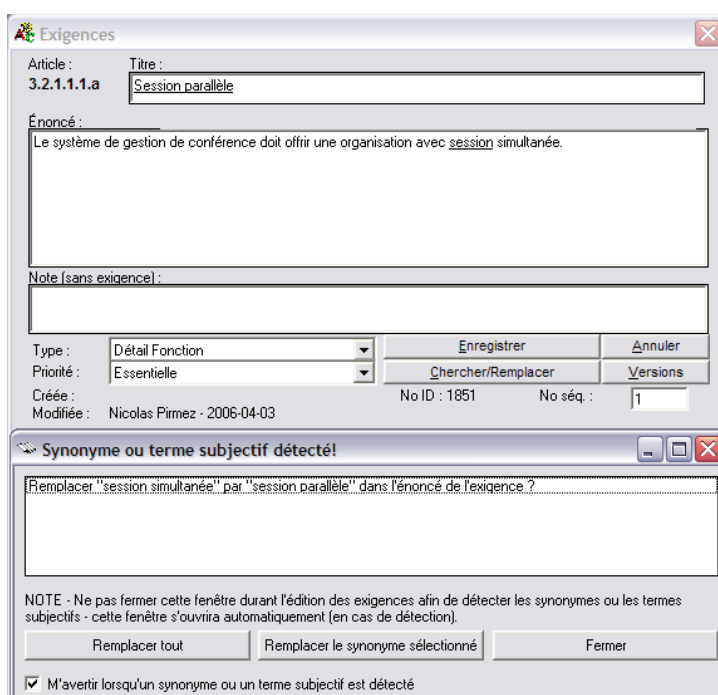


FIG. 6.4 – Détection du synonyme “*session simultanée*” par le terme “*session parallèle*” dans une exigence.

L'utilisateur peut choisir de remplacer le synonyme détecté grâce à la **fenêtre de détection de synonymes et termes subjectifs**. Notons que l'utilisateur n'est pas contraint de remplacer le synonyme par le terme lui étant associé, mais dans ce cas l'utilisateur est conscient qu'il amène de l'ambiguïté dans les exigences.

**Les termes définis** Lorsqu'un terme possédant une définition est utilisé dans les exigences, *GenSpec* le souligne dans la fenêtre des exigences. Le soulignement permet aux usagers d'identifier les termes définis et peuvent ainsi se conformer à la définition du terme pour s'assurer de son utilisation dans le bon contexte.

**Les termes subjectifs** Les termes subjectifs sont à éviter dans les exigences. Ces termes ont une sémantique qui dépend de l'affectivité de l'utilisateur. Prenons par exemple l'exigence suivante : "*Le système doit posséder une interface graphique ergonomique et agréable.*" La notion d'*ergonomie* ou d'*interface agréable* est subjective si aucun critère mesurable n'est défini. *GenSpec* permet également de détecter les termes subjectifs lors de la rédaction des exigences. Aucun terme ne lui est cependant proposé comme alternative.

**Définition des termes dans les documents d'exigences** Les termes définis dans le glossaire peuvent être mentionnés dans les documents d'exigences. Optionnellement, l'utilisateur peut intégrer la définition des termes après la première occurrence de chacun de ces termes dans les documents d'exigences générés. Cela permet au lecteur du document d'exigences de disposer de la définition du terme à proximité de l'exigence dans laquelle apparaît ce terme. La figure 6.5 présente une partie du document "spécification d'exigences" (plus précisément, l'exigence 1.2.1.1.1), dans laquelle les termes "*session*", "*session parallèle*" et "*session plénière*" sont définis. Notons que si le terme "*session*" est rencontré plus loin dans le document d'exigences, sa définition ne sera plus représentée.

**1.2.1.1.1. Type de session**

Le système de gestion de conférences doit permettre qu'une session soit une session parallèle ou une session plénière.

DEFINITIONS :

Session : Décomposition horaire d'une conférence.  
Session parallèle : Session se déroulant au même moment qu'une autre session.  
Session plénière : Session dont tous les membres sont convoqués.

FIG. 6.5 – Exigence accompagnée de définitions dans un document d'exigences.

### 6.1.2 GMDE

**La gestion des regroupements d'exigences** Elle permet à un projet *GenSpec* de contenir plusieurs regroupements d'exigences. Chacun de ces regroupements d'exigences peut contenir un ensemble d'exigences hiérarchisé en six niveaux au maximum. Notons que sans une telle gestion, un projet *GenSpec* ne peut contenir qu'un seul regroupement d'exigences. Chaque racine dans l'arbre des exigences représente un regroupement d'exigences.

La possibilité de définir plusieurs exigences racines est possible grâce à la définition d'exigences de même niveau de profondeur. Avant l'intégration de la fonctionnalité de GMDE, *GenSpec* proposait uniquement de créer des exigences sous une exigence existante, interdisant par conséquent la création de racines. Par exemple, une exigence racine d'un projet *GenSpec* peut être collée au même niveau qu'une autre exigence racine dans un autre projet *GenSpec*, créant dès lors un nouveau regroupement d'exigences. La figure 6.6 présente l'arbre des exigences contenant 3 regroupements (“*Soumission d'articles*”, “*Sessions*”, “*Création de la conférence*”).

Sans GMDE, l'utilisateur aurait dû créer un projet *GenSpec* par regroupement (dans l'exemple : 3 projets *GenSpec*). Par exemple, si un lien existait entre les exigences “*2.1.2 Types de sessions*” et “*3.1.1.3 Calendrier de la conférence*”, les deux problèmes suivants se poseraient à l'utilisateur :

1. il devrait mentionner manuellement, à l'aide d'une référence à un document externe, qu'un lien existe entre ces exigences ;
2. si une modification survenait pour une des exigences, il faudrait revérifier la cohérence entre les deux exigences (notamment si l'une change de titre d'exigence ou d'emplacement dans le projet *GenSpec*).

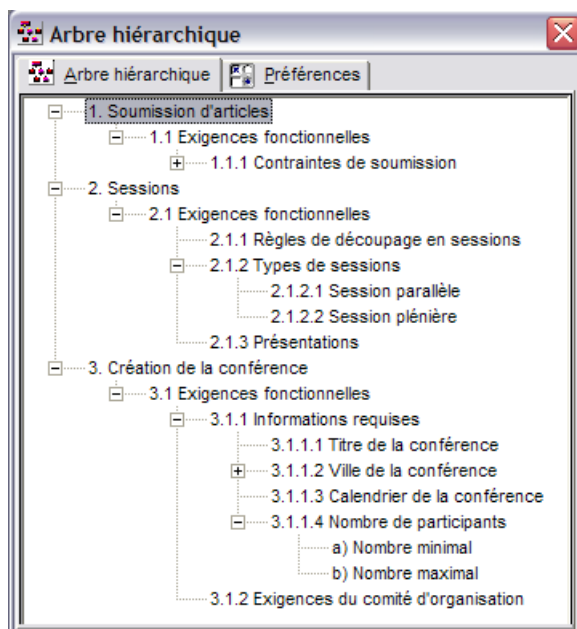


FIG. 6.6 – Arbre des exigences de *GenSpec*.

**La génération de documents** La génération de documents *Microsoft Word* permet d’inclure dans les documents d’exigences n’importe quel regroupement sélectionné par l’utilisateur (cf. FIG. 6.7). Les **options de génération** permettent à l’usager de générer de multiples types de documents (“*Spécification des exigences*”, “*Arbre hiérarchique des exigences*”, “*Cahier des essais*”, etc.) et ce, pour chacun des regroupements existants. L’usager de *GenSpec* peut disposer de documents d’exigences incluant :

- tous les regroupements existants (onglet “*Arbre hiérarchique*” dans la figure 6.7) rassemblant la totalité des exigences du projet ;
- une partie des regroupements existants, en choisissant de générer plus d’un regroupement ;
- un regroupement particulier, en choisissant un seul regroupement à générer.

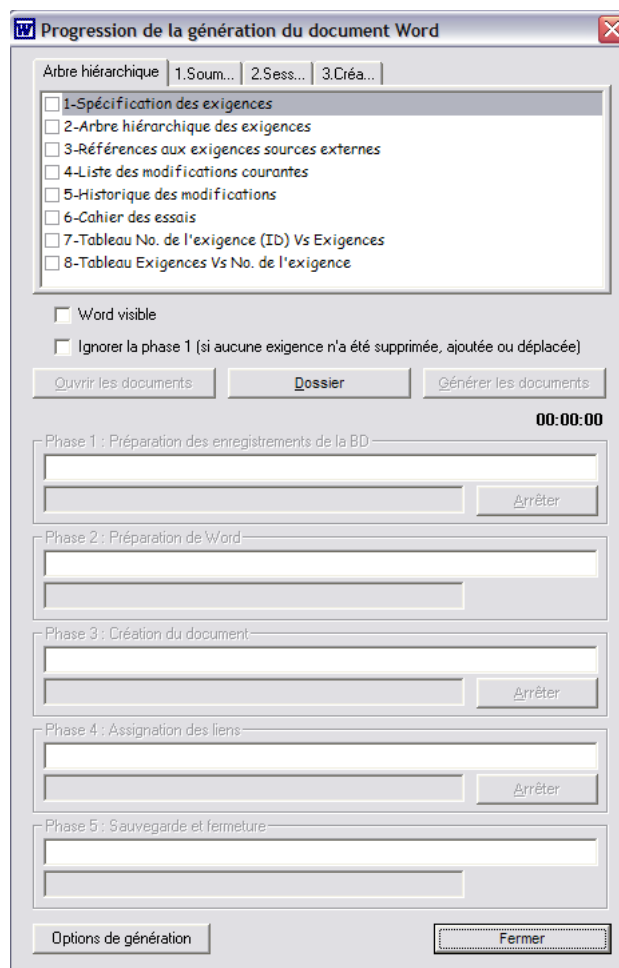


FIG. 6.7 – Fenêtre de génération *Microsoft Word*.

### 6.1.3 Conception du système avec l'AECF

**Composants et fonctions** Sur base des exigences définies par l'usager, le concepteur crée des **composants** et éventuellement des **fonctions** afin de réaliser la conception architecturale du système. La figure 6.8 présente une hiérarchie de composants établie par le concepteur du projet *GenSpec*. Les composants “*Gestion des présentations*”, “*Gestion des sessions*” et “*Gestion des soumissions*” comportent les allocations des exigences. La découpe du système en hiérarchie de composants est réalisée jusqu'à ce que chaque exigence soit allouée à un ou plusieurs composants.

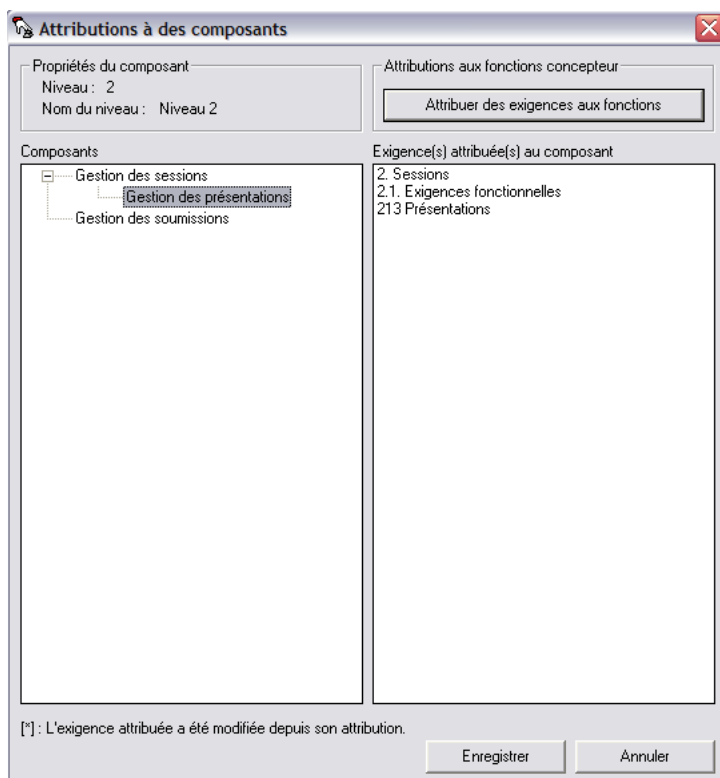


FIG. 6.8 – Exemple de composants du système.

Dans la situation où des modifications sont apportées aux exigences allouées, un **indicateur** (“[\*]”) prévient le concepteur afin qu’il s’assure qu’aucune incohérence ne survienne dans le tableau de conception (voir ci-dessous).

**Tableau de conception** À l’issue des allocations, le concepteur génère le **tableau de conception** (tableau *Microsoft Excel*). La figure 6.9 présente un tableau de conception généré par *GenSpec*. Chaque exigence compte un ensemble d’informations :

- **Produit** : le nom du produit résultant du développement du projet. Correspond au nom du projet *GenSpec* dans lequel apparaît l’exigence.
- **ID** : le numéro identifiant de l’exigence.
- **NoWord** : le numéro de l’exigence dans le document d’exigences *Microsoft Word* généré.
- **Titre** : le titre de l’exigence.
- **Énoncé** : l’énoncé de l’exigence.
- **Fonction** : la fonction à laquelle l’exigence est allouée.
- **Composant** : le composant auquel est alloué l’exigence.

Notons que dans le cas où l'exigence est allouée à plusieurs fonctions ou plusieurs composants, l'exigence apparaît plusieurs fois dans le tableau.

Produit	ID	NoWord	Titre	Énoncé	Fonction	Composant
Le système de gestion de conférences	0	2	Sessions	permettre d'organiser les sessions.	Assurer le déroulement de la conférence	Gestion des présentations
Le système de gestion de conférences	146	2.1	Exigences fonctionnelles	connaître les informations suivantes pour la création d'une conférence :	Assurer le déroulement de la conférence	Gestion des présentations
Le système de gestion de conférences	1880	2.1.3	Présentations	permettre de diviser les sessions en présentations.	Assurer le déroulement de la conférence	Gestion des présentations
Le système de gestion de conférences	0	2	Sessions	permettre d'organiser les sessions.	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	146	2.1	Exigences fonctionnelles	connaître les informations suivantes pour la création d'une conférence :	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	1861	2.1.1	Règles de découpage en sessions	respecter les règles de découpage en sessions. (voir document joint).	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	1877	2.1.2	Types de sessions	supporter les deux types de sessions suivantes :	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	1878	2.1.2.1	Session parallèle	permettre d'organiser des sessions parallèles.	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	1879	2.1.2.2	Session plénière	permettre d'organiser des sessions plénières.	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	1880	2.1.3	Présentations	permettre de diviser les sessions en présentations.	Assurer le déroulement de la conférence	Gestion des sessions
Le système de gestion de conférences	1860	1	Soumission d'articles	gérer les soumission d'articles	Assurer la gestion des soumissions d'article	Gestion des soumissions
Le système de gestion de conférences	1867	1.1	Exigences fonctionnelles	satisfaire aux exigences fonctionnelles suivantes :	Assurer la gestion des soumissions d'article	Gestion des soumissions
Le système de gestion de conférences	1868	1.1.1	Contraintes de soumission	satisfaire au contraintes de soumission suivantes :	Assurer la gestion des soumissions d'article	Gestion des soumissions
Le système de gestion de conférences	1876	1.1.1.1	Date limite	s'assurer que la soumission n'est pas acceptée si elle arrive après la date limite de soumission.	Assurer la gestion des soumissions d'article	Gestion des soumissions

FIG. 6.9 – Tableau de conception.

## 6.2 Intégration avec l'existant

L'intégration des nouvelles fonctionnalités dans *GenSpec* implique des :

- modifications de privilèges aux usagers ;
- changements dans les interfaces graphiques.

**Modifications des privilèges des usagers** Les droits suivants, relatifs aux fonctionnalités de glossaire et de GMDE incluant l'AECF, sont associés aux différents types d'utilisateurs de *GenSpec* :

- l'**administrateur** : l'administrateur possède tous les droits vis-à-vis du glossaire et de la GMDE. Il peut notamment définir des termes, modifier les informations qui leur sont associées ou supprimer des termes. L'administrateur peut créer des regroupements d'exigences, les renommer, les supprimer et effectuer les opérations de déplacement d'exigences entre ces regroupements.
- l'**utilisateur** : l'utilisateur possède autant de droits que l'administrateur vis-à-vis du glossaire et de la GMDE. Notons cependant que l'utilisateur ne peut utiliser l'AECF.
- le **testeur, commentateur et visiteur** : ils ne peuvent, vis-à-vis du glossaire, que consulter les informations qui sont associées à un terme et générer les documents en conséquence. En ce qui concerne la GMDE, ils ne peuvent que consulter les exigences existantes (sauf le commentateur qui peut y apposer des commentaires). L'AECF ne leur est pas disponible.
- le **concepteur** : le concepteur est un nouveau type d'utilisateur. Il dispose des mêmes droits que le testeur, le commentateur et le visiteur vis-à-vis du glossaire. Par contre, il est le seul, hormis l'administrateur, à pouvoir utiliser l'AECF.

**Changements dans les interfaces graphiques** L'intégration des nouvelles fonctionnalités dans *GenSpec* implique des modifications et des nouveautés dans les interfaces graphiques.

- Le **glossaire** constitue à lui seul une nouvelle fenêtre autonome dans l'OSIE et contient l'ensemble des informations associées à un terme. Outre ces informations, nous retrouvons la liste des termes définis dans le glossaire. A partir de cette liste, il est alors aisé de consulter toutes les informations associées à un terme dans cette même fenêtre. Le menu de génération de documents est également plus complet car il contient des informations associées directement au glossaire.
- La possibilité de créer plusieurs **regroupements d'exigences** dans un projet *GenSpec* permet de les paramétrer indépendamment des autres. La fenêtre de génération de documents propose, pour chaque regroupement d'exigences, un onglet qui lui est propre permettant de lui définir ses options de génération.
- Concernant l'**AECF**, de nouvelles fenêtres ont été créées afin de représenter les allocations d'exigences avec les composants et fonctions. Cette fenêtre est accessible par l'intermédiaire d'un nouveau menu dans la fenêtre principale de *GenSpec*. La génération du tableau de conception se réalisant sous un format tabulaire, un nouveau type de document (en plus de *Microsoft Word*) a été intégré à l'outil et ce, par une nouvelle fenêtre destinée à la génération de documents de type *Microsoft Excel*. Les options de génération du document *Microsoft Excel* sont disponibles dans une nouvelle fenêtre.

## 6.3 Résumé du chapitre

Ce chapitre a présenté les nouvelles fonctionnalités intégrées à *GenSpec* par l'intermédiaire d'un exemple illustré. Les différentes captures d'écran présentent en détail les nouvelles fonctionnalités apportées à l'outil. Des extraits de documents *Microsoft Word* et *Microsoft Excel* sont présentés pour montrer l'impact des nouvelles fonctionnalités dans les documents d'exigences produits par *GenSpec*.

L'intégration des nouvelles fonctionnalités entraîne des changements dans l'environnement de *GenSpec*. Les usagers existants voient leurs privilèges adaptés et un nouvel usager est créé (le concepteur) afin de réaliser l'AECF. De nouvelles fenêtres, permettant l'utilisation des nouvelles fonctionnalités, provoquent un changement dans les interfaces graphiques de l'outil.





# Chapitre 7

## Évaluation de l'approche

Ce chapitre comporte une description de :

1. l'évaluation de la démarche adoptée (7.1);
2. l'évaluation de l'implémentation (7.2).

### 7.1 Évaluation de la démarche

Cette section décrit les avantages et les limites de la démarche suivie.

#### 7.1.1 Avantages

Les avantages de notre démarche sont :

- la réalisation de documents de conception ;
- une documentation adaptée ;
- l'échange des documents d'exigences et de conception ;
- le recours à un processus de développement itératif et incrémental ;
- la réalisation de prototypes d'interfaces ;
- la réalisation d'un état de l'art.

**Conception** Nous avons utilisé différents diagrammes UML afin de modéliser les différents aspects de notre solution à développer. Nous avons utilisé :

1. des diagrammes de cas d'utilisation ;
2. un diagramme de composants ;
3. des diagrammes de séquences ;
4. un diagramme de classe ;
5. un diagramme d'état.

À cela s'ajoutent les diagrammes entités-associations qui ont permis de modéliser les modifications à apporter à la BD.

**Documentation adaptée** Afin de faciliter l'intégration et la maintenance des nouvelles fonctionnalités, nous avons documenté la BD existante de *GenSpec* afin de comprendre les concepts existants liés à ces nouvelles fonctionnalités. Cette documentation nous a permis de comprendre la totalité des deux fonctionnalités de sorte à pouvoir les implémenter. Notons également que cette documentation facilitera les développements futurs de *GenSpec*.

**Echange des documents d'exigences et de conception** Par l'échange des documents d'exigences et de conception, nous avons pu évaluer la qualité de notre analyse et de notre conception. Cela nous a permis également de comprendre l'ensemble de la problématique liée au développement des nouvelles fonctionnalités de *GenSpec*.

**Processus itératif et incrémental** Nous avons opté pour un processus itératif et incrémental pour intégrer les nouvelles fonctionnalités à *GenSpec*. Cela nous permettait de découper les exigences selon les types et les priorités des besoins ainsi que de réaliser une phase de vérification et de validation à chaque itération ou incrément. Chaque réalisation d'une itération ou d'un incrément a débouché sur une nouvelle version officielle de *GenSpec*.

**Réalisation de prototypes d'interface graphiques** La réalisation de prototypes d'interface graphique s'est révélée bénéfique pour supporter l'élicitation des exigences. Ces interfaces graphiques ont constitué un réel support lors des discussions avec les parties prenantes. Elles ont participé à la compréhension de la problématique et à la validation de notre vision des besoins.

**Réalisation d'un état de l'art** Une analyse d'un ensemble d'OSIE autre que *GenSpec* a permis d'identifier les éléments standards et les originalités des différentes solutions apportées par plusieurs OSIE vis-à-vis de la problématique décrite au chapitre 3. Cet état de l'art a permis également de détecter les solutions inadaptées et de confirmer notre compréhension des besoins.

### 7.1.2 Limites

Les limites de la démarche adoptée sont :

- la formalisation de la planification limitée ;
- les phases de tests limitées.

**Formalisation de la planification limitée** La planification de notre travail a été réalisée de manière informelle. Nous n'avons pas utilisé un formalisme, tel qu'un *diagramme de Gantt*<sup>1</sup>, pour planifier les différentes étapes de notre développement. Cependant, nous avons établi des dates de remises et une estimation du temps nécessaire pour chaque activité à l'aide de *Microsoft Outlook*. Cette démarche permettait au client 1 de suivre l'avancement des différentes activités du développement de manière informelle.

**Phases de tests limitées** Après la réalisation d'un incrément ou d'une itération, une phase de tests était réalisée afin de vérifier la concordance entre les exigences et leur implémentation. Chaque exigence était vérifiée individuellement. Cependant, l'attention portée à ces phases de tests aurait pu être plus importante. Sous la pression des fonctionnalités à implémenter, les phases de tests ont été quelque peu réduites et certains cas n'ont pas pu être complètement vérifiés. Nous aurions pu, après chaque itération ou incrément, utiliser obligatoirement un projet réel afin de tester l'ensemble des nouvelles fonctionnalités développées. Cette démarche, bien que réalisée à plusieurs reprises, n'a pas été suivie assidûment.

---

<sup>1</sup>Outil utilisé en ordonnancement et gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. Il permet de représenter graphiquement l'avancement du projet [Wik07b].

## 7.2 Évaluation de l'implémentation

Cette section décrit :

1. les aspects non implémentés des fonctionnalités (7.2.1) ;
2. les avantages de notre implémentation (7.2.2) ;
3. les limites de notre implémentation (7.2.3).

### 7.2.1 Description des aspects non implémentés des fonctionnalités

Cette sous-section présente les aspects non implémentés, par manque de temps, des nouvelles fonctionnalités. Leurs objectifs étaient d'ajouter ou de raffiner les aspects de fonctionnalités implémentés. Ils sont découpés en trois parties :

#### 1. Le **Glossaire** :

- l'**importation** de glossaires : l'importation des données contenues dans des glossaires existants permettent d'importer, d'un projet *GenSpec* à l'autre, les termes et leurs informations associées. L'importation de glossaires requiert une gestion permettant de tenir compte des conflits. Par exemple, dans le cas où un même terme se retrouve dans le projet source et cible de *GenSpec*, il faut assister l'importation afin de définir la politique à suivre (remplacer ou non).
- la **recherche** : la recherche de termes dans le glossaire à partir de différents types de critères. Par exemple, retrouver un terme à partir de mots-clés, de sa définition ou de ses synonymes.
- la **vérification** liée aux termes du glossaire : la vérification étend la reconnaissance de synonymes et de termes subjectifs dans le glossaire même de *GenSpec*, mais également les vérifications d'orthographe dans les informations relatives à un terme.
- la **gestion des termes prioritaires** : les termes prioritaires sont des termes, non définis dans le glossaire, qui apparaissent de nombreuses fois dans les exigences. Disposer d'une liste de termes prioritaires permet de recenser les termes importants à définir.
- l'**historique** : l'historique du glossaire basé sur un mécanisme de version de termes permet, d'une part, de retenir l'ensemble des modifications (création, modification et suppression) apportées aux termes du glossaire et d'autre part, de consulter les modifications d'un terme appartenant au glossaire. Chaque modification est accompagné d'une justification décrivant la raison du changement.

#### 2. La **GMDE** :

- la **gestion multi-fenêtres** : la gestion multi-fenêtres permet de représenter un regroupement d'exigences par nouvelle fenêtre. Grâce à l'utilisation de telles fenêtres, l'analyste peut travailler sur un regroupement d'exigences sans se soucier des éventuels autres regroupements d'exigences. Cela lui permet également de déplacer, copier, coller et créer des liens plus aisément entre des exigences appartenant à des regroupements d'exigences distincts. Grâce à une disposition adaptée de ces fenêtres, l'analyste peut visualiser plusieurs regroupements d'exigences simultanément dans des fenêtres distinctes sur un même écran.
- la **gestion de la vérification** : la gestion de la vérification permet de détecter les exigences n'ayant aucune allocation dans les composants (ou fonctions) définis par le concepteur de *GenSpec*. Ces exigences sont donc définies par l'analyste mais ne

se retrouvent dans aucun composant (ou fonction) du système.

- la **gestion de la synchronisation d'exigences** : la gestion de la synchronisation d'exigences consiste, d'une part, en la génération d'un document regroupant l'ensemble des exigences synchronisées entre elles et d'autre part, en l'amélioration du mécanisme existant en terme de stockage dans la BD des exigences synchronisées. Une exigence synchronisée avec une autre signifie qu'une modification apportée à l'une se répercute automatiquement sur l'autre.

### 3. L'AECF :

- L'exigence non implémentée relative à l'AECF consiste en la **copie d'allocations** : la copie d'allocations sert à dupliquer ou à déplacer une allocation existante d'un composant à un autre composant ou une fonction existante.

## 7.2.2 Avantages

Les avantages de notre solution sont :

- Uniformisation du vocabulaire ;
- la gestion efficace de la génération des documents d'exigences ;
- le support de l'activité de conception.

**Uniformisation du vocabulaire** L'utilisation d'un glossaire permet aux différents usagers de s'accorder sur le vocabulaire à utiliser et réduit ainsi les ambiguïtés et les incohérences dans les exigences.

**Gestion efficace de la génération des documents d'exigences** La fonctionnalité de GMDE offre une plus grande flexibilité dans la gestion des exigences d'un projet *GenSpec*, et permet également une plus grande flexibilité vis-à-vis de la génération des documents d'exigences. Cette flexibilité s'explique par le choix offert à l'utilisateur de personnaliser les différents documents d'exigences par regroupements d'exigences, et non plus uniquement par projet *GenSpec* tout entier.

**Support de l'activité de conception** L'AECF facilite l'activité de conception. Une notation signale les modifications effectuées sur les exigences depuis leurs allocations à la génération du tableau de conception.

## 7.2.3 Limites

Les limites de notre solution sont :

- la gestion limitée de l'ambiguïté ;
- la nécessité de définir les synonymes de chaque terme ;
- les relations entre les termes, synonymes et les termes subjectifs ;
- la détection inexistante a posteriori des synonymes et des termes subjectifs.

Notons que certaines de ces limitations sont notamment dues à un manque de temps nécessaire pour implémenter certaines exigences spécifiées en début de projet (cf. sous-section 7.2.1).

**Gestion limitée de l'ambiguïté** La fonctionnalité de glossaire pour *GenSpec* a pour objectif de répondre à l'ambiguïté lexicale des exigences. Les autres types d'ambiguïté décrits au chapitre 3 ne sont donc pas pris en compte. Une extension du glossaire pourrait apporter, par des techniques plus avancées que celles intégrées à *GenSpec*, une solution à l'ensemble

des 4 types d'ambiguïtés. Ce qui en théorie permettrait, par l'intermédiaire de *GenSpec*, la rédaction de documents d'exigences non ambigus.

**Nécessité de définir les synonymes associés à un terme par l'utilisateur** Afin d'uniformiser le vocabulaire à utiliser dans les exigences, les usagers doivent lister les synonymes et raccourcis des termes présents dans le glossaire. Cette tâche peut se révéler fastidieuse si le nombre de termes est conséquent. L'importation de glossaire pré-existant, tel que spécifié mais non implémenté, fournirait une solution adéquate.

**Redéfinition des relations entre termes, synonymes et termes subjectifs** Notre schéma de BD relative au glossaire pourrait mettre différemment en relation les termes, synonymes et termes subjectifs. Nous aurions pu établir une relation de spécialisation entre ces trois TE (cf. FIG 7.1). Les termes du glossaire, les synonymes de ces termes et les termes subjectifs forment une partition du TE “*Terme*”. Chacune des spécialisations de “*Terme*” peut à un moment ou à un autre être défini dans le glossaire. Notons également que le TE “*Niveau Ambiguïté*” est représenté de sorte à ce que chaque terme possède un niveau d'ambiguïté accompagné d'une justification.

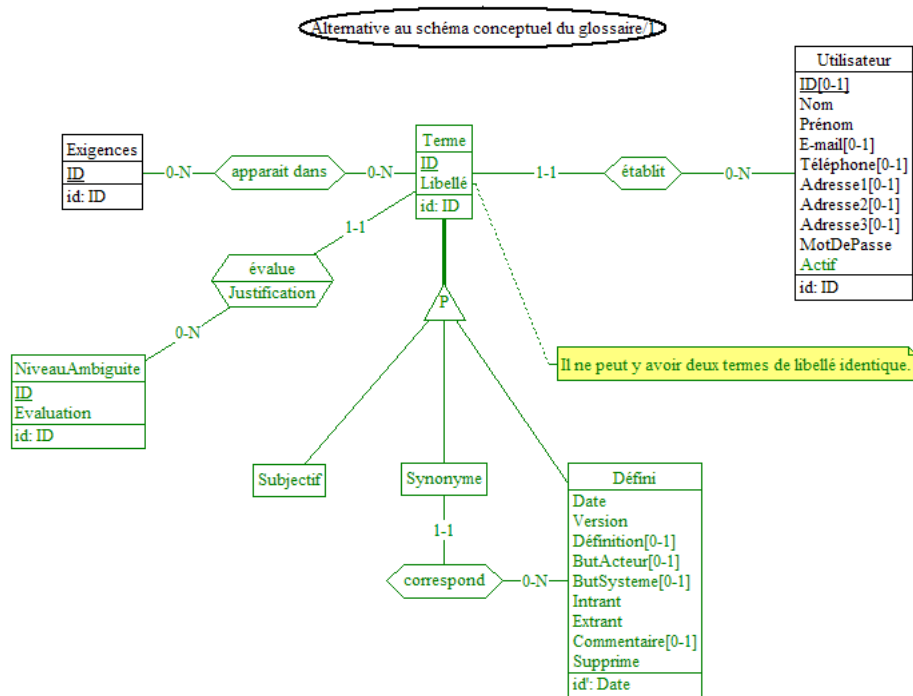


FIG. 7.1 – Alternative au schéma de BD du glossaire.

Dans la solution implémentée, aucune relation de spécialisation n'apparaît entre ces trois TE. Établir une relation entre le TE “*Terme*” et les TE “*Synonyme*” et “*TermeSubjectif*” semble être tout aussi adapté que la solution implémentée.

**Détection a posteriori des synonymes et termes subjectifs inexistante** Un aspect de la fonctionnalité de glossaire permet de détecter les synonymes et les termes subjectifs au moment de la frappe. Cependant, il n'est pas possible, lors de la vérification générale des exigences, de détecter d'éventuels synonymes ou termes subjectifs présents dans les exigences.

### 7.3 Résumé du chapitre

Ce chapitre a relevé les avantages et les limitations relatives, d'une part, à la démarche adoptée et d'autre part, à l'implémentation des fonctionnalités intégrées à *GenSpec*. Certaines limitations sont dues à un manque de temps suffisant pour implémenter les exigences spécifiées. Ces aspects non implémentés ont notamment pour objectif d'ajouter ou de raffiner les aspects de fonctionnalités implémentés.







# Conclusion et Perspectives

Ce mémoire avait pour objectif de présenter et de justifier des solutions visant à améliorer la qualité des exigences produites par l'OSIE *GenSpec*.

*GenSpec* représente les exigences en LN. Comme nous l'avons développé dans ce mémoire, l'utilisation du LN pour la rédaction des exigences amène systématiquement de l'ambiguïté susceptible d'apporter des interprétations distinctes d'une même exigence. L'utilisation possible de différents termes pour référer à une même sémantique ajoute de l'ambiguïté aux exigences. Les erreurs qui en résultent peuvent être coûteuses si elles ne sont pas détectées tôt dans le développement d'un projet.

*GenSpec* regroupe les exigences en projet *GenSpec*. Plusieurs projets *GenSpec*, correspondant aux regroupements d'exigences, sont nécessaires pour rédiger les exigences d'un projet. Comme cela a été présenté, une même exigence peut appartenir à des regroupements d'exigences différents entraînant une gestion manuelle de la traçabilité et de tout autre type de liens entre ces exigences ainsi qu'une répétition de celles-ci amenant de la redondance.

*GenSpec* ne fournit pas de support à la conception architecturale d'un système. Les exigences sont allouées manuellement aux différents composants du système. Cette gestion manuelle est susceptible d'occasionner des erreurs.

Nous avons développé différentes fonctionnalités répondant à ces problèmes. Pour diminuer l'ambiguïté, nous avons opté pour la réalisation d'un glossaire avancé permettant de décrire un ensemble d'informations relatives aux termes utilisés. Ce glossaire permet d'uniformiser la rédaction des exigences et d'accorder les usagers quant à la terminologie à utiliser. Le risque d'erreur lors de l'implémentation se voit alors réduit par une meilleure qualité des exigences.

Pour la GMDE, nous avons opté pour une gestion centralisée de la totalité des exigences qui permette de les regrouper en un seul projet *GenSpec*. Il en découle une gestion plus efficace de la traçabilité et des liens entre exigences, ainsi qu'une diminution de la redondance des exigences. Comme nous l'avons vu, la GMDE permet de réduire les erreurs liées à la gestion des exigences produisant des exigences et des documents d'exigences de meilleure qualité.

Pour l'AECF, nous avons opté pour l'allocation d'exigences dans une hiérarchie de composants et une liste de fonctions. Ces allocations permettent au concepteur de réaliser la conception architecturale d'un projet. L'AECF fournit ainsi à *GenSpec* la possibilité de faire le lien entre l'ingénierie des exigences et l'activité de conception.

Les perspectives de travail relatives aux solutions apportées et à *GenSpec* sont présentées ci-après.

Premièrement, l'ensemble des exigences que nous avons rédigé et modélisé pourraient être implémenté. Étant donné la quantité des exigences demandées et le temps imparti, nous avons implémenté en premier lieu les exigences prioritaires en accord avec le responsable du développement. L'implémentation des autres aspects liés aux fonctionnalités serait facilité par les documents d'exigences et de conception que nous avons établis.

Deuxièmement, outre l'ambiguïté lexicale, il serait intéressant de traiter les autres types d'ambiguïté.

Troisièmement, un support plus flexible de l'ensemble des regroupements d'exigences pourrait être développé. Ce support permettrait l'insertion d'images ou de différents diagrammes, éventuellement établis par un autre outil, pour compléter l'ensemble des exigences établies. Les différents regroupements constitueraient alors des dossiers et non plus une exigence ordinaire.

*GenSpec* est un OSIE en constante évolution. L'ajout de nouvelles fonctionnalités est planifié. Outre ces fonctionnalités, nous avons relevé différentes perspectives de travail.

Premièrement, une réingénierie de la BD pour limiter la redondance des données ainsi qu'une centralisation et une uniformisation dans le code des appels à la BD (actuellement dispersés). En plus de la suppression de la redondance, cela améliorerait la performance générale de *GenSpec*. À ce titre, une proposition de schéma de BD reprenant les aspects essentiels de *GenSpec* est consultable à l'annexe J.

Deuxièmement, il faudrait faire évoluer le code existant en *Visual Basic 6* vers un langage plus évolutif.

Troisièmement, *GenSpec* devrait pouvoir supporter d'autres formalismes, tels qu'UML.

Remarquons enfin que depuis notre intervention, *GenSpec* dispose de moyens permettant d'éviter l'ambiguïté lexicale dans les exigences, de gérer plus efficacement les regroupements d'exigences et dispose d'un mécanisme d'allocation à des composants et fonctions. Ces atouts apportent davantage de rigueur, de flexibilité et facilite l'utilisation à ses usagers.

# Bibliographie

- [Art] Artisan Studio. “*Manuel d’aide de Artisan Studio 6.1.21*”.
- [BKK03] D. Berry, E. Kamsties, and M. Krieger. “From Contract Drafting to Software Specification : Linguistic Sources of Ambiguity”. Technical report, University of Waterloo, University of Essen and University of California, Canada (Waterloo), Germany (Essen), USA (Los Angeles), November 2003.
- [Bra02] I. K. Bray. “*An Introduction to Requirements Engineering*”. Addison Wesley, first edition, August 2002.
- [Buj04] R. Bujold. “Ingénierie des exigences : Une méthode simple et systématique”. *IEEE Canadian Review*, (48) :6–10, Fall 2004. (<http://iee.ca/canrev/cr48/pages6-10.pdf>).
- [Buj05] R. Bujold. “Ingénierie des exigences : L’outil de support GenSpec”. *IEEE Canadian Review*, (51) :13–16, Fall 2005. (<http://www.evh.ieee.org/reg/7/canrev/cr51/CR51-GenSpec.pdf>).
- [Cha06] F. Chantree. “*Identifying Nocuous Ambiguity in Natural Language Requirements*”. PhD thesis, The Open University, Milton Keynes, UK, September 2006.
- [Coc00] A. Cockburn. “*Writing Effective Use Cases*”. Addison-Wesley Professional, first edition, January 2000.
- [DBRT01] A. Durán, B. Bernárdez, A. Ruiz, and M. Toro. “An XML-based Approach for the Automatic Verification of Software Requirements Specifications”. 2001.
- [ddBdD03] Laboratoire d’Ingénierie des Bases de Données. “Les outils CASE”. Technical report, University of Namur, Namur, Belgium, September 2003. (<http://www.info.fundp.ac.be/libd/jibd/Intro-CASE.pdf>).
- [def04] “L’ingénierie système”, November 2004. <http://www.afis.fr/praoout/ingsys/ingsys.htm> (Date of access 08/05/07).
- [Dev07] G. Deville. “Introduction aux méthodes et concepts essentiels d’ingénierie linguistique”. Technical report, University of Namur, Namur, Belgium, 2006-2007.
- [Dor99] M. Dorfman. “Requirements Engineering”. *SEI Interactive*, page 30, March 1999. (<http://www.sei.cmu.edu/news-at-sei/features/1999/mar/Background.mar99.pdf>).
- [Eas05] S. Easterbrook. “Requirements Engineering”. Technical report, University of Toronto, department of computer science, Toronto, Ontario, Canada, 2005. (<http://www.cs.toronto.edu/sme/CSC2106S/>).
- [Fir03] D. Firesmith. “Modern Requirements Specification”. *Journal of Object Technology*, 2(1) :53–64, March-April 2003. ([http://www.jot.fm/issues/issue\\_2003\\_03/column6](http://www.jot.fm/issues/issue_2003_03/column6)).
- [FP04] J. M. Favre and I. Parissis. “Case study - Gestion d’une conférence”. Technical report, Institut d’informatique et mathématiques appliquées de Grenoble, Grenoble, France, August 2004.
- [GM03] F. J. Garcia and M. N. Moreno. “C-REQUIREMENTS SPECIFICATION TEACHING”. In *33rd ASEE/IEEE Frontiers in Education Conference*, volume 3, pages S2C 1–6, Boulder, Colorado, USA, November 2003. (<http://fie.engrng.pitt.edu/fie2003/papers/1128.pdf>).

- [Gro94] The Standish Group. “The CHAOS Report”, 1994.
- [Hab07] N. Habra. “*Processus logiciels et modèles du cycle de vie*” (Note de cours). Namur, Belgium, 2006-2007.
- [Hai02] J. L. Hainaut. “*Ingénierie des Base de Données*” (Note de cours). Namur, Belgium, 2002.
- [HD97] E. Haywood and P. Dart. “Towards Requirements for Requirements Modelling Tools”. Technical report, University of Melbourne, department of computer science, October 1997. (<http://citeseer.ist.psu.edu/111985.html>).
- [Hey05] P. Heymans. “*Analyse et modélisation de Systèmes d’Information*” (Note de cours). University of Namur (FUNDP), Namur, Belgium, 2004-2005.
- [Hyd07a] “Hydro-Québec”, April 2007.  
<http://www.hydroquebec.com> (Date of access 04/04/07).
- [Hyd07b] Hydro-Québec. “*Manuel d’aide de GenSpec 6.4.24*”, February 2007.
- [IEE98] IEEE-SA Standards Board. “*IEEE Guide for Information Technology - System Definition - Concept of Operations Document*”, 1998.
- [IEE04] IEEE Computer Society. “*Requirements for Requirements Management Tools*”, Kyoto, Japan, 2004. 12th IEEE International Requirements Engineering Conference (RE’04).
- [Inc07] “INCOSE Requirements Management Tools Survey”, 2007.  
<http://www.paper-review.com/tools/rms/read.php> (Date of access 08/05/07).
- [IRq] IRqA. “*Manuel d’aide de IRqA 3.5.0*”.
- [Jac98] M. Jackson. “*Software Requirements & Specifications*”. Addison Wesley, 1998.
- [LD04] M. Lang and S. Duggan. “A Tool to Support Collaborative Software Requirements Management”. *Requirements Engineering*, 6(3) :161–172, February 2004.
- [LW03] D. Leffingwell and D. Widrig. “*Managing Software Requirements : A Use Case Approach*”. Addison-Wesley, second edition, May 2003.
- [Mat05] R. Matulevičius. “*Process Support for Requirements Engineering : A Requirements Engineering Tool Evaluation Approach*”. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2005.  
(<http://www.diva-portal.org/ntnu/theses/abstract.xhtml?dbid=576>).
- [not07] “Notation Z”, May 2007.  
<http://vl.zuser.org/french/> (Date of access 10/05/07).
- [Poh93] Klaus Pohl. “the three dimensions of requirements engineering”. In *Conference on Advanced Information Systems Engineering*, pages 275–292, 1993.
- [sit06] “SysML : vers un langage de modélisation unifié pour l’IS”, August 2006.  
<http://www.afis.fr/praoout/modelisation/modelis6.htm>  
(Date of access 08/05/07).
- [Sut02] A. Sutcliffe. “*User-Centred Requirements Engineering*”. Springer, May 2002.
- [Syn06] “Synonymes et antonymes”, December 2006.  
[http://fr.wiktionary.org/wiki/Aide:Synonymes\\_et\\_antonymes](http://fr.wiktionary.org/wiki/Aide:Synonymes_et_antonymes)  
(Date of access 29/12/06).
- [Syn07] “Synonymie”, February 2007.  
<http://fr.wikipedia.org/wiki/Synonymie> (Date of access 06/02/07).
- [Tel] Telelogic. “*Manuel d’aide de Telelogic Doors 8.1*”.
- [Web07] “Natural Language”, March 2007.  
<http://se.uwaterloo.ca/dberry/natural.language.html>  
(Date of access 30/03/07).
- [Wie99] K. Wieger. “Automating Requirements Management”. *Software Development*, July 1999. ([http://www.processimpact.com/articles/rm\\_tools.pdf](http://www.processimpact.com/articles/rm_tools.pdf)).
- [Wik06] “Homonyme”, December 2006.  
<http://fr.wiktionary.org/wiki/homonyme> (Date of access 17/12/06).

- 
- [wik07a] “Brainstorming”, March 2007.  
*http://fr.wikipedia.org/wiki/Brainstorming* (Date of access 27/03/07).
- [Wik07b] “Diagramme de Gantt”, March 2007.  
*http://fr.wikipedia.org/wiki/Diagramme\_de\_Gantt* (Date of access 05/05/07).
- [Wik07c] “Effet Hawthorne”, March 2007.  
*http://fr.wikipedia.org/wiki/Effet\_Hawthorne* (Date of access 30/03/07).
- [Wik07d] “Waterfall model”, May 2007.  
*http://en.wikipedia.org/wiki/Waterfall\_model* (Date of access 03/05/07).



Facultés Universitaires Notre-Dame de la Paix, Namur  
Institut d'informatique.

**Année académique 2006-2007**

**Améliorations de l'outil  
d'Ingénierie des Exigences  
*GenSpec***

**Annexes**

*Olivier Pire et Nicolas Pirmez*

Mémoire présenté en vue de l'obtention du grade de  
Maître en Informatique.









# Description des annexes

Nous décrivons brièvement ci-dessous les 12 annexes de ce mémoire :

L'**annexe A** présente les cas d'utilisation relatifs à la fonctionnalité de glossaire.

L'**annexe B** présente les cas d'utilisation relatifs à la fonctionnalité de GMDE (incluant l'AECE).

L'**annexe C** présente l'étude préliminaire concernant le développement de la fonctionnalité de glossaire. Notons que cette étude se base sur le standard IEEE 1362 de documents de conception opérationnel (ConOps).

L'**annexe D** présente l'étude préliminaire concernant le développement de la fonctionnalité de GMDE (incluant l'AECE). Notons que cette étude se base sur le standard IEEE 1362 de documents de conception opérationnel (ConOps).

L'**annexe E** est le document d'exigences, issu de *GenSpec*, présentant la fonctionnalité relative au glossaire. Il reprend "*la spécification des exigences*" et "*l'arbre hiérarchique des exigences*".

L'**annexe F** est le document d'exigences, issu de *GenSpec*, présentant la fonctionnalité relative à la GMDE (incluant l'AECE). Il reprend "*la spécification des exigences*" et "*l'arbre hiérarchique des exigences*".

L'**annexe G** correspond au document de conception du glossaire. Il décrit les différents diagrammes UML réalisés.

L'**annexe H** correspond au document de conception de la GMDE (AECE). Il décrit les différents diagrammes UML réalisés.

L'**annexe I** fournit une définition des différents types d'exigences ainsi qu'un ensemble d'informations et de règles générales sur la gestion de l'ambiguïté au sein des exigences. Ce résumé a été établi à partir de Berry, Kamsties et Krieger [BKK03].

L'**annexe J** propose un schéma conceptuel alternatif de la BD d'un projet *GenSpec*. Il correspond à une nouvelle structure de la BD dans le cas d'une ré-ingénierie totale de *GenSpec*.

L'**annexe K** présente notre article issu des développements des nouvelles fonctionnalités de *GenSpec*. Notons que cet article a été soumis à deux revues (*La revue canadienne de l'IEEE* et *Génie Logiciel*) ainsi qu'à la conférence *ICSSEA 2007*. Nous en espérons une publication dans ces revues et une présentation lors de cette conférence.

L'**annexe L** reprend l'ensemble des tableaux définissant les différentes exigences fonctionnelles et non-fonctionnelles du cadre d'évaluation appliqué à *GenSpec* (cf. chapitre 2).



## **Annexe A**

# **Cas d'utilisation : Glossaire**

La dernière modification du contenu de ce document date du 4 octobre 2006.

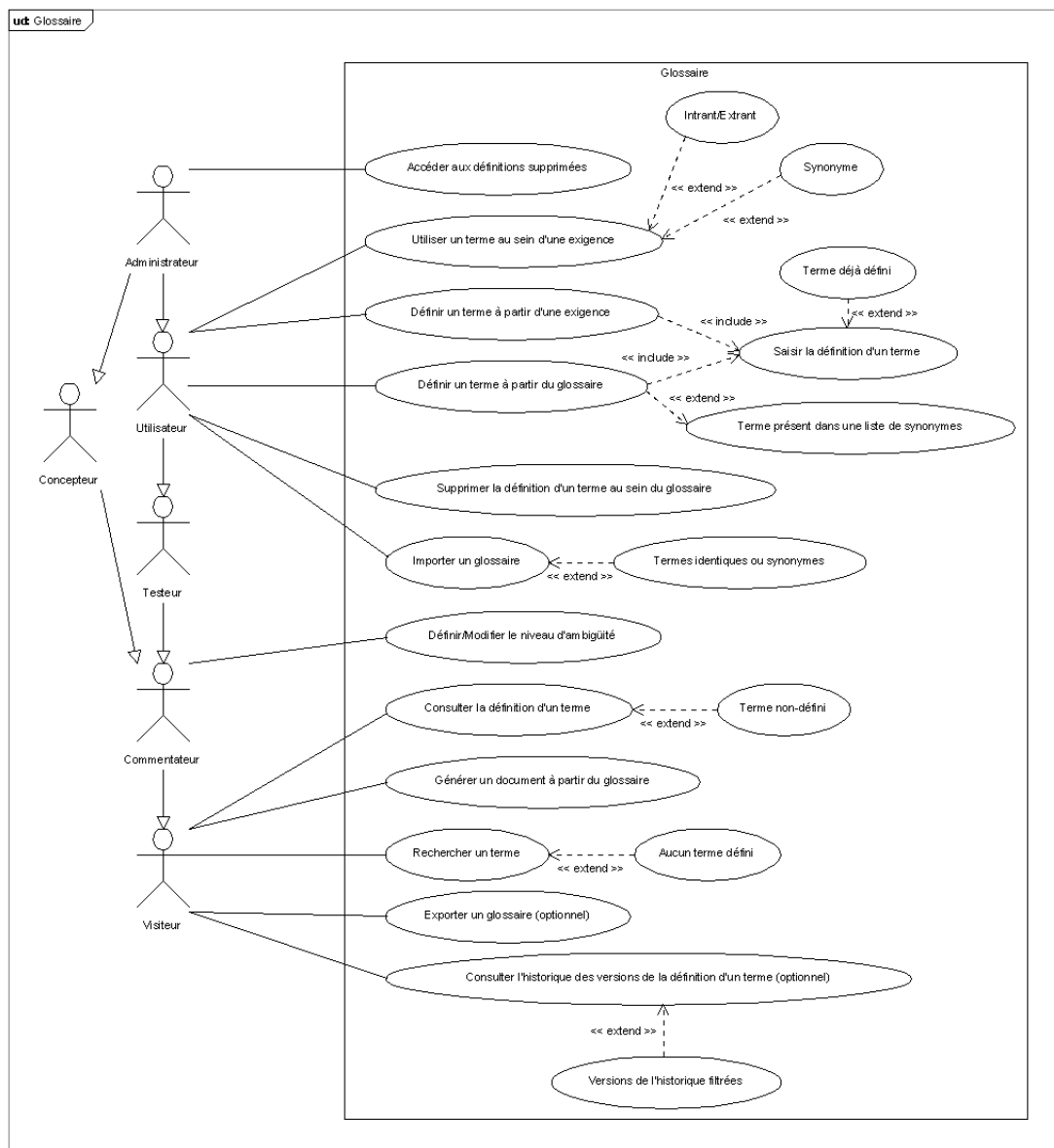
## A.1 Diagramme des *Cas d'utilisation*

Cette section comporte le diagramme des *Cas d'utilisation* (cf. FIG. A.1). On peut retrouver 6 acteurs sur ce diagramme correspondant aux droits que possèdent les différents utilisateurs du logiciel *GenSpec*.

### A.1.1 Les différents acteurs

Il existe 6 types d'acteurs au sein de *GenSpec*. Ceux-ci sont distingués par le type de droit qui leurs sont accordés vis-à-vis de l'élaboration des exigences :

- Administrateur : L'administrateur a accès à toutes les fonctionnalités de *GenSpec*.
- Utilisateur : Correspond à l'administrateur sauf qu'il ne peut gérer les différents accès des usagers ni générer de versions officielles ni également consulter les définitions des termes supprimés.
- Visiteur : Le visiteur n'a qu'un droit de consultation et de génération de documents sur l'ensemble des données.
- Commentateur : Le commentateur correspond au visiteur à ceci près qu'il peut compléter les champs liés aux commentaires.
- Concepteur : Le concepteur possède les mêmes droits que le commentateur excepté qu'il peut réaliser un regroupement des exigences en composants.
- Testeur : Le testeur, quant à lui, possède les mêmes droits que le commentateur mais peut également compléter les champs liés aux tests.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

FIG. A.1 – Diagramme des cas d'utilisation du glossaire.



## A.2 Scénarii

Dans cette partie, nous présentons les actions possibles par les différents acteurs du système sous forme de *Cas d'utilisation*. Les différents scénarii décrivent les *Cas d'utilisation* représentés dans le diagramme précédent.

La description des scénarii respecte le canevas suivant :

NOM  
BRÈVE DESCRIPTION  
PRÉ CONDITIONS GLOBALES  
POST CONDITIONS GLOBALES  
CAS NORMAL :  
    PRÉ CONDITIONS NORMALES  
    POST CONDITIONS NORMALES  
    FLUX NORMAL  
CAS ALTERNATIF :  
    PRÉ CONDITIONS SPÉCIFIQUES  
    POST CONDITIONS SPÉCIFIQUES  
    FLUX ALTERNATIF

*Remarques :*

- Ce canevas n'est pas suivi par le *Cas d'utilisation* à inclure car ses pré et post conditions dépendent du *Cas d'utilisation* appelant.
- Les PRÉ CONDITIONS GLOBALES et POST CONDITIONS GLOBALES correspondent à des *Pré et post conditions* s'appliquant sur l'ensemble des flux correspondants. Ces conditions peuvent être vide dans le cas où cela dépend uniquement de la spécificité du flux.
- Nous faisons l'hypothèse générale que le système fonctionne correctement.

### A.2.1 Cas d'utilisation 1 (à inclure) : Saisir la définition d'un terme

**Brève Description :** Le système demande à l'utilisateur de lui fournir la définition d'un terme et vérifie l'ensemble des informations fournies.

#### Cas Normal

##### Flux Normal :

<i>Utilisateur</i>	<i>Système</i>
2. L'utilisateur fournit la définition du terme.	1. Le système demande à l'utilisateur de définir le terme spécifié.  3. Le système vérifie la liste des informations introduites par l'utilisateur et ajoute à ces informations l'ensemble des exigences dans lesquelles ce terme apparaît.

#### Cas alternatif : Terme déjà défini

##### Pré condition :

Le terme que l'utilisateur souhaite définir est déjà défini au sein du glossaire.

##### Post condition :

Le terme que l'utilisateur souhaitait définir a été redéfini.

##### Flux alternatif :

<i>Utilisateur</i>	<i>Système</i>
2. L'utilisateur effectue des modifications sur la définition et/ou l'orthographe du terme.  3. L'utilisateur justifie la modification qu'il vient de réaliser à la définition du terme.	1. Le système fournit à l'utilisateur la définition du terme que l'utilisateur souhaite redéfinir.  4. Le système demande à l'utilisateur de vérifier la cohérence entre la nouvelle définition du terme et la liste des exigences où ce terme apparaît.

(Retour au point 3 du Cas d'utilisation normal)

## A.2.2 Cas d'utilisation 2 : Définir un terme à partir d'une exigence

**Brève Description :** L'utilisateur souhaite définir un terme au sein du glossaire directement à partir d'un terme utilisé dans une exigence.

**Pré condition globale :** /

**Post condition globale :** /

Cas Normal
------------

**Pré conditions :**

L'utilisateur travaille sur une exigence

et le terme que souhaite définir l'utilisateur apparaît dans cette exigence.

**Post conditions :**

Le terme a été défini au sein de cette exigence

et il apparaît de manière claire dans l'exigence que ce terme a été défini.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur marque son intention de définir un terme employé au sein de l'exigence sur laquelle il travaille.	
2. (<<Include>> Saisie d'informations sur un terme)	
	3. Le système signale à l'utilisateur que le terme a été défini au sein du glossaire.
	4. Le système retourne à l'exigence sur laquelle l'utilisateur travaille.

(Fin du Cas d'utilisation)

*Remarque :* L'utilisateur ne peut définir un terme qui apparaît dans une liste de synonymes directement à partir d'une exigence car l'utilisateur ne peut déjà pas employer ce terme au sein d'une exigence (voir *Cas d'utilisation alternatif "synonyme" du Cas d'utilisation "Utiliser un terme au sein d'une exigence"*).

### A.2.3 Cas d'utilisation 3 : Définir un terme à partir du glossaire

**Brève Description :** L'utilisateur souhaite définir directement un terme à partir du glossaire.

**Pré conditions globale :** /

**Post conditions globale :** /

Cas Normal
------------

**Pré condition :**

Le terme à définir n'est pas présent au sein d'une liste de synonymes.

**Post condition :**

Le nouveau terme a été défini au sein du glossaire.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur marque son intention de définir un nouveau terme.	
2. L'utilisateur fournit le terme qu'il souhaite définir.	
3.(<<Include>>Saisie d'informations sur un terme)	
	4. Le système signale à l'utilisateur que le terme a été défini au sein du glossaire.

(Fin du Cas d'utilisation)

**Cas alternatif : Terme présent dans une liste de synonymes****Pré condition :**

Le terme à définir se trouve dans une liste de synonymes.

**Post condition :**

Le terme n'est pas défini au sein du glossaire.

**Flux alternatif :**

<i>Utilisateur</i>	<i>Système</i>
<p>1. (<i>Idem point 2 du Cas d'utilisation normal</i>)</p>	<p>2. Le système signale à l'utilisateur que ce terme ne peut être défini au sein du glossaire car il existe un synonyme (de sens, de buts, etc.) déjà défini.</p> <p>3. Le système renvoi l'utilisateur au terme synonyme du terme précédemment fournit par l'utilisateur.</p>

*(Retour au point 4 du Cas d'utilisation normal)*

#### A.2.4 Cas d'utilisation 4 : Définir/Modifier le niveau d'ambiguïté

**Brève Description :** Le commentateur souhaite définir le niveau ambiguïté de la définition d'un terme selon une échelle de mesure préétablie.

**Pré conditions globale :** /

**Post conditions globale :** /

Cas Normal
------------

**Pré conditions :**

Il y a au moins un terme défini au sein du glossaire.

**Post condition :**

Le niveau ambiguïté a été défini/modifié.

**Flux Normal :**

<i>Commentateur</i>	<i>Système</i>
1. Le commentateur sélectionne le terme dont il souhaite modifier le niveau ambiguïté	
2. Le commentateur marque son intention de modifier le niveau ambiguïté du terme.	
	3. Le système lui demande de définir le niveau ambiguïté par le biais d'une échelle de mesure non numérique et de justifier cette mesure.
4. Le commentateur fournit sa mesure et sa justification au système.	
	5. Le système a modifié le niveau ambiguïté du terme et un nouvel historique de la définition a été créé.

(Fin du Cas d'utilisation)

**A.2.5 Cas d'utilisation 5 : Utiliser un terme au sein d'une exigence**

**Brève Description :** L'utilisateur souhaite employer un terme au sein d'une exigence.

**Pré conditions globale :**

Il y a au moins un terme défini au sein du glossaire  
et l'utilisateur est en train de définir (ou modifier) une exigence.

**Post conditions globale :** /

**Cas Normal**

**Pré conditions :**

Aucune

**Post condition :**

L'utilisateur a employé un terme défini au sein du glossaire dans une exigence.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur utilise un terme défini au sein du glossaire dans une exigence.	2. Le système fait un lien entre le terme au sein du glossaire et l'exigence.

*(Fin du Cas d'utilisation)*

**Cas alternatif : Synonyme**

**Pré conditions :**

L'utilisateur emploi un terme se trouvant dans une liste de synonymes au sein d'une exigence.

**Post condition :**

L'utilisateur a changé de terme pour établir (ou modifier) l'exigence.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur emploi dans une exigence un terme présent dans une liste de synonymes.	
	2. Le système informe l'utilisateur que ce terme est présent au sein d'une liste de synonymes.
	3. Le système propose à l'utilisateur un autre terme de sens identique déjà défini dans le glossaire.
4. L'utilisateur change de terme pour établir/modifier l'exigence.	

*(Retour au point 2 du Cas d'utilisation normal)*

**Cas alternatif : Intrant/Extrant****Pré conditions :**

Le terme défini utilisé est un intrant et/ou un extrant.

**Post condition :**

Un lien de type intrant/extrant a été établi entre le terme défini et l'exigence.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. <i>(Idem point 1 du Cas d'utilisation normal).</i>	
2. L'utilisateur définit le terme comme intrant/extrant et peut l'attacher aux exigences correspondantes.	
	3. Le système fait un lien entre le terme au sein du glossaire et l'exigence.
	4. Le système note que ce terme constitue un intrant ou un extrant de cet exigence.

*(Fin du Cas d'utilisation)*



### A.2.6 Cas d'utilisation 6 : Consulter la définition d'un terme

**Brève Description :** Le visiteur cherche des informations concernant un terme dans le glossaire.

**Pré conditions globale :**

Le visiteur souhaite obtenir des informations concernant un terme.

**Post conditions globale :** /

<b>Cas Normal</b>
-------------------

**Pré condition :**

Le terme recherché est présent au sein du glossaire.

**Post condition :**

Le visiteur a obtenu les informations concernant un terme recherché.

**Flux Normal :**

<i>Visiteur</i>	<i>Système</i>
1. Le visiteur signale son intention de consulter un terme au sein du glossaire (soit directement à partir d'une exigence, soit à partir du glossaire).  2. Le visiteur fournit au système le terme qu'il recherche.	3. Le système fournit au visiteur la définition du terme recherché.

*(Fin du Cas d'utilisation)*

**Cas alternatif : Terme non défini****Pré condition :**

Le terme recherché n'est pas présent au sein du glossaire.

**Post condition :**

L'utilisateur n'a pas obtenu les informations concernant le terme recherché.

**Flux alternatif :**

<i>Visiteur</i>	<i>Système</i>
1. ( <i>Idem point 2 du Cas d'utilisation normal</i> )	2. Le système signale au visiteur que le terme recherché n'est pas défini au sein du glossaire.

(*Fin du Cas d'utilisation*)

### A.2.7 Cas d'utilisation 7 : Rechercher un terme

**Brève Description :** Le visiteur souhaite rechercher un terme selon certains critères au sein du glossaire.

**Pré conditions globale :** /

**Post conditions globale :** /

Cas Normal
------------

**Pré condition :**

Le visiteur possède au moins un critère pour pouvoir effectuer la recherche d'un terme.

**Post condition :**

Le visiteur a en sa possession une liste de termes définis au sein du glossaire répondant aux critères qu'il avait établis.

**Flux Normal :**

<i>Visiteur</i>	<i>Système</i>
1. Le visiteur marque son intention de rechercher un terme au sein du glossaire.  2. Le visiteur fournit au système un ensemble de critères pour la recherche d'un terme.	3. Le système recherche un terme au sein du glossaire correspondant aux critères établis par l'utilisateur.  4. Le système fournit à l'utilisateur une liste (qui peut être vide) des termes définis au sein du glossaire correspondant aux critères établis.

*(Fin du Cas d'utilisation)*

### A.2.8 Cas d'utilisation 8 : Supprimer la définition d'un terme

**Brève Description :** L'utilisateur souhaite supprimer la définition d'un terme défini au sein du glossaire.

**Pré conditions globale :** /

**Post conditions globale :** /

Cas Normal
------------

**Pré condition :**

Il y a au moins un terme défini au sein du glossaire.

**Post condition :**

Un terme défini au sein du glossaire a été supprimé  
et l'historique de ce terme est conservé.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur marque son intention de supprimer un terme défini au sein du glossaire.  2. L'utilisateur sélectionne le terme à supprimer.	3. Le système supprime le terme défini du glossaire et le système conserve un historique du terme supprimé.

*(Fin du Cas d'utilisation)*

### A.2.9 Cas d'utilisation 9 : Accéder aux définitions supprimées

**Brève Description :** L'administrateur souhaite consulter les définitions des termes du glossaire qui ont été supprimés.

**Pré conditions globale :** /

**Post conditions globale :** /

Cas Normal
------------

**Pré condition :**

Il y a au moins un terme défini au sein du glossaire qui a été supprimé.

**Post condition :**

Les définitions des termes supprimés ont été accédées par l'administrateur.

**Flux Normal :**

<i>Administrateur</i>	<i>Système</i>
1. L'administrateur marque son intention de consulter les termes qui ont été supprimés au sein du glossaire.	
	2. Le système fournit un accès aux termes supprimés du glossaire à l'administrateur.
3. L'administrateur peut consulter les termes supprimés au sein du glossaire.	

*(Fin du Cas d'utilisation)*

### A.2.10 Cas d'utilisation 10 : Générer un document à partir du glossaire

**Brève Description :** Le visiteur veut réaliser un document de type *Microsoft Word* spécifique au glossaire venant compléter le cahier de charge.

**Pré conditions globale :** /

**Post conditions globale :** /

Cas Normal
------------

**Pré condition :** /

**Post condition :**

Un document de type *Microsoft Word* spécifique au glossaire a été réalisé.

**Flux Normal :**

<i>Visiteur</i>	<i>Système</i>
1. Le visiteur marque son intention de vouloir générer un document de type <i>Microsoft Word</i> spécifique au glossaire.	2. Le système génère un document de type <i>Microsoft Word</i> en fonction de l'ensemble des informations comprises au sein du glossaire et des propriétés de tri.  3. L'utilisateur dispose du document généré.

(Fin du Cas d'utilisation)

### A.2.11 Cas d'utilisation 11 : Importer un glossaire

**Brève Description :** L'utilisateur souhaite utiliser, au sein d'un projet, un glossaire émanant d'un projet différent. Pour cela, il doit importer un glossaire venant de ce projet différent au sein du glossaire actuel.

**Pré conditions globale :**

Un glossaire est défini au sein d'un autre projet.

**Post conditions globale :** /

<b>Cas Normal</b>
-------------------

**Pré condition :**

Il n'existe pas 2 termes identiques ou 2 synonymes entre les 2 glossaires.

**Post condition :**

Le glossaire de l'autre projet a été ajouté au glossaire du projet actuel.

**Flux Normal :**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur marque son intention d'importer un glossaire venant d'un autre projet au sein du projet actuel.	
	2. Le système demande à l'utilisateur de lui fournir un glossaire venant d'un autre projet.
3. L'utilisateur fournit au système un glossaire venant d'un autre projet.	
	4. Le système vérifie qu'il n'y a pas 2 termes identiques ou de synonymes entre les 2 glossaires.
	5. Le système ajoute les informations du glossaire venant d'un autre projet au projet actuel.

*(Fin du Cas d'utilisation)*

**Cas alternatif : Termes identiques ou synonymes****Pré condition :**

Il existe au minimum 2 termes identiques ou 2 synonymes entre les 2 glossaires.

**Post conditions :**

Le glossaire de l'autre projet a été ajouté au projet actuel  
et il n'y a pas 2 termes identiques ni 2 synonymes.

**Flux alternatif :**

<i>Utilisateur</i>	<i>Système</i>
<p>3. L'utilisateur fait un choix entre les versions des différents projets. <i>(Retour au point 5 du Cas d'utilisation normal)</i></p>	<p>1. <i>(Idem point 4 du Cas d'utilisation normal)</i></p> <p>2. Le système propose à l'utilisateur de choisir (comparaison) la version venant de l'autre projet ou du projet actuel pour chaque termes identiques ou synonymes.</p>



### A.2.12 Cas d'utilisation 12 : Consulter l'historique des versions de la définition d'un terme (optionnel)

**Brève Description :** Le visiteur souhaite consulter les anciennes versions de la définition d'un terme au sein du glossaire.

**Pré conditions globale :**

Il y a au moins un terme au sein du glossaire dont la définition a été modifiée au moins une fois.

**Post conditions globale :** /

Cas Normal
------------

**Pré condition :** /

**Post condition :**

Les versions précédentes de la définition d'un terme ont été consultées.

**Flux Normal :**

<i>Visiteur</i>	<i>Système</i>
1. Le visiteur marque son intention de consulter les versions précédentes de la définition d'un terme.	
	2. Le système fournit au visiteur la liste des versions existantes (l'historique) émanant des modifications successives (avec leurs justifications) de la définition d'un terme.
3. Le visiteur sélectionne dans la liste une version qu'il souhaite consulter.	
	4. Le système affiche la version sélectionnée à l'utilisateur.
5. L'utilisateur marque son intention de consulter une autre version (retour au point 3) ou de quitter l'historique.	

*(Fin du Cas d'utilisation)*

<b>Cas alternatif : Versions de l'historique filtrées</b>
---

**Pré condition :** Idem cas normal

**Post condition :**

Les versions historiques de la définition d'un terme ont été filtrées selon certains critères.

**Flux Normal :**

<i>Visiteur</i>	<i>Système</i>
	1. ( <i>Idem point 2 du Cas d'utilisation normal</i> )
2. Le visiteur établit les critères de filtrage à appliquer à l'historique.	
	3. Le système affiche au visiteur la liste des historiques filtrés selon les critères établis par l'utilisateur.
<i>(Retour au point 3 du Cas d'utilisation normal)</i>	

**A.2.13 Cas d'utilisation 13 : Exporter un glossaire (optionnel)**

**Brève Description :** L'utilisateur souhaite réutiliser, au sein d'un autre projet, le glossaire défini au sein du projet actuel. Pour cela, il doit exporter le glossaire actuel.

**Pré conditions globale :** /

**Post conditions globale :** /

**Cas Normal****Pré condition :**

Il y a au moins un terme défini au sein du glossaire.

**Post condition :**

Le glossaire du projet actuel a été exporté.

**Flux Normal :**

<i>Visiteur</i>	<i>Système</i>
1. Le visiteur marque son intention d'exporter le glossaire venant du projet actuel.  3. L'utilisateur possède le glossaire qui est indépendant du projet <i>Gen.Spec</i> où il avait été établi.	2. Le système exporte le glossaire du projet actuel.

*(Fin du Cas d'utilisation)*

**Cas alternatif : Aucun termes définis****Pré condition :**

Aucun terme n'est défini au sein du glossaire.

**Post conditions :**

Le glossaire du projet n'a pas été exporté.

**Flux alternatif :**

<i>Utilisateur</i>	<i>Système</i>
	<ol style="list-style-type: none"><li>1. (<i>Idem point 1 du Cas d'utilisation normal</i>)</li><li>2. Le système signale à l'utilisateur qu'il n'y a aucun termes définis au sein du glossaire du projet actuel.</li><li>3. Le système signale à l'utilisateur que le glossaire du projet n'est pas exporté.</li></ol>

*(Fin du Cas d'utilisation)*

### A.3 Lexique

- \* **Exigences dans lesquelles le terme apparaît** : Correspond à la liste des exigences dans lesquelles le terme, employé ici, est utilisé.
- \* **Générer un document de type *Microsoft Word*** : Réaliser un document de type *Microsoft Word* à partir des informations comprises au sein du glossaire.
- \* **Identifiant de version** : Correspond à l'élément permettant de distinguer les différentes versions de l'ensemble des informations d'un terme.
- \* **Termes identiques** : termes s'écrivant de la même manière.
- \* **Propriétés de tri** : Propriétés établies par le visiteur pour la réalisation du document de type *Microsoft Word*. Ces propriétés définissent l'ordre d'apparition des définitions au sein du document.
- \* **Sens identiques** : Correspond à 2 termes s'écrivant de manière différente mais ayant le même sens. Au moins un des 2 termes apparaît dans la liste de synonymes de l'autre.

## **Annexe B**

# **Cas d'utilisation : GMDE et AECF**

La dernière modification du contenu de ce document date du 8 novembre 2006.

*Modèle d'objectifs et cas d'utilisation*

## **Gestion multi-documents**

### **GenSpec**

**NOVEMBRE 2006**

## 1. INTRODUCTION

Cette annexe s'intéresse à la réalisation d'une gestion multi-documents pour *GenSpec*. Il a pour but de présenter le modèle d'objectifs ainsi que les cas d'utilisation nécessaires à une telle gestion.

Il est composé de deux sections :

- Premièrement, un modèle d'objectifs qui met en évidence la motivation des choix d'analyse et de conception, les aspects intentionnels des acteurs du projet ainsi que les aspects non-fonctionnels. ce modèle d'objectifs est réalisé dans le formalisme *i\**.
- Deuxièmement, les cas d'utilisation et les scénarii qui y sont associés. Ces derniers représentent le comportement affiché par le système sous certaines conditions de manière à satisfaire un objectif de l'un des acteurs.

Notons également que l'ensemble des cas d'utilisation répondent à des buts que l'on retrouve dans le modèle d'objectifs.

## 2. MODÈLE D'OBJECTIFS

Cette section présente d'une part, le diagramme *i\** qui modélise les relations entre les différents objectifs des acteurs du système, d'autre part, un glossaire qui couvre la terminologie de certains termes présents dans le diagramme et qui pourraient paraître ambiguës au lecteur.

### 2.1. Diagramme *i\**

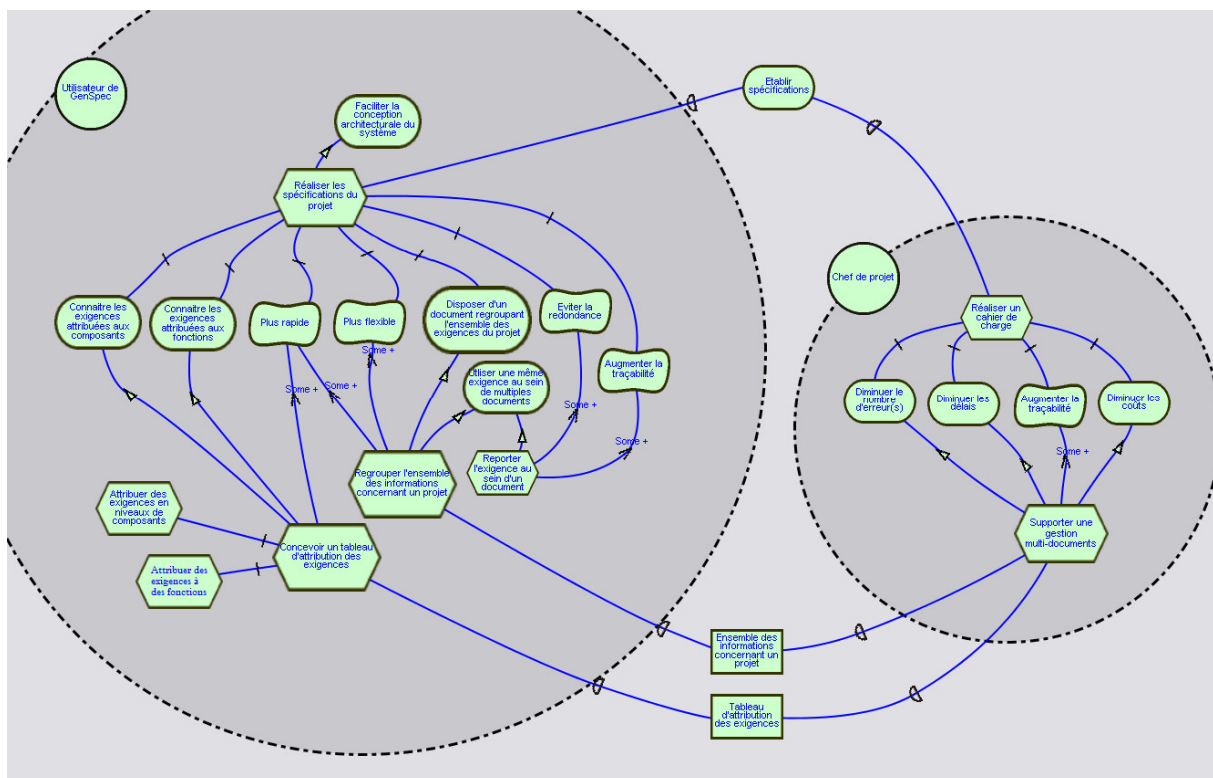


Figure 1 - Modèle d'objectifs

### 2.2. Glossaire

Nous retrouvons dans cette sous-section les termes utilisés dans le modèle d'objectifs qui pourraient se révéler ambigus pour le lecteur.



Ces termes sont les suivants :

- **Spécifications** : Les spécifications correspondent à l'ensemble des exigences d'un projet réalisé par l'utilisation de *GenSpec*.
- **Cahier de charge** : Le cahier de charge ne se limite pas à l'ensemble des spécifications. Selon la terminologie employée chez *Hydro-Québec*, la notion de cahier de charge réfère à un élément plus général comportant, entre autre, les spécifications.
- **Tableau d'attribution des exigences aux composants** : Tableau contenant une hiérarchie de composants et une liste de fonctions. Des exigences peuvent être assignées à ces composants ou fonctions.
- **Gestion multi-documents** : Gestion du projet permettant de générer différents documents en fonction des regroupements d'exigences réalisés dans *GenSpec*.

### **3. LES CAS D'UTILISATION**

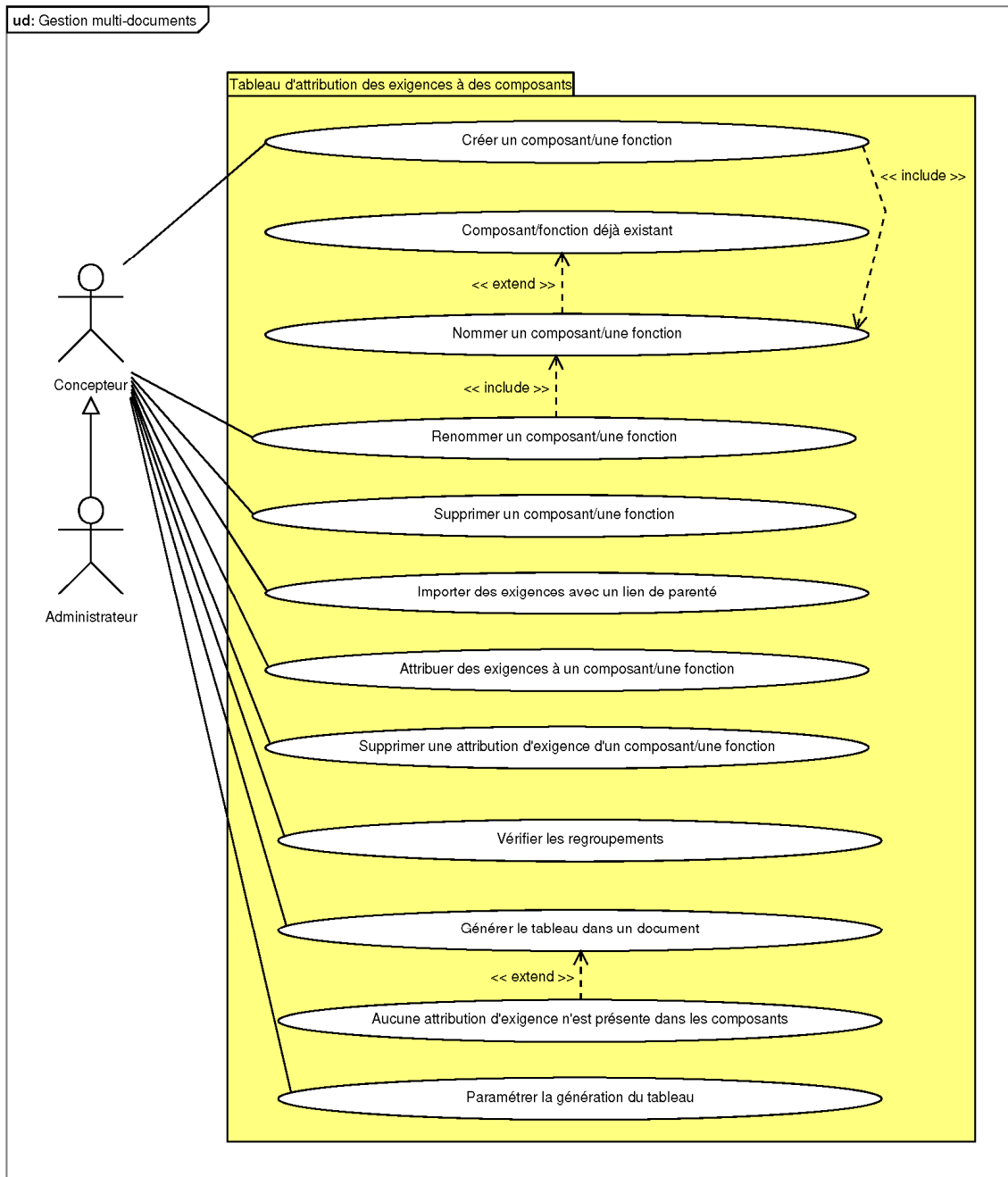
Au sein de cette section, vous retrouverez les diagrammes des *Cas d'utilisation* ainsi que l'ensemble des scénarii associés.

#### **3.1. Diagrammes de cas d'utilisation**

Les cas d'utilisation sont représentés dans 2 parties distinctes, et ce afin de différencier les différents aspects relatifs à la gestion multi-documents. Nous retrouvons les deux sous-systèmes suivants :

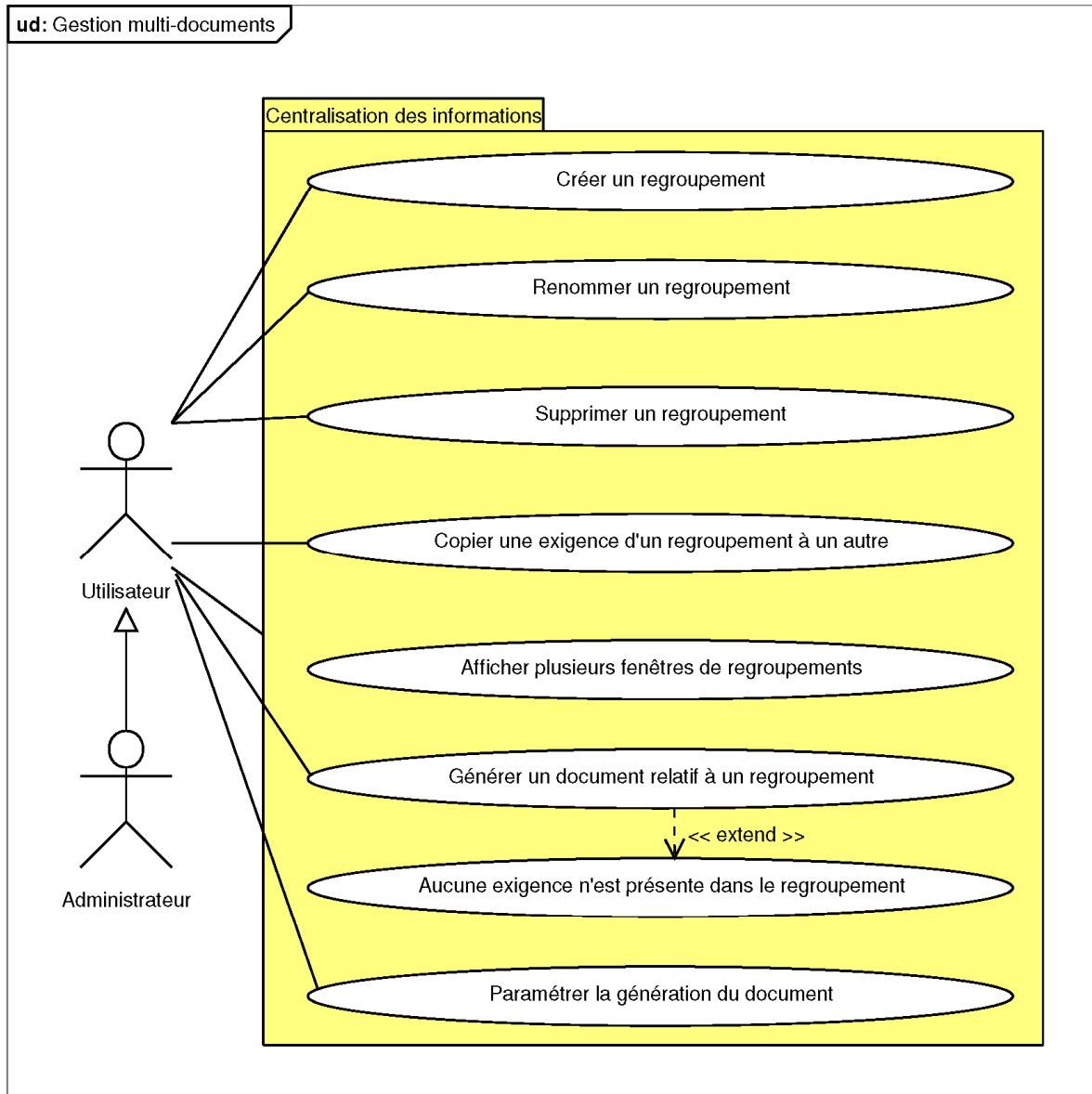
- Le tableau d'attribution des exigences à des composants ;
- La centralisation des informations.

**3.1.1. Tableau d'attribution des exigences à des composants**



**Figure 2 - Diagramme des cas d'utilisation pour l'attribution des exigences à des composants**

**3.1.2. Centralisation des informations**



**Figure 3 - Diagramme des cas d'utilisation pour la centralisation des informations**

### 3.2. Classes d'utilisateurs

Les privilèges associés à chacun de ces acteurs pour *GenSpec* sont les suivants :

- **Administrateur** : Il a accès à toutes les fonctionnalités de *GenSpec*.
- **Utilisateur** : Les fonctionnalités suivantes ne sont pas accessibles aux usagers ayant le niveau d'accès Utilisateur :
  - Génération de versions officielles ;
  - Ajout d'un usager ;
  - Retrait d'un usager ;
  - Modification du profil d'un autre usager ;
  - Changement du niveau d'accès ;
  - Attribution des exigences à des composants.
- **Concepteur** : Un concepteur peut uniquement ajouter des commentaires aux exigences et gérer les attributions d'exigences à des composants.
- **Visiteur** : Les usagers de type Visiteur ont seulement des droits de lecture de tous les champs et des droits de génération des documents. Ils ne peuvent donc pas ajouter, déplacer, modifier ou supprimer des exigences. Les fonctionnalités qui sont inaccessibles pour l'Utilisateur le sont aussi pour le Visiteur.
- **Testeur** : Un Testeur a les mêmes restrictions qu'un Visiteur excepté qu'il a les droits d'écriture sur les champs Commentaires et les champs de l'onglet Essais.
- **Commentateur** : Un Commentateur a les mêmes restrictions qu'un Visiteur excepté qu'il a les droits d'écriture sur les champs Commentaires.

Pour en savoir davantage sur les différents types d'utilisateurs, se référer au manuel d'aide de *GenSpec*.

Dans la suite de cette section, chacune de ces parties sera analysée séparément et sera accompagnée de son lexique respectif. Ce dernier ayant pour objectif de faire référence aux termes utilisés dans les cas d'utilisation.

### 3.3. Canevas des cas d'utilisation

Comme présenté ci-après, les cas d'utilisation respectent le formalisme suivant :

- **Le nom du cas d'utilisation** tel que représenté dans le diagramme.
- **Une brève description** qui décrit le cas d'utilisation.
- **La séquence des étapes** apparaissant dans le flux normal ou alternatif.
- **La pré-condition globale** qui décrit les conditions qui doivent être vérifiées pour que le cas d'utilisation puisse se dérouler. Elle porte d'une part sur le cas normal et d'autre part, sur le cas alternatif.
- **La post-condition globale** qui décrit les conditions toujours vérifiées à la fin du cas d'utilisation. Elle porte d'une part sur le cas normal et d'autre part, sur le cas alternatif.
- **La pré-condition spécifique** décrit les conditions préalables qui doivent être vérifiées pour que le flux – soit normal, soit alternatif – puisse se dérouler.
- **La post-condition spécifique** décrit les conditions qui sont vérifiées après que le flux – soit normal, soit alternatif – se soit déroulé.
- **Le flux du cas normal** présente la séquence d'interactions du cas normal.

- **Le flux du cas alternatif** présente la séquence d'interactions du cas alternatif.

### 3.4. Lexiques relatifs aux cas d'utilisation

#### 3.4.1. Lexique des cas d'utilisation pour le tableau d'attribution des exigences à des composants

Le lexique ci-dessous donne le sens des termes utilisés dans les cas d'utilisation ci-dessus.

- **Regroupement d'exigences** : Rassemblement d'exigences dans un même composant ou fonction.
- **Composant** : Module regroupant des exigences en son sein. Ces dernières permettent de montrer les exigences qui doivent être rencontrées pour satisfaire la réalisation du composant.
- **Fonction** : Fonction associée à une exigence mettant en évidence quelles sont les fonctions des exigences dans le système.
- **Sauvegarder** : Enregistre les modifications effectuées de manière permanente dans la base de données du projet.
- **Tableau** : Tableau représentant les attributions des exigences à des composants et fonctions du projet.
- **Générer** : Représenter les données dans un document Microsoft Word ou Excel selon un schéma prédéfini.
- **Composant/fonction vide** : Composant ou fonction n'incluant aucune exigence.
- **Avertissement** : Problème qui nécessite l'attention du concepteur afin de vérifier s'il s'agit d'une incohérence.
- **Lien entre des exigences** : Relations de parenté entre des exigences.

#### 3.4.2. Lexique des cas d'utilisation pour le regroupement d'exigences

Le lexique ci-dessous donne le sens des termes utilisés dans les cas d'utilisation ci-dessus.

- **Regroupement** : Regroupement d'exigences sous forme d'arbre hiérarchique.
- **Fenêtre de regroupements** : Fenêtre contenant l'ensemble des regroupements.
- **Arbre global** : Regroupement contenant l'ensemble des exigences d'un projet sous forme d'arbre hiérarchique. Ce regroupement ne peut être supprimé.

### 3.5. Scénarii des Cas d'utilisation pour le tableau d'attribution des exigences aux composants

Dans cette section, nous utiliserons le terme *Concepteur* pour mettre en évidence le type d'utilisateur possédant le droit de réaliser les scénarii des cas d'utilisation. Or, notons que l'administrateur possède au moins tous les droits du concepteur, et par conséquent est autorisé également à réaliser les scénarii décrits ci-dessous.

### 3.5.1. Nommer un composant/une fonction

Brève Description : Attribue un nom à un composant/une fonction.

**Flux normal:**

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur nomme le composant/la fonction.	2. Le système vérifie si le nom existe déjà dans le niveau sélectionné. 3. Le système accepte le nom donné par le concepteur.

*(Fin du Use Case)*

Cas alternatif : Composant/fonction déjà existant

Brève Description : Le concepteur souhaite nommer un composant/une fonction avec un nom qui n'est pas unique dans le niveau de composants ou de fonctions.

Pré condition : Il existe un composant/une fonction de même nom.

Post condition : La modification du nom du composant/de la fonction a échoué

et le système propose à nouveau d'introduire un nom pour le composant/la fonction.

**Flux alternatif :**

<i>Concepteur</i>	<i>Système</i>
1. ( <i>Idem jusqu'au point 2 du cas d'utilisation normal.</i> )	2. Le système signale qu'un composant/une fonction possède déjà ce nom. 3. Le système n'enregistre pas la modification du nom du composant/de la fonction.

(Fin du Use Case)



### 3.5.2. Créer un composant/une fonction

Brève Description : L'utilisateur crée un composant/une fonction.

Cas Normal

Pré condition : /

Post condition : Le composant/ la fonction est créée

et le composant/la fonction ne possède aucune exigence.

**Flux Normal:**

<i>Concepteur</i>	<i>Système</i>
<p>1. Le concepteur sélectionne l'espace des composants ou des fonctions.</p> <p>3. Le concepteur signale son intention de créer un nouveau composant/une nouvelle fonction.</p>	<p>2. Le système affiche l'espace des composants ou des fonctions.</p>
<p>4. ( includes ) Nommer un composant/une fonction.</p>	<p>5. Le système crée le nouveau composant/la nouvelle fonction respectivement dans l'espace des composants/des fonctions.</p>
<p>6. Le concepteur dispose d'un nouveau composant/une nouvelle fonction.</p>	

*(Fin du Use Case)*

### 3.5.3. Générer le tableau dans un document

Brève Description : Permet de générer le tableau dans un format prédéfini. Le format concerne le type de document de sortie (Microsoft Word ou Excel).

Pré condition globale : /

Post condition globale : /

Cas Normal

Pré condition : Au moins une exigence est assignée à au moins un composant.

Post condition : Le tableau est généré dans un nouveau document selon le format prédéfini.

#### **Flux Normal:**

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur marque son intention de générer le tableau.	
	2. Le système vérifie qu'il existe au moins une exigence associée à un composant.
	3. Le système génère le document sur base des regroupements effectués.
4. Le concepteur possède le document avec le tableau.	

(Fin du Use Case)

Cas alternatif : Aucune attribution d'exigence n'est présente dans les composants

Brève Description : Le concepteur souhaite générer le tableau sans qu'aucun regroupement ne soit effectué ou aucune exigence assignée à un composant.

Pré condition : Tous les composants sont vides.

Post condition : Aucun document n'est généré.

**Flux alternatif :**

<i>Concepteur</i>	<i>Système</i>
	<ol style="list-style-type: none"><li>1. (<i>Idem jusqu'au point 2 du cas d'utilisation normal.</i>)</li><li>2. Le système signale que tous les composants sont vides.</li><li>3. Le système ne génère aucun document.</li></ol>

*(Fin du Use Case)*

### 3.5.4. Renommer un composant/une fonction

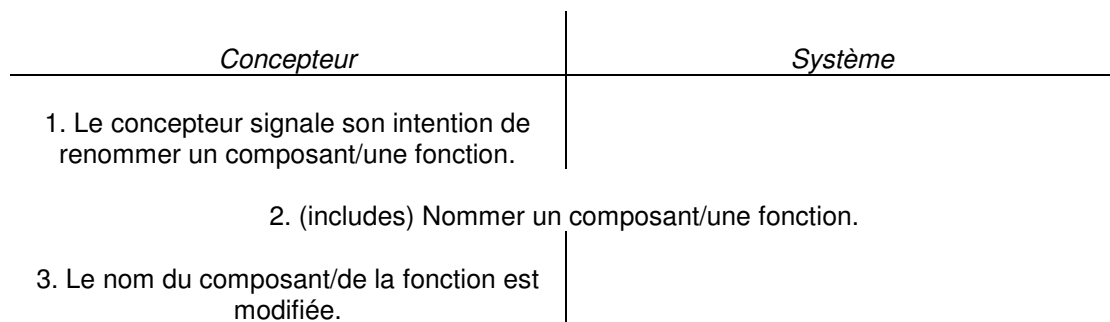
Brève Description : L'utilisateur renomme un composant/une fonction.

Cas Normal

Pré condition : Le concepteur à la possibilité d'accéder au composant/à la fonction qu'il souhaite renommer.

Post condition : Le composant/la fonction est renommée.

#### Flux Normal:



*(Fin du Use Case)*

### 3.5.5. Vérifier les regroupements

Brève Description : Le concepteur demande une vérification des regroupements d'exigences pour les composants et les fonctions.

Cas Normal

Pré condition : \

Post condition : Un rapport indique si des avertissements ont été détectés lors des regroupements

et dans le cas où des avertissements ont été découverts, le rapport indique quelles sont les sources des avertissements

#### Flux Normal:

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur signale son intention de vérifier les regroupements.	
	2. Le système effectue une analyse de tous les regroupements.
	3. Le système regroupe les composants et/ou les fonctions qui sont à l'origine d'un problème.
4. Le concepteur est averti du résultat de la vérification et peut atteindre les composants/fonctions qui sont à l'origine de l'avertissement.	

(Fin du Use Case)

### 3.5.6. Supprimer un composant/une fonction

Brève Description : Le concepteur supprime un composant ou une fonction existante.

Cas Normal

Pré condition : Il existe au moins un composant ou une fonction à supprimer.

Post condition : Le composant ou la fonction est supprimée de la hiérarchie des composants ou de la liste des fonctions.

#### **Flux Normal:**

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur signale son intention de supprimer un composant ou une fonction.	2. Le système supprime le composant ou la fonction correspondante.

*(Fin du Use Case)*

### 3.5.7. Paramétrer la génération du tableau

Brève Description : Le concepteur paramètre les différentes options de génération du tableau.

Cas Normal

Pré condition : \

Post condition : Les différentes options sont ajustées pour que le concepteur génère le tableau comme il le souhaite.

#### Flux Normal:

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur signale son intention de paramétrer les options de génération du tableau.	
	2. Le système propose toutes les options disponibles au concepteur.
3. Le concepteur effectue les choix adaptés à ses besoins.	
	4. Le système enregistre les options sélectionnées par le concepteur.

(Fin du Use Case)

### 3.5.8. Attribuer des exigences à un composant/une fonction

Brève Description : Le concepteur attribue des exigences à des composants et/ou des fonctions.

Cas Normal

Pré condition : Les exigences à attribuer sont disponibles dans un arbre hiérarchique de *GenSpec*.  
et les composants ou fonctions cibles existent.

Post condition : Les exigences sont attribuées à des composants ou des fonctions.

#### Flux Normal:

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur effectue les attributions d'exigences dans des composants ou fonctions.	
	2. Le système classe les exigences dans les composants et/ou les fonctions.
3. Le concepteur enregistre les attributions effectuées.	
	4. Le système enregistre les informations relatives aux attributions.

(Fin du Use Case)



### 3.5.9. Importer des exigences avec un lien de parenté

Brève Description : Le concepteur souhaite attribuer les parents et/ou enfants d'une exigence au sein d'un composant ou d'une fonction lors de l'attribution de l'exigence en question.

Cas Normal

Pré condition : Il existe au moins une exigence à attribuer.

Post condition : Les exigences liées sont également attribuées à un composant/une fonction, selon le(s) lien(s) sélectionné(s) par le concepteur.

**Flux Normal:**

<i>Concepteur</i>	<i>Système</i>
1. Le concepteur attribue une exigence à un composant/une fonction.	
	2. Le système propose au concepteur d'attribuer également les exigences liées à la première.
3. Le concepteur sélectionne les liens qui s'appliquent pour l'attribution des exigences liées.	
	4. Le système attribue les exigences liées selon le type de lien sélectionné par le concepteur.

(Fin du Use Case)

### 3.5.10. Supprimer une attribution d'exigence d'un composant/une fonction

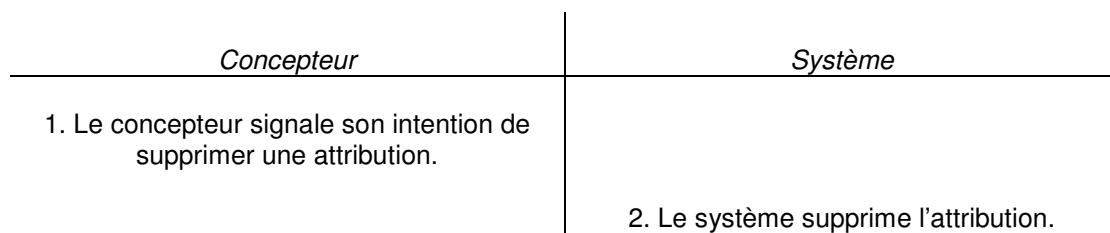
Brève Description : Le concepteur supprime une attribution vers une exigence.

Cas Normal

Pré condition : Une attribution existe entre une exigence et un composant/une fonction.

Post condition : L'attribution entre l'exigence et le composant/la fonction n'existe plus.

#### **Flux Normal:**



*(Fin du Use Case)*

### **3.6. Scénarii des Cas d'utilisation pour la centralisation des informations**

Dans cette section, nous utiliserons le terme *Utilisateur* pour mettre en évidence le type d'utilisateur possédant le droit de réaliser les scénarii des cas d'utilisation. Or, notons que l'administrateur possède au moins tous les droits de l'utilisateur, et par conséquent est autorisé également à réaliser les scénarii décrits ci-dessous.

### 3.6.1. Nommer un regroupement

Brève Description : L'utilisateur donne un nom à un regroupement.

**Flux normal:**

<i>Concepteur</i>	<i>Système</i>
1. L'utilisateur nomme le regroupement.	2. Le système vérifie si le nom existe déjà dans un des regroupements. 3. Le système accepte le nom donné par l'utilisateur.

*(Fin du Use Case)*

Cas alternatif : Regroupement déjà existant

Brève Description : L'utilisateur souhaite nommer un regroupement de la même manière qu'un autre regroupement.

Pré condition : Il existe déjà un regroupement de même nom.

Post condition : La modification du nom du regroupement a échoué

et le système propose à nouveau d'introduire un nom pour le regroupement.

**Flux alternatif :**

<i>Concepteur</i>	<i>Système</i>
	<ol style="list-style-type: none"><li>1. (<i>Idem jusqu'au point 2 du cas d'utilisation normal.</i>)</li><li>2. Le système signale qu'un regroupement possède déjà ce nom.</li><li>3. Le système n'enregistre pas la modification du nom du regroupement</li></ol>

(Fin du Use Case)

### 3.6.2. Créer un regroupement

Brève Description : L'utilisateur crée un regroupement.

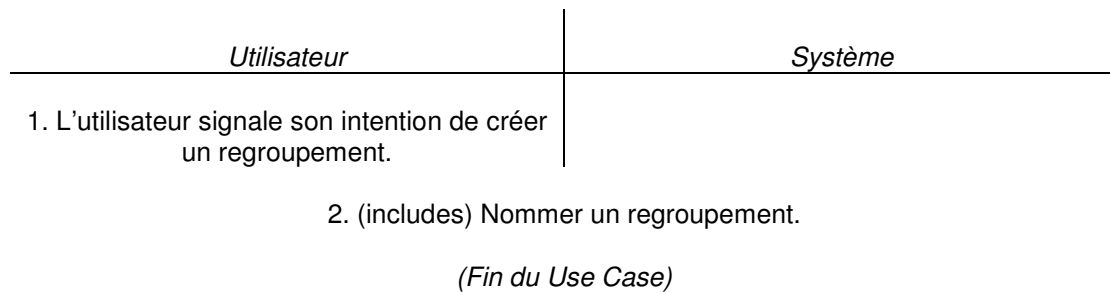
Cas Normal

Pré condition : \

Post condition : Le regroupement est créé.

et le regroupement ne possède aucune exigence.

#### **Flux Normal:**



### 3.6.3. Renommer un regroupement

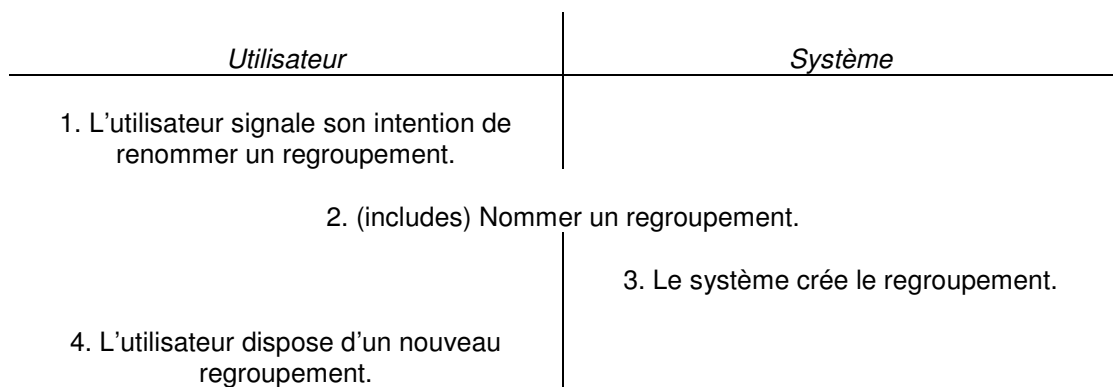
Brève Description : L'utilisateur renomme un regroupement.

Cas Normal

Pré condition : Il existe au moins un regroupement à renommer.

Post condition : Le regroupement est renommé.

#### Flux Normal:



(Fin du Use Case)

### 3.6.4. Afficher plusieurs fenêtres de regroupement

Brève Description : L'utilisateur peut afficher autant de fenêtres de regroupements qu'il le souhaite.

Cas Normal

Pré condition : \

Post condition : Une fenêtre de regroupements supplémentaire est affichée.

#### Flux Normal:

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur signale son intention d'afficher une fenêtre de regroupements.	
	2. Le système affiche une nouvelle fenêtre de regroupements.
3. L'utilisateur dispose d'une nouvelle fenêtre de regroupements.	

*(Fin du Use Case)*



### 3.6.5. Copier une exigence d'un regroupement à un autre

Brève Description : L'utilisateur coupe ou copie une exigence d'un regroupement à un autre.

Cas Normal

Pré condition : Il existe au moins une exigence à déplacer

Post condition : L'exigence est déplacée ou copiée dans le regroupement cible.

#### Flux Normal:

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur sélectionne une exigence à partir d'un regroupement et la déplace ou la copie vers un autre regroupement.	
	2. Le système déplace ou copie l'exigence vers le regroupement ciblé.
3. L'utilisateur dispose de l'exigence dans le regroupement cible.	

(Fin du Use Case)

### 3.6.6. Générer un document relatif à un regroupement

Brève Description : L'utilisateur génère un regroupement selon les différents formats de documents présents dans *GenSpec*.

Pré condition globale : Le regroupement à générer existe.

Post condition globale : /

Cas Normal

Pré condition : /

Post condition : Le regroupement est généré.

#### **Flux Normal:**

<i>Utilisateur</i>	<i>Système</i>
1. L'utilisateur marque son intention de générer un document relatif à un regroupement.	
	2. Le système vérifie que le regroupement contient des exigences.
	3. Le système génère le regroupement.
4. L'utilisateur dispose du regroupement généré.	

(Fin du Use Case)

Cas alternatif : Aucune exigence n'est présente dans le regroupement

Brève Description : Le concepteur souhaite générer le tableau sans qu'aucune exigence ne soit attribuée au regroupement.

Pré condition : /

Post condition : Aucun document n'est généré.

**Flux alternatif :**

<i>Utilisateur</i>	<i>Système</i>
1. ( <i>Idem jusqu'au point 2 du cas d'utilisation normal.</i> )	2. Le système avertit l'utilisateur qu'aucune exigence n'est présente dans le regroupement. 3. Le système ne génère aucun document.

*(Fin du Use Case)*

### 3.6.7. Supprimer un regroupement

Brève Description : L'utilisateur supprime un regroupement.

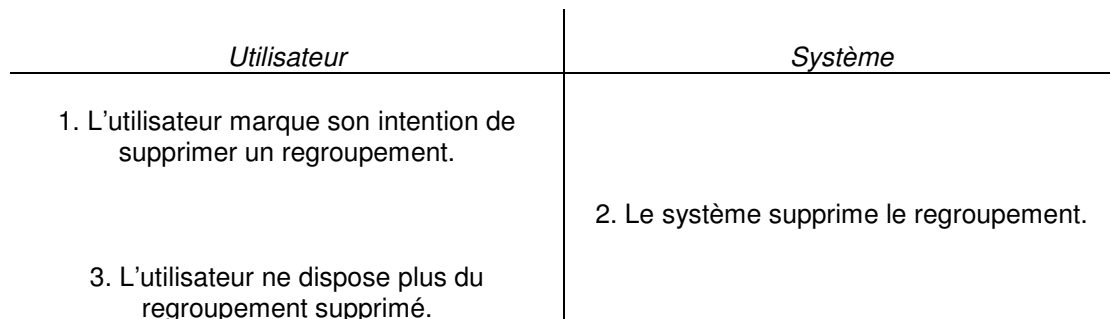
Cas Normal

Pré condition : Il existe au moins un regroupement à supprimer.

Post condition : Le regroupement est supprimé

et les exigences présentes dans le regroupement ne sont pas supprimées de l'arbre global.

#### **Flux Normal:**



*(Fin du Use Case)*

### 3.6.8. Paramétrer la génération

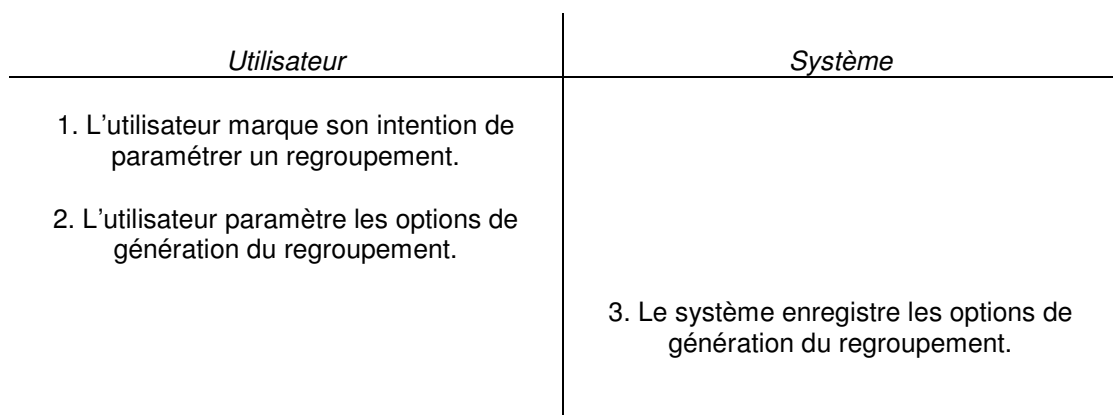
Brève Description : L'utilisateur paramètre les options de génération du regroupement.

Cas Normal

Pré condition : Il existe au moins un regroupement à générer.

Post condition : Les options de génération pour le regroupement pour le regroupement sont paramétrées.

#### Flux Normal:



(Fin du Use Case)

#### 4. CONCLUSION

Ce document a mis en évidence, par l'intermédiaire d'un diagramme d'objectifs, les besoins relatifs à une gestion multi-documents pour *GenSpec*. De plus, il a mis en évidence les différents cas d'utilisation se rapportant à une telle gestion.

Il constitue la première phase d'analyse des besoins. En effet, trois documents viendront le compléter. Nous retrouverons donc :

- une étude qui s'assimile au document de conception opérationnel (norme ISO 12207). Ce dernier mettra en évidence les besoins d'un développement d'une gestion multi-documents et les fonctionnalités apportées grâce au développement d'une telle gestion. En effet, il explicitera les besoins requis pour *GenSpec* et montrera la solution apportée pour répondre à ces besoins. Notons que le présent-document s'intégrera en annexe de cette étude.
- le document des spécifications d'exigences. Ce dernier sera produit par l'intermédiaire de *GenSpec* même, afin de conserver une représentation cohérente des exigences vis-à-vis des habitudes d'*Hydro-Québec*.
- un document de conception qui présentera différents diagrammes UML. Ce dernier contiendra, d'une part, les diagrammes de composants, de classes et de séquences, d'autre part, les schémas conceptuels entités-associations relatifs aux deux parties de la gestion multi-documents



## **Annexe C**

# **Etude : glossaire**

La dernière modification du contenu de ce document date du 26 octobre 2006.



*Étude*

**GenSpec**

**Glossaire**

**ETU10004**

**OCTOBRE 2006**

## 1. INTRODUCTION

### 1.1. Objet

Ce document présente une vue globale de l'outil *GenSpec* ainsi que l'ensemble des modifications qui vont lui être appliquées afin de répondre aux nouveaux souhaits des différents utilisateurs. Au travers de ce document sera reporté l'ensemble des informations permettant de définir un point de vue de haut niveau concernant l'ajout d'une nouvelle fonctionnalité au sein de *GenSpec* : Un glossaire.

Ce document s'adresse à tout utilisateur de *GenSpec*.

### 1.2. Portée

Le glossaire de *GenSpec* va permettre de réduire l'ambiguïté inhérente des exigences résultant de l'utilisation du langage naturel pour établir les spécifications. Ce glossaire va permettre également de définir les termes qui sont employés de manière courante au sein des exigences (intrants, extrants, etc.). Le glossaire ne se limite donc pas qu'à la définition d'un terme mais bien à un ensemble d'informations permettant une gestion et une compréhension plus aisée et plus rapide de l'ensemble des exigences.

### 1.3. Définitions, acronymes et abréviations

Aucuns objets.

### 1.4. Références

- [1] IEEE Std 1362-1998, IEEE Guide for information Technology – System Definition – Concept of Operations (ConOps) Document, 1998.
- [2] From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity A Handbook, Daniel M. Berry & Erik Kamsties, 2003.  
(<http://se.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf>)

### 1.5. Vue d'ensemble

Ce document se compose des 5 chapitres et 1 annexe suivante :

1. INTRODUCTION : Ce chapitre fournit une vue d'ensemble de toute la spécification. Il en décrit :
  - l'objet ;
  - la portée ;
  - les références ;
  - une vue d'ensemble.
  
2. SITUATION ACTUELLE : Ce chapitre fournit une description du système actuel. Il en décrit :
  - le contexte, objectif et étendue ;
  - les classes d'utilisateurs ;
  - la structure organisationnelle ;
  - le profil de chaque classe d'utilisateur ;
  - l'autre personnel impliqué.
  
3. SOLUTIONS PROPOSEES : Ce chapitre fournit une description des différentes alternatives de fonctionnalités que l'on peut associer à la solution. Il en décrit :
  - la justification et la nature des changements ;
  - les éléments de solution au sein de logiciels concurrents.
  
4. RECOMMANDATIONS : Ce chapitre fournit une description plus détaillée de la solution à établir. Il en décrit :
  - le contexte, objectif et étendue du nouveau système
  
5. ANALYSE DU SYSTÈME PROPOSÉ : Ce chapitre fournit une analyse du système proposé. Il en décrit :
  - le sommaire des améliorations ;
  - les désavantages et limites.

## 2. SITUATION ACTUELLE

Ce chapitre comporte une description du logiciel *GenSpec* tel qu'il existe actuellement ainsi qu'une description des problèmes rencontrés actuellement.

### 2.1. Contexte, objectifs et étendue du système actuel

Afin d'améliorer son processus d'ingénierie, l'entreprise québécoise Hydro-Québec a développé un logiciel permettant une meilleure gestion des exigences : *GenSpec*.

*GenSpec* est un logiciel permettant une gestion plus aisée et plus efficace de l'ensemble des spécifications d'un projet. Il permet de répondre à l'ensemble des problèmes souvent rencontrés au sein de l'ingénierie des exigences. Ce logiciel fournit la possibilité d'entrer des exigences au sein d'une base de données, d'effectuer un ensemble de vérifications sur celles-ci et de générer un document d'exigences. Le but essentiel de ce logiciel est d'arriver à une gestion des exigences qui se veut la plus adéquate et optimale.

L'ingénierie des exigences est une phase essentielle en ce qui concerne le développement d'un logiciel. Une bonne gestion des exigences est importante car elle constitue le fondement même de la réalisation du travail à effectuer.

### 2.2. Classes d'utilisateurs

On décrit ici les différentes classes d'utilisateurs selon les droits qui leur sont associés.

On peut retrouver actuellement 5 classes d'utilisateurs au sein de *GenSpec* :

- Administrateur
- Utilisateur
- Testeur
- Commentateur
- Visiteur

Chacune de ces classes d'utilisateurs présentes des caractéristiques propres. Ces différentes classes rendent compte des différentes responsabilités et niveaux de compétences de ses membres :

- **Administrateur** : Un administrateur possède tous les droits vis-à-vis d'un projet (il peut gérer les droits d'accès des différents usagers d'un projet, etc.), il a accès à toutes les fonctionnalités de *GenSpec*.
- **Utilisateur** : Un utilisateur correspond à administrateur excepté qu'il ne peut gérer les droits d'accès des différents usagers ni générer de versions officielles.
- **Visiteur** : Un visiteur ne possède qu'un droit de consultation et génération de documents à partir d'un projet.
- **Commentateur** : Un commentateur correspond à un visiteur outre le fait que celui-ci peut effectuer des commentaires.
- **Testeur** : Un testeur correspond à un visiteur mais il peut gérer les différentes fonctionnalités liées aux tests.

Cependant, notons qu'une nouvelle classe d'utilisateur, non encore présente au sein de *GenSpec* va être introduite de part la gestion multi-documents. Il s'agira du concepteur :

- **Concepteur** : Un concepteur possède le même statut qu'un commentateur. Cependant, celui-ci peut effectuer différents actions vis-à-vis de la gestion multi-documents que seul l'administrateur peut également réaliser.

### 2.3. Structure organisationnelle

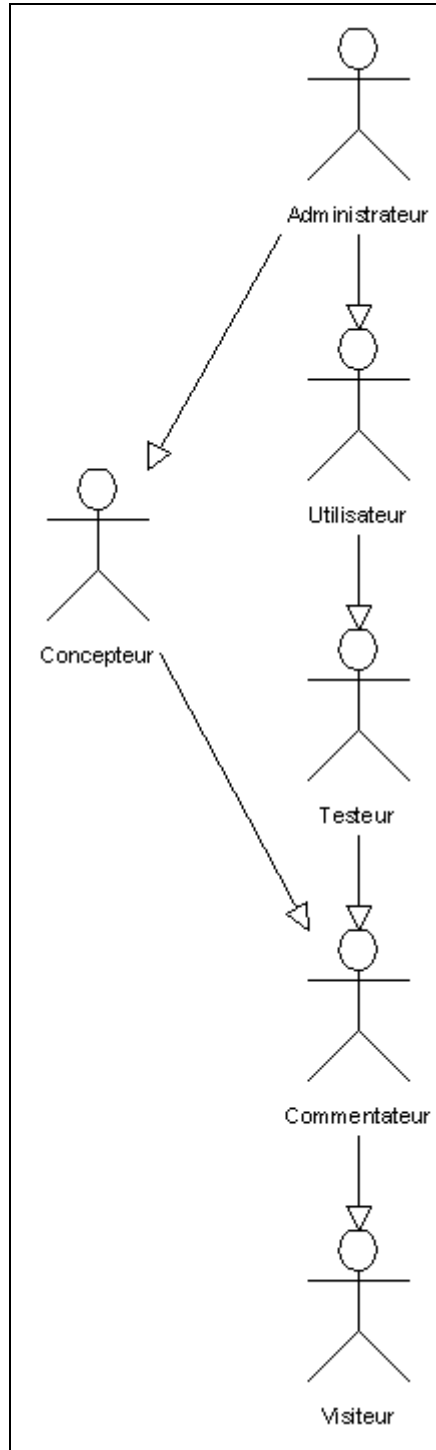


Figure 1 - Structure organisationnelle

Les différents types d'utilisateurs définis au sein de la section définissant les classes d'utilisateurs correspondent aux différents intervenants vis-à-vis d'un projet.

On a pu voir, au sein de cette section qu'il y avait une certaine hiérarchie quant aux droits accordés aux différents utilisateurs. Cette hiérarchie correspond à la hiérarchie présente entre les différents usagers lors de l'établissement des spécifications.

#### 2.4. Profil des classes d'utilisateurs

On décrit ici les différentes classes d'utilisateurs non pas en terme de droits associés aux fonctionnalités du logiciel mais bien en terme de caractéristiques liées à l'utilisateur. La description renvoie à la hiérarchie établie au sein de la section définissant la structure organisationnelle.

- **Administrateur** : On peut retrouver l'administrateur au sommet de la hiérarchie. Celui-ci est chargé de la gestion globale d'un projet. Il est chargé de contrôler et de mener à bien la réalisation des spécifications.
- **Utilisateur** : L'utilisateur, qui se situe juste au-dessous de l'administrateur, est chargé d'établir les exigences et l'ensemble des informations nécessaires qui leurs sont associées.
- **Testeur** : Le testeur est chargé de vérifier si l'ensemble des exigences ont bien été satisfaites.
- **Commentateur** : Le commentateur se retrouve dans les niveaux les plus bas de la hiérarchie. Celui-ci est chargé de commenter l'ensemble des exigences s'il le souhaite.
- **Visiteur** : Au plus bas de la hiérarchie, on peut retrouver le visiteur. Celui-ci correspond à tout lecteur du document des exigences. Il n'intervient pas à proprement parler dans le processus d'établissement des exigences (il peut s'agir d'un client, etc.).
- **Concepteur** : Le concepteur, quant à lui, ne répond pas à cette hiérarchie. Il se situe juste après l'administrateur et au-dessus du commentateur. En effet, il est chargé de réaliser des regroupements d'exigences en composants et il peut, s'il le souhaite également, établir des commentaires sur les exigences.

#### 2.5. Autre personnel impliqué

De nombreuses personnes sont influencées par l'utilisation de *GenSpec*. Outre les différents usagers, toutes les personnes ayant un lien quelconque avec le projet en cours de spécification/développement sont concernées par le travail réalisé par le biais de cet outil.

### 3. SOLUTIONS PROPOSEES

Au sein de ce chapitre, nous allons passer en revue les différentes solutions ou alternatives de solutions face au problème mis en lumière lors du chapitre précédent.

#### 3.1. Justification et nature des changements

Cette section comporte une discussion des différents problèmes relevés vis-à-vis de la version actuelle de *GenSpec*. Ensuite, nous envisagerons une solution générale à ces problèmes et les différentes variantes qu'il serait possible de lui associer.

##### 3.1.1. Justification des changements

*GenSpec* est un outil permettant l'établissement et un suivi des exigences concernant le développement d'un projet. Ces différentes exigences sont introduites au sein de *GenSpec* sous forme textuelle en langage naturel. Cela présente certains avantages :

- Il n'est pas nécessaire d'apprendre un nouveau formalisme pour comprendre les spécifications établies.
- La communication entre les différents acteurs se veut plus aisée (le client est apte à comprendre les spécifications).
- Etc.

Cependant, le caractère textuel de ces spécifications présente un désavantage assez important : **l'ambiguïté**.

En effet, et c'est inhérent à toute forme de langage naturel, l'ambiguïté en est une caractéristique essentielle, elle est inévitable. Même si l'auteur connaît les règles de langage afin d'éviter toute ambiguïté lors de l'écriture d'une exigence, l'ambiguïté peut néanmoins apparaître lors de la lecture de celle-ci. Si le lecteur ne connaît pas ou n'est pas conscient des règles de langage, le contenu des exigences peut être compris d'une tout autre manière que souhaité [2].

De plus, de nombreuses difficultés peuvent apparaître lors de la lecture d'exigences. En effet, si les termes employés par les différents auteurs d'exigences ne sont pas définis, cela peut provoquer des confusions ou des incompréhensions entre les différents acteurs au sein d'un projet. De tels problèmes peuvent s'avérer coûteux pour le développement d'un projet.

Ces 2 problèmes principaux peuvent être retrouvés au sein de la version actuelle de *GenSpec*. C'est pourquoi, il semble opportun de développer un nouveau système répondant à ces nouveaux besoins.

### 3.1.2. Description des changements désirés

Pour répondre aux deux problèmes majeurs mis en lumière au sein du chapitre précédent, une nouvelle fonctionnalité va être insérée au sein de *GenSpec* : un **glossaire**.

Le glossaire doit répondre au deux besoins identifiés au sein du chapitre précédent. Premièrement, diminuer l'ambiguïté présente actuellement au sein des exigences et deuxièmement permettre de fournir un ensemble d'informations sur les termes employés au sein des exigences afin de minimaliser les incompréhensions.

#### 3.1.2.1. Changements aux fonctionnalités

On peut retrouver ici un ensemble d'informations concernant les modifications à apporter aux fonctionnalités actuellement présente au sein de *GenSpec* ainsi que les nouveaux aspects à introduire au sein du logiciel.

##### 3.1.2.1.1. Liaison du glossaire aux autres fonctionnalités

Le glossaire est une nouvelle fonctionnalité mais elle reste cependant très liée aux autres fonctionnalités déjà présentes au sein de *GenSpec*.

En effet, tout terme apparaissant au sein d'une exigence peut être défini au sein du glossaire. De ce fait, pour chaque terme défini au sein du glossaire, on peut retrouver une liste des exigences où apparait le terme considéré. Un terme du glossaire peut se retrouver dans tout type d'exigence (Fonction, Intrans, Extrans, Détail, etc.).

Le glossaire offre la possibilité de fournir la définition d'un terme directement à partir d'une exigence voire même à partir du titre d'une exigence. On peut également définir un terme directement à partir du glossaire. Tous les termes présents au sein du glossaire pourront également être directement inclus au sein de l'arbre hiérarchique de *GenSpec*, fournissant ainsi la possibilité de décrire de manière claire et précise tout intrant et extrant vis-à-vis des exigences.

Le lien entre le glossaire et les exigences va plus loin encore. En effet, le glossaire permettra de réduire la quantité de termes de même sémantique au sein des exigences de par la proposition de terme défini au sein du glossaire pour établir le même sens.

On peut s'apercevoir alors que le glossaire, bien qu'il soit une nouvelle fonctionnalité, présente une liaison importante avec les autres fonctionnalités existantes au sein de *GenSpec*.

##### 3.1.2.1.2. Informations associées à un terme

Afin de limiter les incompréhensions entre les termes employés, un ensemble d'informations doit être enregistré concernant un terme employé au sein des exigences. Ces informations ne se limitent pas uniquement à la définition (sémantique) du terme, on retrouve d'autres éléments vis-à-vis d'un terme :

- **Version** de l'ensemble des informations :
  - **Auteur** : Correspond à l'auteur de la version de l'ensemble des informations associées à un terme.
  - **Numéro de version** : Correspond au numéro de version de l'ensemble des informations associées à un terme.
  - **Date de modification** : Correspond à la date de modification de l'ensemble des informations associées à un terme (ou dans le cas de la création, à la date de création).



- **Propriétés :**
  - S'il s'agit d'un **Intrant** : Permet de définir le terme constitue un intrant.
  - S'il s'agit d'un **Extrant** : Permet d'établir si le terme constitue un extrant.
- **Définition** (Sémantique) : Correspond à l'explication du concept, du sens du mot selon l'utilisateur qui l'a défini. La définition correspond à la sémantique du terme.
- **But pour l'acteur** (humain ou autre système) : Correspond au but du terme pour tout acteur lié au projet en cours de spécification. L'acteur peut être un humain ou tout autre système.
- **But pour le système** : Décrit quelle est l'utilité du terme employé pour le système associé au projet en cours de spécification.
- Une **liste des exigences** dans lesquelles le terme apparaît : Correspond à une liste des exigences où le terme apparaît (titre, énoncé). Notons que cette liste d'exigence est générée de manière automatique et ne doit pas être complétée par l'utilisateur.
- Une **liste de synonymes** correspondant au terme : Correspond à une liste de mots synonymes du terme défini au sein du glossaire.
- Un **niveau d'ambiguïté** sur les informations contenues : Correspond à une définition du niveau d'ambiguïté de l'ensemble des informations associées à un terme.
- Un **commentaire** : Correspond à l'ensemble des commentaires que l'auteur de la dernière version de l'ensemble des informations juge comme important.

Notons qu'à ces informations peuvent être ajouté également la gestion de l'historique ainsi que les différentes niveaux de correspondance définis plus loin au sein de ce même document.

Il n'est pas nécessaire que la totalité de ces informations soit reprise au sein du glossaire pour définir un terme. Certains éléments peuvent être implicites (définition, but l'acteur, but pour le système) tout comme d'autres facultatifs.

#### 3.1.2.1.3. Génération d'un document associé au glossaire

Un autre élément important vis-à-vis du glossaire est la génération d'un document qui lui sera associé. Ce document comporte l'ensemble des informations retenues associées à chaque terme défini au sein du glossaire. Le document contient l'ensemble des informations associées à un terme et les différents termes seront classés par ordre alphanumérique. Il est prévu également, de manière optionnelle, de permettre une génération de document selon différents critères. Par exemple : Selon une sélection d'informations, selon une sélection de terme, dans un autre ordre qu'alphanumérique, etc.

La génération des informations associées à un terme (définition, but pour l'acteur, but pour le système) au sein du glossaire s'effectue de manière tabulaire. Deux options sont possibles :

- la génération du document avec quadrillage du tableau
- la génération du document avec ou sans titre des colonnes du tableau

#### 3.1.2.1.4. Importation de glossaire

Une autre fonctionnalité à associer au glossaire est liée à l'importation de données. Le glossaire de *GenSpec* doit permettre la réutilisation d'un glossaire émanant d'un projet *GenSpec* précédant. Cette fonctionnalité présente certains avantages. En effet, une fois qu'un terme aura été défini au

sein du projet, celui-ci pourra être réutilisé rapidement au sein d'un projet différent et ce, sans avoir à réintroduire l'ensemble des informations de multiples fois.

#### **3.1.2.1.5. Vérification de termes subjectifs**

Un autre aspect non directement lié à la fonction de glossaire mais qui permet néanmoins d'améliorer la qualité des exigences par diminution de l'ambiguïté est la vérification de l'emploi de termes subjectifs au sein des exigences. En effet, cette fonctionnalité permet de définir des termes qui, par uniquement par leur emploi, crée de l'ambiguïté car cela dépend de la perception de chacun. Par exemple : correcte, complet, clair, etc.

#### **3.1.2.1.6. Gestion de l'historique**

Une gestion de l'historique est proposée au sein du glossaire. Cette option présente une avancée importante vis-à-vis de la globalité du logiciel. Il est question de gérer l'ensemble des versions de l'ensemble des informations associées à un terme au sein du glossaire. Chaque modification constitue un nouvel historique pour lequel il est possible de consulter les informations.

Un filtre est également établi vis-à-vis de l'historique. On pourrait consulter les différentes informations selon certains critères (auteur, date, etc.).

Une justification est demandée à l'utilisateur à chaque modification des informations associées à un terme au sein du glossaire. Cela permet alors de consulter connaître de manière simple et efficace les éléments qui ont subis une modification et la cause de cette modification.

#### **3.1.2.1.7. Niveaux de correspondance**

Le glossaire peut fournir également un moyen de correspondance entre les termes définis au sein du glossaire par le biais de l'utilisation de niveaux de correspondance. Ceux-ci permettent de définir la similarité sémantique entre 2 termes de manière simple et efficace par l'utilisation de niveaux prédéfinis. Notons qu'il est impossible de définir deux termes comme étant sémantiquement identique car cela induirait la possibilité d'utiliser deux termes distincts au sein des exigences pour référer à un même concept. Or, c'est une situation que l'on souhaite éviter.

#### **3.1.2.2. Les changements aux interfaces**

Le glossaire au sein de *GenSpec* entraîne quelques modifications en termes d'interfaces vis-à-vis du logiciel. Il n'y a pas de modifications importantes vis-à-vis des fenêtres déjà présentes au sein du logiciel. Le glossaire constitue à lui seul une nouvelle fenêtre autonome au sein de l'application. Cependant quelques éléments de base au sein du glossaire tel que le menu et l'aide subiront des modifications.

Tout d'abord, la fenêtre exclusivement associée au glossaire ne contient que l'ensemble des informations associées à un terme – Voir *Informations associées à un terme* plus haut au sein de ce document -. Outre ces informations, on peut retrouver au sein de cette fenêtre une liste des termes définis au sein glossaire. A partir de cette liste, il est alors aisé de consulter toutes informations associées à un terme au sein de cette même fenêtre.

Les menus de génération de documents et de recherche sont également plus complets. Ils contiennent des informations associées directement au glossaire. Une recherche est alors possible vis-à-vis des termes définis au sein du glossaire. Différentes options de génération associées au glossaire sont également présentes dans le menu de génération.

L'aide se veut également plus complète. Des explications concernant le fonctionnement du glossaire y sont consignées.

Dans la barre de menu générale, il est également possible de réorganiser, tout comme les autres fenêtres dans *GenSpec*, la fenêtre associée au glossaire.

Les modifications apportées par le nouveau système aux interfaces déjà présentes sont donc très légères. Le changement le plus conséquent est associé à la création d'une nouvelle fenêtre récapitulative vis-à-vis de l'ensemble des données liées à un terme.

### 3.1.2.3. Changements aux classes d'utilisateurs

De nouveaux droits sont associés aux différentes catégories d'utilisateurs présentes au sein de *GenSpec* :

- **Administrateur** : L'administrateur possède tous les droits vis-à-vis du glossaire. Il peut définir des termes, modifier les informations qui leurs sont associées, accéder aux versions historiques d'un terme - supprimées ou non – (fonctionnalité optionnelle) etc. De plus, l'administrateur peut définir le caractère obligatoire ou non des différentes informations associées à un terme du glossaire (fonctionnalité optionnelle). De ce fait, si une information est obligatoire, lors de l'insertion d'un terme au sein du glossaire par tout utilisateur, il sera dans l'obligation de fournir l'information associée.
- **Utilisateur** : L'utilisateur possède l'ensemble des droits de l'administrateur vis-à-vis du glossaire excepté qu'il ne peut établir quels sont les informations obligatoires à fournir lors de la définition. De plus, il ne peut accéder aux versions historiques supprimées d'un terme (fonctionnalité optionnelle).
- **Testeur, Commentateur, Visiteur, Concepteur** : Ces 4 types d'utilisateurs présentent tous les mêmes droits vis-à-vis du glossaire. Ils ne peuvent que consulter les informations qui sont associées à un terme et générer les documents en conséquence. De plus, ils peuvent également définir le niveau d'ambiguïté associé aux informations liées à un terme.

### 3.1.3. Ordre de priorité dans les changements

Cette partie renferme une description des différents niveaux de priorités associés aux diverses nouvelles fonctionnalités que l'on a passé en revue au sein de la sous-section *3.1.2. Description des changements désirés*.

La liaison du glossaire aux autres fonctionnalités présente un caractère essentiel vis-à-vis de la nouvelle fonctionnalité que constitue le glossaire. En effet, la possibilité de définir un élément au sein du glossaire pour l'insérer dans l'Arbre hiérarchique répond à une demande majeure de la part des utilisateurs de *GenSpec*. De plus, la gestion des synonymes afin de limiter le nombre de mots de même sémantique au sein des exigences est également un élément important à considérer comme essentiel.

La génération des documents liés au glossaire et l'ensemble des informations qui sont associées aux termes est également fortement liée et considérés comme fonctionnalités essentielles. En effet, la génération du document associé au glossaire est tributaire des informations fournies par l'utilisateur lors de la définition d'un terme.

Ces fonctionnalités représentent les éléments essentiels à établir vis-à-vis du glossaire. Ces éléments constituent la base du développement. Si l'on ne retrouve pas ces fonctionnalités, la fonction de glossaire ne serait que très peu utile.

Par contre, en ce qui concerne la gestion de l'historique et des niveaux de correspondance définies ci-dessus, ces deux fonctionnalités peuvent être catégorisée d'optionnelles. En effet, ces fonctionnalités s'avèrent pratiques voire élégantes mais très peu utiles réellement.

Il en est de même vis-à-vis de la fonctionnalité d'importation. Bien qu'on ne puisse lui associer une importance optionnelle, celle-ci possède un niveau d'importance moindre vis-à-vis des fonctionnalités essentielles définies précédemment. Cependant, cela n'enlève pas le caractère essentiel à associer à cette fonctionnalité.

### **3.1.4. Changements considérés mais non inclus**

On peut retrouver au sein de cette sous-section les changements qui ont été considéré lors de la sous-section *3.1.2. Description des changements désirés* mais qui ne seront pas inclus dans la version finale du glossaire de *GenSpec*.

#### **3.1.4.1. Gestion de l'historique**

La gestion de l'historique qui était une fonction élégante à associer au glossaire de *GenSpec* a été écartée. Le caractère avancé de cette fonctionnalité par rapport à la globalité du logiciel en est la cause majeure. En effet, cette fonctionnalité devrait s'appliquer sur la globalité du logiciel, y compris les exigences et non uniquement le glossaire.

#### **3.1.4.2. Niveaux de correspondance**

La fonction de niveaux de correspondance à associer au glossaire aurait été intéressante à développer. Elle aurait permis de relier des termes de sens proches mais non identiques au sein du glossaire. Selon un point de vue plus avancé, elle aurait également pu proposer à l'utilisateur des termes alternatifs à ceux employés de par les liaisons établies par ces niveaux.

Cependant, cette fonctionnalité jugée utile dans un premier temps a été supprimée car elle aurait obtenu un très faible taux d'utilisation.

### 3.2. Éléments de solution au sein de logiciels concurrents

Cinq logiciels concurrents ont été analysés afin d'observer les différents types de solutions apportés aux problèmes considérés. Nous n'avons pas relevé une fonctionnalité de glossaire aussi avancée que celle que nous prévoyons d'insérer au sein de *GenSpec*. Dans les subdivisions ci-dessous, nous reportons les éléments que nous avons considérés comme probant vis-à-vis du développement d'une nouvelle fonctionnalité telle que nous l'envisageons.

#### 3.2.1. Telelogic Doors 8.1

Au sein de logiciel *Doors*, nous n'avons pas retrouvé de glossaire au sens où nous l'entendons. Néanmoins, on peut retrouver deux types de glossaire au sein de ce logiciel. Un glossaire associé à l'ensemble des utilisateurs d'un projet et un glossaire personnel associé à chaque utilisateur. De plus, de nombreuses options de vérifications orthographiques sont disponibles au sein du logiciel.

Cependant un élément intéressant, semblable à la fonction de *liste de synonymes* a été retrouvée dans ce logiciel. Ainsi, il est possible d'associer un terme à un autre au sein du glossaire. De cette manière, lorsque le terme associé est employé au sein d'une quelconque formulation, lors de la vérification, on va proposer pour remplacer le terme associé le terme auquel il est associé.

#### 3.2.2. IRqA 3.5.0

Le logiciel *IRqA* présente un glossaire. La fonctionnalité de glossaire présente au sein de ce logiciel est une des fonctionnalités les plus avancées que l'on ait pu constater.

A chaque terme est associé un ensemble d'informations, tel que l'auteur, la date de modification, de validation, etc. On peut associer également une liste d'éléments qui sont en relation ou encore les éléments qui ne sont pas en relation avec le terme au sein du projet. Il est également possible de lui associer une liste de synonymes et de définir toute une liste de propriétés qui peuvent lui être associées. On retrouve également une gestion de l'historique assez importante permettant de consulter toutes les versions émanant de ces modifications ainsi que la raison de la modification. Cela permet une bonne gestion de la traçabilité vis-à-vis des spécifications du projet. Il existe également un lien entre les mots définis au sein du glossaire et les exigences. Ainsi, pour chaque exigence, il est possible d'obtenir une liste des mots définis au sein de cette exigence.

#### 3.2.3. REM 1.2.2

On ne retrouve pas de fonctionnalité de glossaire en tant que tel au sein de ce logiciel.

#### 3.2.4. Requisite Pro 7.0.0.0

*Requisite Pro* fournit la possibilité de définir des termes au sein d'un glossaire "simplifié". En effet, au sein de ce logiciel, il est possible de fournir une simple définition à un terme. Cependant, cette définition possède la même forme que les exigences. La structure des exigences est détournée au sein de ce logiciel afin de définir des termes employés. Néanmoins, on peut associer un niveau d'ambiguïté associé à la définition ainsi qu'un ensemble d'autres attributs.

Une fonctionnalité intéressante que l'on retrouve également au sein de *Requisite Pro* est la possibilité d'établir une discussion sur n'importe quel élément et ce, y compris pour les termes définis. Différents acteurs peuvent alors intervenir sur un certain sujet associé à un élément.

Une gestion importante de l'historique est encore présente ici également. On peut consulter les différentes versions de manière simple et complète.

On ne peut affirmer qu'il y a un réel glossaire au sein de logiciel mais plus une possibilité de définir des termes et quelques propriétés selon la structure des exigences.

### 3.2.5. Artisan Studio 6.1.21 (SysML)

Il n'y a pas de fonction de glossaire comme nous l'entendons au sein de ce logiciel. *Artisan Studio* fournit, en guise de glossaire, une liste de tous les types d'objets et de liens présents dans le projet. Par exemple, on peut y retrouver les acteurs, les dépendances, les cas d'utilisation, etc. Il n'y a pas de glossaire permettant de définir les termes utilisés dans les exigences.

L'ensemble de ces informations peuvent être générée au sein d'un document *Word*.

## 4. RECOMMANDATIONS

Cette partie du document renferme la solution à développer pour répondre aux attentes et aux souhaits des clients en tenant compte de différents éléments définis plus haut au sein de ce document.

### 4.1. Contexte, objectifs et étendue du nouveau système

Cette partie comporte une description plus précise et plus complète de la solution à établir. Pour cela, nous allons utiliser plusieurs captures d'écrans correspondant à une proposition d'interface afin de décrire les différents éléments de solution.

Notons que le contenu général des informations et des fonctionnalités à associer au glossaire de *GenSpec* a été défini au sein du chapitre précédent. Il ne s'agit alors ici que de fournir un exemple de solution tout en expliquant certains éléments de manière plus détaillée.

#### 4.1.1. Description Générale

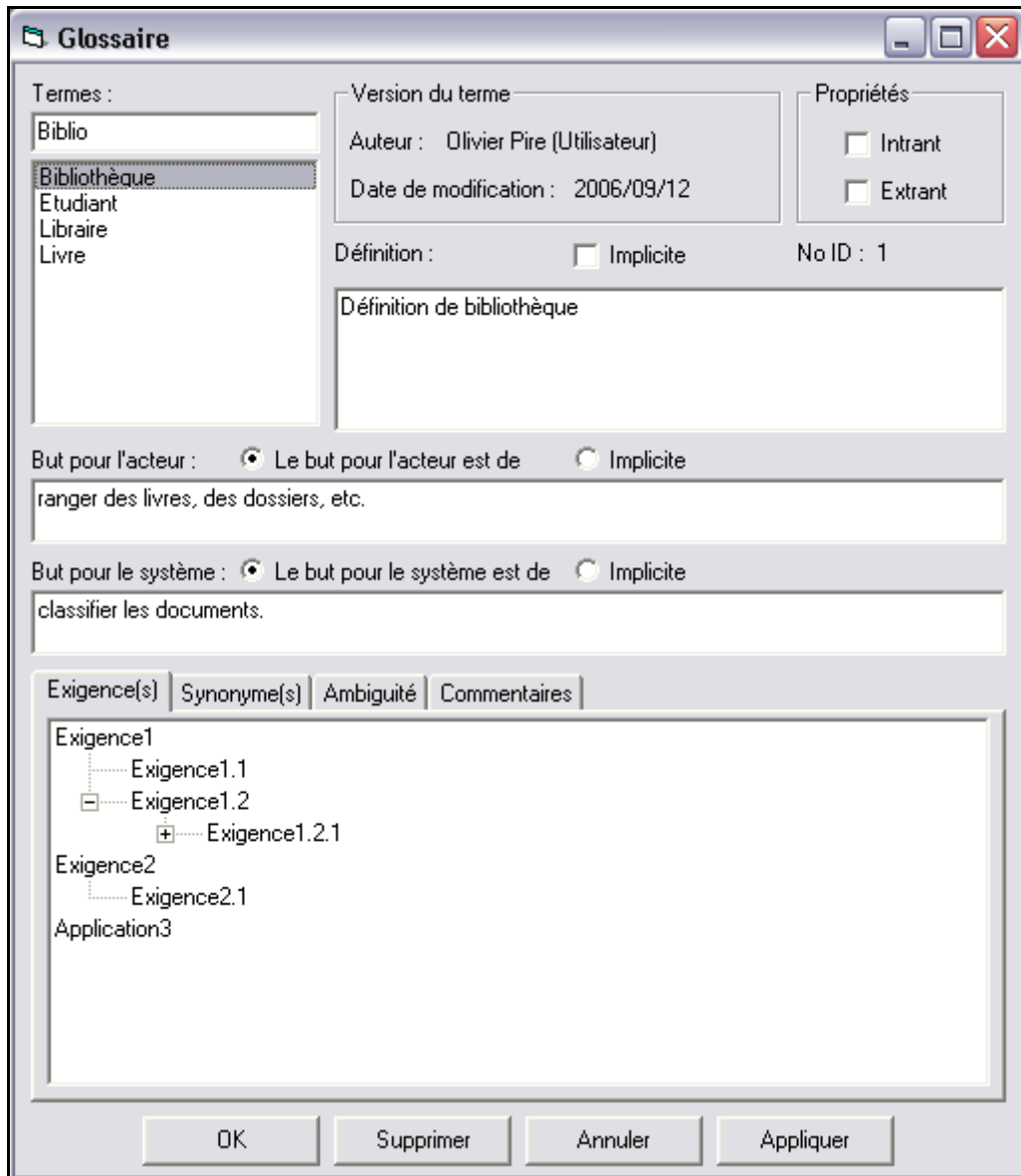
Le glossaire pour *GenSpec* permet de répondre aux problèmes définis précédemment. Tout d'abord, Il permet de limiter l'ambiguïté lors de la réalisation des spécifications associées à un projet. Ensuite, il permet d'uniformiser l'utilisation de certains termes au sein des exigences afin de produire des spécifications les plus simples, claire et non ambiguës pour l'ensemble des acteurs. Il répond également à un besoin de compréhension général. En effet, le fait de pouvoir associer un ensemble d'informations à chaque terme permet, à tous les acteurs, de mieux se comprendre et d'éviter toute discussion lié à une mauvaise définition.

C'est dans cette optique d'optimisation et de compréhension qu'un glossaire va être inséré au sein de *GenSpec*.

#### 4.1.2. Description détaillée

Cette partie comporte une description de la solution à établir pour répondre aux besoins établis. Afin de faciliter la description de la solution, un ensemble de capture d'écran va être utilisé pour fournir un exemple de solution. Il est important de noter que ces interfaces ne correspondent pas obligatoirement à la solution finale à apporter. Les interfaces présentes au sein de cette sous-section sont donc sujettes à modifications.





**Glossaire**

Termes :

- Biblio
- Bibliothèque**
- Etudiant
- Libraire
- Livre

Version du terme

Auteur : Olivier Pire (Utilisateur)

Date de modification : 2006/09/12

Propriétés

Intrant

Extrant

Définition :  Implicite

No ID : 1

Définition de bibliothèque

But pour l'acteur :  Le but pour l'acteur est de  Implicite

ranger des livres, des dossiers, etc.

But pour le système :  Le but pour le système est de  Implicite

classifier les documents.

Exigence(s) | Synonyme(s) | Ambigüité | Commentaires

- Exigence1
  - Exigence1.1
  - Exigence1.2
    - Exigence1.2.1
- Exigence2
  - Exigence2.1
- Application3

OK   Supprimer   Annuler   Appliquer

Figure 2 - Glossaire avec exigences

Pour chaque terme défini au sein du glossaire de *GenSpec*, on retrouve un ensemble d'informations (cf. 3.1.2.1.2. Informations associées à un terme).

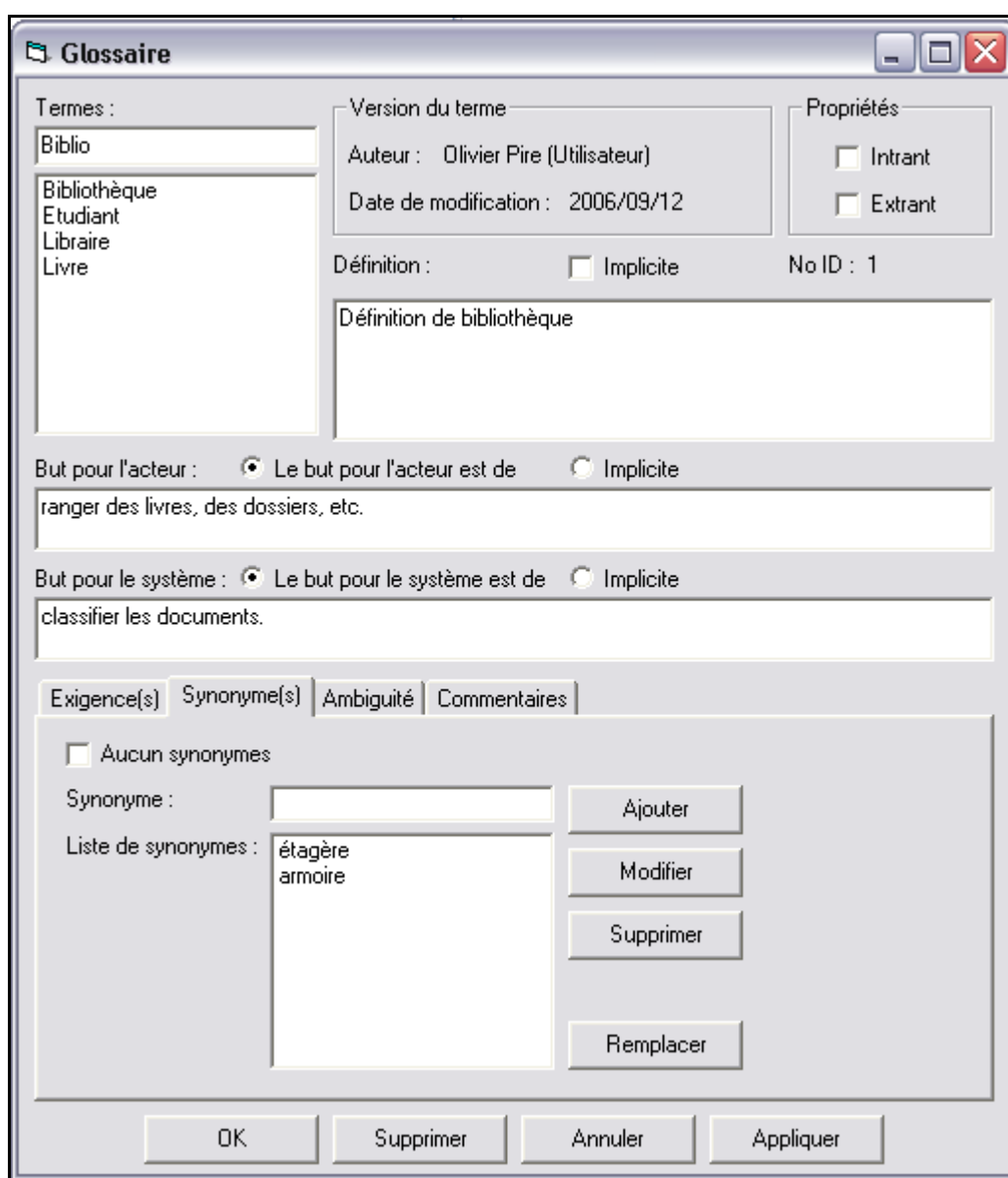
La Figure 2 - Glossaire avec exigences présente que certaines de ces informations peuvent être déclarée comme implicites marquant le fait que ces informations ne nécessitent pas d'explications. Dans le cas où ces informations ne sont pas implicites, l'utilisateur doit respecter un style de phrase imposé tel que "Le but pour l'acteur est de" et "Le but pour le système est de".

On peut constater de part la structure établie au sein de la fenêtre qu'un terme peut être à la fois comme propriété un intrant, un extrant, les deux en même temps ou aucun des deux. Et ce, afin de bien représenter l'utilisation possible de la définition d'intrant et d'extrant au sein de *GenSpec*.

Un numéro d'identifiant est également associé à chaque terme de tel manière que quelque soit la modification qui lui soit apportée, il soit toujours reconnaissable pour tout utilisateur.

Notons également qu'au sein de l'onglet *Exigence(s)*, on retrouve la liste des exigences – sous forme d'arbre - où le terme apparaît sous forme de structure d'arbre. Et ce, quelque soit le statut du terme, c'est-à-dire qu'il soit simplement utilisé au sein de l'exigence, dans le titre de l'exigence ou qu'il constitue un intrant ou un extrant. De plus, de part la sélection dans l'arbre des exigences se retrouvant au sein de la fenêtre des exigences, il est possible d'accéder à l'énoncé de l'exigence.

La structure de la fenêtre présentée fournit un avantage certain. En effet, ce sont les informations les plus importantes qui sont affichées de manière constante et ce quelque soit l'onglet d'option sélectionné.



The screenshot shows a window titled "Glossaire" with several sections:

- Termes :** A list box containing "Biblio", "Bibliothèque", "Etudiant", "Libraire", and "Livre". "Biblio" is selected.
- Version du terme :**
  - Auteur : Olivier Pire (Utilisateur)
  - Date de modification : 2006/09/12
- Propriétés :**
  - Intrant
  - Extrant
- Définition :**
  - Implicite
  - No ID : 1
  - Text area containing "Définition de bibliothèque"
- But pour l'acteur :**
  - Le but pour l'acteur est de
  - Implicite
  - Text area containing "ranger des livres, des dossiers, etc."
- But pour le système :**
  - Le but pour le système est de
  - Implicite
  - Text area containing "classifier les documents."
- Exigence(s) | Synonyme(s) | Ambiguïté | Commentaires** (Current tab: Synonyme(s))
  - Aucun synonymes
  - Synonyme : [Empty text box] [Ajouter]
  - Liste de synonymes :
    - étagère
    - armoire
  - [Modifier]
  - [Supprimer]
  - [Remplacer]
- Buttons at the bottom: OK, Supprimer, Annuler, Appliquer

Figure 3 - Glossaire et liste de synonymes

La *Figure 3 - Glossaire et liste de synonymes* présente un exemple de solution vis-à-vis de la liste des synonymes. Il est possible ici pour l'utilisateur de définir tous les termes qu'il considère comme étant des synonymes au sein de la définition d'un terme. La liste des synonymes doit d'ailleurs être visible pour utilisateur. On retrouve également la possibilité de remplacer le terme général défini au sein du glossaire par un de ses synonymes (dans le cas où celui-ci serait considéré comme plus

opportun). Bien entendu, la liste de synonyme peut être vide et l'utilisateur doit pouvoir la définir comme tel.



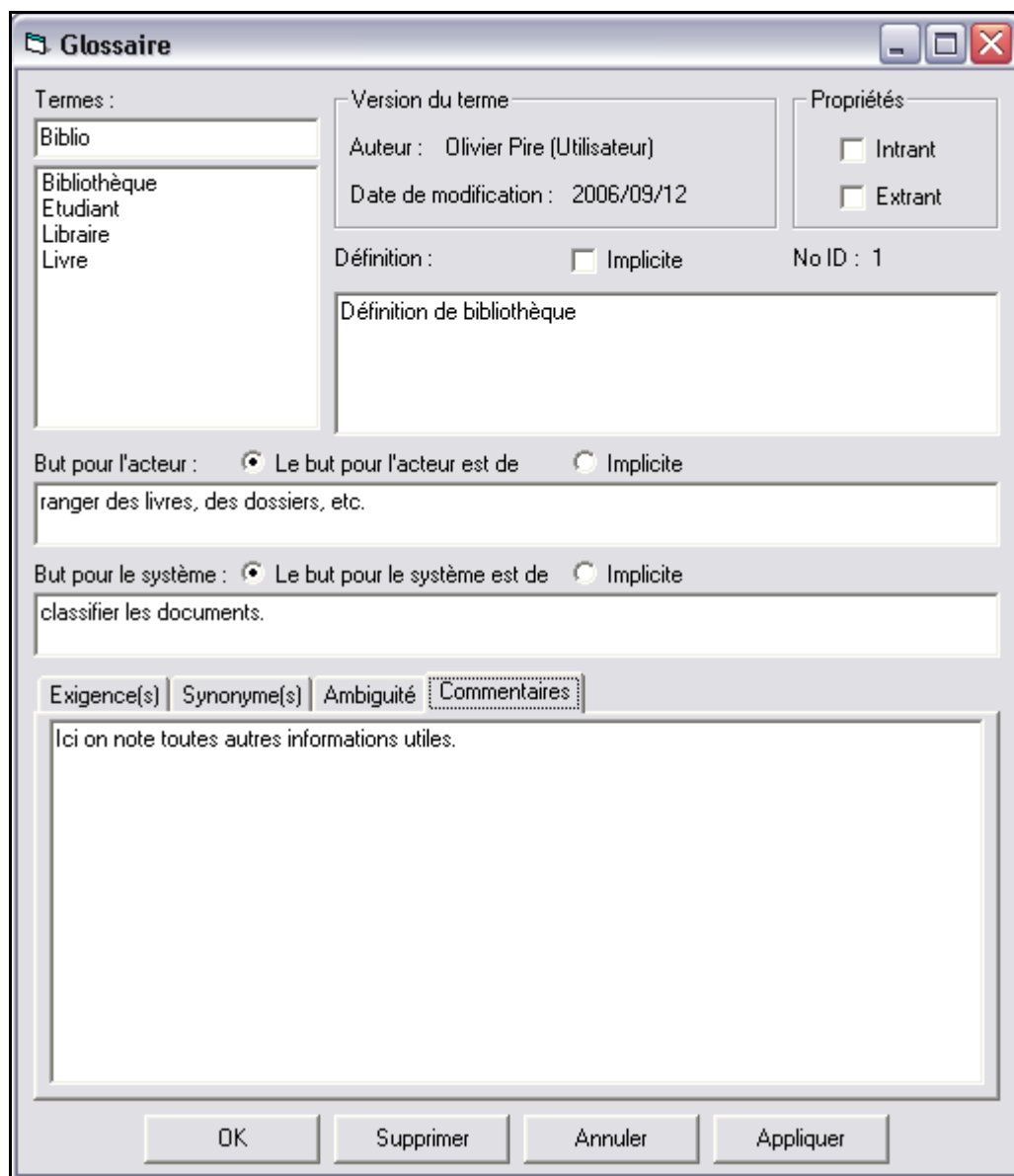
The screenshot shows a window titled "Glossaire" with a list of terms on the left: "Biblio", "Bibliothèque", "Etudiant", "Libraire", and "Livre". The "Bibliothèque" term is selected. The main area is divided into several sections:

- Version du terme:** Ateur : Olivier Pire (Utilisateur), Date de modification : 2006/09/12
- Propriétés:**  Intrans,  Extrans
- Définition:**  Implicite, No ID : 1
- Définition de bibliothèque:** A text area containing "Définition de bibliothèque".
- But pour l'acteur:**  Le but pour l'acteur est de,  Implicite. Text area: "ranger des livres, des dossiers, etc."
- But pour le système:**  Le but pour le système est de,  Implicite. Text area: "classifier les documents."
- Ambiguïté tab:**
  - Niveau d'ambiguïté : Faible
  - Justification : La justification du niveau faible d'ambiguïté de la définition de bibliothèque.
  - Modification:** Niveau d'ambiguïté : Faible (dropdown menu)
  - Justification de l'évaluation : (empty text area)

Buttons at the bottom: OK, Supprimer, Annuler, Appliquer.

Figure 4 - Glossaire et ambiguïté

La *Figure 4 - Glossaire et ambiguïté* présente l'aspect lié à la définition de l'ambiguïté d'un terme. Elle permet de fournir un niveau d'ambiguïté préétabli et justifier le pourquoi d'une telle évaluation.



The screenshot shows a window titled "Glossaire" with a list of terms on the left: "Biblio", "Bibliothèque", "Etudiant", "Libraire", and "Livre". The "Biblio" term is selected. The main area displays the following information:

- Version du terme:** Auteur : Olivier Pire (Utilisateur), Date de modification : 2006/09/12
- Propriétés:**  Intransit,  Extransit
- Définition:**  Implicite, No ID : 1
- Définition de bibliothèque:** (Empty text box)
- But pour l'acteur:**  Le but pour l'acteur est de,  Implicite  
ranger des livres, des dossiers, etc.
- But pour le système:**  Le but pour le système est de,  Implicite  
classifier les documents.
- Exigence(s) | Synonyme(s) | Ambiguïté | Commentaires:** (The "Commentaires" tab is active)
- Commentaires:** Ici on note toutes autres informations utiles.

Buttons at the bottom: OK, Supprimer, Annuler, Appliquer.

Figure 5 - Glossaire et commentaires

Pour terminer, la *Figure 5 - Glossaire et commentaires* montre qu'il est possible, pour tout utilisateur possédant les droits de définir un terme, de noter un quelconque commentaire jugé utile au sein de l'onglet commentaire.

Il est important de noter que le glossaire pour *GenSpec* ne se limite pas qu'à ces interfaces. On doit également prendre en compte les aspects liaisons aux exigences et à l'arbre hiérarchique en compte (cf. 3.1.2.1.1. *Liaison du glossaire aux autres fonctionnalités*).

De plus, Il est important également de ne pas oublier la génération de document (cf. 3.1.2.1.3. *Génération d'un document associé au glossaire*) ainsi que le mécanisme d'importation de glossaire (cf. 3.1.2.1.4. *Importation de glossaire*) bien qu'il soit moins important.

## 5. ANALYSE DU SYSTEME PROPOSÉ

Ce chapitre fournit une analyse des avantages et inconvénients liés à la réalisation de cette solution.

### 5.1. Sommaire des améliorations

Il apparaît de manière claire que le glossaire de *GenSpec* va permettre d'améliorer la qualité des exigences établies par l'intermédiaire du logiciel.

Il procurera une meilleure compréhension de l'ensemble des termes employés et diminuera les incompréhensions entre les différents acteurs du projet.

L'emploi de la liste de synonymes permettra de limiter le nombre de termes employés référant à la même sémantique. De cette manière, les exigences seront simplifiées et il y aura une cohérence plus importante entre les différents éléments. De plus, cette fonctionnalité permet également l'emploi de raccourcis. À savoir, l'utilisateur peut employer cette fonctionnalité pour remplacer un raccourci qu'il aurait noté au sein d'une exigence par le terme considéré au sein du glossaire.

Le glossaire fournit également un gain de temps considérable lors de la rédaction des exigences. De part l'utilisation des raccourcis, la possibilité de définir des intrants et extrants et de les déplacer directement dans l'arbre hiérarchique et la compréhension plus rapide des termes employés, la rédaction des exigences s'en verra facilitée.

De plus, de part la génération du document associé au glossaire, cela permettra une meilleure compréhension des spécifications à tout lecteur intéressé par ces spécifications.

### 5.2. Désavantages et limites

La réalisation du glossaire ne peut répondre complètement au besoin de bannir complètement l'ambiguïté des spécifications et ce, pour deux choses. Tout d'abord, et pour la simple raison, que le fonctionnement du glossaire est basé sur l'ensemble des informations que les utilisateurs voudront bien lui fournir. Le glossaire est tributaire des informations fournies par les utilisateurs. Une gestion automatisée semble inapplicable également. En effet, certains termes sont bien spécifiques à un projet et sont donc, par conséquent difficilement généralisable. De plus, l'ambiguïté est une caractéristique inhérente du langage. Par définition, la richesse, l'expressivité d'un langage vient de la possibilité d'exprimer de différentes manières des idées correspondantes.

Le glossaire nécessite un investissement important en termes de temps pour une utilisation correcte et complète. Cependant, il est important de noter qu'une fois que l'ensemble des termes ont été correctement défini, le temps imparti à établir le glossaire est rentabilisé lors de la rédaction des exigences.

## **Annexe D**

# **Etude : GMDE et AECF**

La dernière modification du contenu de ce document date du 8 novembre 2006.

*Étude*

**GenSpec**

**Gestion multi-documents**

**ETU10003**

**OCTOBRE 2006**

## 1. INTRODUCTION

### 1.1. Objet

Cette étude décrit les nouvelles fonctionnalités qui vont être intégrées à GenSpec. Il présente :

- la situation actuelle de GenSpec ;
- les propositions de solution ;
- les recommandations ;
- l'analyse du système proposé.

Cet étude s'adresse aussi bien à toute personne concernée par la définition des exigences d'un produit ou service, qu'à celle qui utilise GenSpec ou est soucieuse des améliorations qui y sont apportées [1].

### 1.2. Portée

GenSpec est un outil permettant de capturer des exigences, d'effectuer des vérifications sur ces dernières et de les générer dans un document d'exigences.

Cependant, de nouvelles fonctionnalités doivent y être intégrées afin que l'outil puisse, d'une part, réaliser une attribution des exigences à des composants et des fonctions, d'autre part, être plus flexible quant à la manière d'agencer les exigences en arbres hiérarchiques d'exigences.

Ces développements permettront la génération de nouveaux types de documents personnalisables. C'est pourquoi nous les caractériserons comme faisant partie d'une gestion multi-documents dans la suite de cette étude.

### 1.3. Définitions, acronymes et abréviations

Regroupement	: Ensemble d'exigences regroupées dans un arbre hiérarchique d'exigences. Un regroupement ne possède qu'une seule exigence racine. Son but est de pouvoir travailler sur une partie uniquement des exigences.
Arbre global	: Arbre hiérarchique d'exigences pouvant posséder plusieurs racines et regroupant l'ensemble des exigences d'un projet.
Fenêtre de regroupement	: Fenêtre incluant l'arbre global, ainsi que les éventuels regroupements sous forme d'onglets.
Composant	: Un composant est un élément de base d'un ensemble plus complexe. Un système peut être décomposé en composants. Des exigences peuvent être assignées à des composants. L'ensemble des exigences d'un composant étant satisfaites implique que le composant est lui-même satisfait.
Fonction	: Les exigences attribuées à des fonctions indiquent la fonction - pour le concepteur - que jouent ces exigences dans le système.
SES	: Spécification d'exigences de système.
SEI	: Spécification d'exigences d'interface.
Spécification	: Ensemble d'exigences agencées selon un format hiérarchique. Une spécification est l'ensemble des exigences présentes sous la racine d'un arbre hiérarchique – tel qu'utilisé actuellement dans GenSpec – ou un des ensembles d'exigences présentes sous une des racines de l'arbre global.



#### 1.4. Références

- [1] Ingénierie des exigences, une méthode simple et systématique, revue canadienne de l'IEEE, automne 2004. (<http://iee.ca/canrev/cr48/cr48.pdf>)
- [2] Ingénierie des exigences, l'outil de support GenSpec, revue canadienne de l'IEEE, automne 2005. (<http://iee.ca/canrev/cr51/cr51.pdf>)
- [3] Rubrique Aide de GenSpec.

#### 1.5. Vue d'ensemble

Ce document se compose des 5 chapitres suivants :

1. INTRODUCTION : Ce chapitre fournit une vue d'ensemble de toute l'étude. Il en décrit :
- l'objet ;
  - la portée ;
  - les définitions, acronymes et abréviations ;
  - les références ;
  - une vue d'ensemble.
2. SITUATION ACTUELLE : Ce chapitre décrit la situation actuelle telle qu'elle existe dans GenSpec et montre les effets du développement d'une gestion multi-documents. Il présente :
- une description de GenSpec ;
  - le profil de chaque type d'utilisateurs.
3. PROPOSITION DE SOLUTIONS : Ce chapitre décrit les propositions de solution apportées permettant de répondre aux besoins d'une gestion multi-documents. Il présente :
- une analyse des outils commerciaux de support à l'ingénierie des exigences ;
  - la justification des changements ;
  - la description des changements ;
  - l'ordre de priorité des changements ;
  - les classes d'utilisateurs affectées ;
  - la centralisation des informations ;
  - l'attribution des exigences à des composants.
4. RECOMMANDATIONS : Ce chapitre décrit, en fonction de propositions d'interfaces graphiques, les recommandations à adopter afin de répondre aux besoins. Les recommandations sont décomposées selon les deux parties du projet :
- l'attribution des exigences à des composants ;
  - la centralisation des informations.
5. ANALYSE DU SYSTÈME PROPOSÉ : Ce chapitre fournit une analyse des bénéfices, limites, avantages et inconvénients de l'ajout de fonctionnalités relatives à une gestion multi-documents pour GenSpec. Il présente donc :

- le sommaire des améliorations ;
- les désavantages et limites.

## 2. SITUATION ACTUELLE

Cette section présente la situation actuelle de GenSpec telle qu'elle existe et montre les effets du développement d'une gestion multi-documents dans son contexte.

### 2.1. Description de GenSpec

GenSpec a été conçu afin de répondre à un problème majeur lié au processus de développement d'un système, à savoir à l'ingénierie des exigences [2].

En effet, cette dernière est bien souvent une étape négligée dans le processus de développement d'un système. Les conséquences d'une telle négligence peuvent alors se révéler coûteuses, les besoins du client pouvant être mal compris par le fournisseur.

Il faut dès lors s'attendre, d'une part, à une augmentation des coûts et des délais de réalisation, d'autre part, à une diminution de la qualité du produit ou du service.

Afin de répondre à cette problématique, l'outil GenSpec propose une méthode simple et systématique [1] permettant entre autre :

- la rédaction simple et structurée des exigences ;
- leur classement selon un modèle hiérarchique ;
- la possibilité de lier les exigences entre elles afin de mettre en évidence les relations entre elles ;
- la génération des exigences dans un format personnalisable.

### 2.2. Profil de chaque type d'utilisateurs

Pour en savoir plus sur le profil de chacun des types d'utilisateurs de GenSpec, se référer à l'aide de GenSpec [3]. Cette dernière présente les 5 types d'utilisateurs présents au sein de l'outil.

Notons que ces 5 types d'utilisateurs sont les suivants : Administrateur, utilisateur, commentateur, visiteur et le testeur.

## 3. PROPOSITIONS DE SOLUTIONS

Dans cette section, nous proposons des alternatives de solution répondant aux besoins décrits ci-dessus. Une analyse préalable de certains outils commerciaux de support à l'ingénierie des exigences a permis de collecter des informations intéressantes en ce qui concerne l'organisation des exigences.

Notons que nous pouvons décomposer la gestion multi-documents en deux parties distinctes:

- l'attribution des exigences à des composants ;
- la centralisation des informations.

Cette découpe permet de traiter ces deux besoins distinctement. De plus, elle donne une vision plus claire de la gestion multi-documents.

### 3.1. Analyse des outils commerciaux de support à l'ingénierie des exigences

Nous citerons les idées pertinentes retenues suite à l'analyse de ces logiciels. Rappelons que l'analyse de ces outils s'est limitée aux fonctionnalités similaires à une gestion multi-documents. Les idées retenues pour chacun de ces logiciels sont synthétisées dans la suite de cette section.

Les logiciels qui ont été analysés sont les suivants :

- Telelogic Doors 8.1
- SysML pour Artisan Studio 6.1.21
- REM 1.2.2
- IRqA 3.5.0
- Requisite Pro 7.0.0.0

L'analyse de ces outils s'est révélée être très intéressante, cette dernière a mis en évidence de nombreuses façons de représenter les exigences. Qu'il s'agisse de listes, de tableaux, de diagrammes ou de graphiques. Tout ce qui a trait à une gestion multi-documents au sens de GenSpec, se matérialise par le fait que des exigences peuvent être représentées dans des hiérarchies de composants. Chaque composant peut être agencé de manière indépendante aux autres composants. De plus, chaque composant peut être généré selon une politique prédéfinie.

Remarquons enfin que dans plusieurs de ces outils, les composants peuvent contenir plusieurs types d'objets (exigences, composants, notes, acteurs, diagrammes UML, procédures de tests, etc.).

Si on s'intéresse de plus près au tableau d'attribution des exigences à des composants, nous remarquons que les outils analysés fournissent des moyens pour réaliser cette tâche. Nous pouvons même dire qu'ils supportent cette fonctionnalité de manière confondue avec la représentation des exigences. Ceci est rendu possible par ces outils de par l'utilisation de hiérarchie de dossiers et de par la polyvalence des objets que peut contenir un composant.

Retenons par exemple la possibilité d'assigner à un composant ou à une exigence, une URL pouvant faire des références internes ou externes à des fichiers. De plus, certains de ces logiciels permettent de déplacer ou copier des exigences en utilisant la technique du glisser déplacer ou copier.

### **3.2. Justification des changements**

Cette section décrit les problèmes rencontrés avec le système actuel et justifie le développement de nouvelles fonctionnalités. En effet, elle décrit les fonctionnalités présentes dans GenSpec et montre les limitations que rencontrent ces dernières. Cela permet de mettre en évidence le besoin d'une gestion multi-documents et ce, afin de surpasser ces limitations.

#### **3.2.1. Attribution des exigences à des composants**

Afin de réaliser la conception architecturale d'un projet, il est souhaitable de créer des composants. Ces derniers peuvent être organisés en niveaux afin de permettre une décomposition entre eux. Chaque composant contient ses propres exigences qu'il doit satisfaire. L'ensemble des composants devant couvrir toutes les exigences du système.

De plus, afin de déterminer les fonctions - pour le concepteur - des exigences pour le système, il doit être possible de créer des regroupements fonctionnels avec les exigences. Notons qu'aucune fonctionnalité présente dans GenSpec ne permet de répondre à ce besoin. Par conséquent, cette tâche d'attribution des exigences à des composants étant une étape essentielle pour le processus de développement établi chez Hydro-Québec, le besoin de développer une telle fonctionnalité peut être qualifiée d'essentielle.

Notons que cette tâche est jusqu'à présent réalisée de manière manuelle. Cela pose donc de nombreux problèmes de traçabilité. En effet, si une exigence est modifiée alors qu'elle est déjà référencée dans un composant, il se peut que des modifications doivent être apportées dans le document – réalisé manuellement - présentant ces exigences attribuées à des composants. Cette procédure de travail amène alors rapidement des erreurs et des oublis.

Enfin, elle permettra de générer un tableau qui mettra en correspondance les exigences avec les composants et fonctions du concepteur et ce, afin d'obtenir une vue tabulaire claire des informations relatives aux attributions d'exigences.

### 3.2.2. Centralisation des informations

Toutes les exigences d'un système sont introduites sous la forme d'un arbre hiérarchique, de sorte à fournir une représentation hiérarchique de ces dernières. Or, un projet peut couvrir un nombre important d'exigences. Par conséquent, cet arbre hiérarchique peut devenir suffisamment conséquent pour qu'une gestion efficace des exigences devienne ardue.

Ce qui se fait actuellement est de décomposer l'ensemble de ces exigences en plusieurs parties (nommées spécifications ou fonctions). Dans la suite de ce document, nous nommerons ces parties des regroupements. L'intérêt de cette méthode de travail est qu'il est dès lors possible de travailler sur des ensembles d'exigences plus restreint et donc plus facilement utilisables.

Chacun de ces regroupements peut alors être généré sous différents formats afin d'obtenir des documents mettant en forme l'ensemble des exigences destinées à la génération.

Cependant, vis-à-vis de l'utilisation de l'outil, un projet GenSpec a besoin d'être créé pour chacun de ces nouveaux regroupements, chacun enregistrant ses propres exigences dans une base de données Access.

Le problème majeur découlant de cette procédure de travail est que de nombreux projets, et donc de bases de données, doivent être créés pour enregistrer les données. Nous retrouvons alors plusieurs projets GenSpec et plusieurs bases de données pour un seul et même projet à spécifier.

Enfin, de par l'existence d'éventuelles interrelations entre les spécifications, une même exigence peut apparaître dans plusieurs regroupements distincts. Par conséquent, une même information relative à une exigence est enregistrée dans plusieurs bases de données différentes. Nous percevons donc un problème relatif à la redondance des données.

### 3.3. Description des changements

Le développement d'une gestion multi-documents va impliquer des changements, notamment liés aux fonctionnalités, processus et interfaces.

#### 3.3.1. Changements dans les fonctionnalités

De nouvelles fonctionnalités vont être intégrées à GenSpec, notamment la possibilité de créer plusieurs regroupements d'exigences au sein d'un même projet GenSpec ou encore la possibilité de posséder plusieurs racines dans un seul et même arbre hiérarchique d'exigences.

De plus, il sera possible de glisser des exigences d'un regroupement à l'autre en glissant les exigences.

Chaque regroupement pourra être généré selon des paramètres définis. Il sera possible de générer un document regroupant l'ensemble des exigences d'un projet, aussi bien les exigences du système que les exigences d'interfaces qui y sont associées.

Enfin, il sera possible d'attribuer des exigences, et ce à partir de n'importe quel regroupement, à des composants et des fonctions. Une vérification guidera l'utilisateur afin de détecter d'éventuelles incohérences. Une génération de document paramétrable sera disponible.

Remarque : Une description plus précise des changements de fonctionnalités est présente dans la section *Propositions de solutions* et *Recommandations*.

### 3.3.2. Changements dans les processus

La manière de décomposer les exigences en regroupements se fera beaucoup plus simplement. Cela a donc un impact considérable quant à la manière de gérer le processus. En effet, de nouvelles fonctionnalités rendront le processus plus simple à utiliser.

Vis-à-vis de l'attribution des exigences à des composants, cette fonctionnalité étant nouvelle, le processus de réalisation de cette activité sera radicalement différent par rapport à l'ancienne méthode qui consistait à réaliser les attributions « de façon manuelle ».

### 3.3.3. Changements dans les interfaces

Par interfaces, nous entendons les interfaces graphiques de travail.

Les nouvelles fonctionnalités vont notamment faire l'objet de nouvelles fenêtres qui seront disponibles dans l'interface graphique de GenSpec.

Pour la centralisation des informations, un arbre global représentera l'ensemble des exigences d'un projet, et ce éventuellement par l'intermédiaire de plusieurs racines dans l'arbre. Ce qui est un changement majeur par rapport à la situation actuelle. En effet, l'arbre hiérarchique des exigences actuel ne peut posséder qu'une seule racine. Cette situation est rendue possible par l'ajout d'une nouvelle entrée dans le menu contextuel relatif aux exigences. Coller au même niveau que l'exigence sélectionnée viendra donc se greffer aux entrées existantes.

Afin de représenter les différents regroupements créés par les utilisateurs, des nouveaux onglets seront présent, côte à côte avec l'arbre global. Chaque onglet contiendra donc au maximum un arbre hiérarchique.

Chaque regroupement possédant ses propres préférences, il est nécessaire de présenter toutes les options de génération existantes dans GenSpec à chacun de ces regroupements. Il en est de même avec la fenêtre de génération de documents. Chaque regroupement pouvant être généré dans un format de document qui lui est propre. Dès lors, une réorganisation des icônes de GenSpec est nécessaire afin d'agencer cette nouvelle organisation.

Pour l'attribution des exigences à des composants, de nouvelles fenêtres seront disponibles afin de réaliser les attributions proprement dites. Cette fenêtre sera accessible par l'intermédiaire du menu existant dans la fenêtre principale de GenSpec.

Enfin, l'accès aux options de génération du tableau implique une nouvelle entrée dans la liste des documents à générer.

## 3.4. Ordre de priorité des changements

Cette section priorise les nouveaux changements. Ceci est essentiel pour une prise de décision durant le processus de développement. Notons que le développement de ces nouvelles fonctionnalités va se dérouler par succession d'itérations. Nous classifions les fonctionnalités de haut niveau selon les trois catégories suivantes : essentielle, désirable, optionnelle.

Notons également que les premières itérations vont couvrir les exigences jugées essentielles, les suivantes, par contre, couvriront d'une part le reste des fonctionnalités, et d'autre part, raffineront les fonctionnalités développées dans les itérations antérieures.

### 3.4.1. Essentielle

- Utiliser une seule base de données qui enregistre l'ensemble des informations relatives aux exigences d'un projet. Et par conséquent, le fait qu'une même exigence n'est plus dupliquée au sein d'un même projet.

- Créer des regroupements d'exigences à l'intérieur d'un même projet GenSpec, les renommer, modifier, supprimer et générer des documents relatifs aux exigences présentes dans ces regroupements. Il s'agit de l'essence même de la gestion multi-documents de pouvoir gérer plusieurs regroupements, et ce afin de pouvoir les générer dans des documents distincts.
- La possibilité de générer un tableau d'attribution des exigences à des composants. Cela implique la réalisation de moyens simples et efficaces pour effectuer les attributions des exigences à des composants, ainsi que pour réaliser des regroupements fonctionnels. Il s'agit d'un besoin essentiel car il permettra d'automatiser une tâche jusqu'à présent réalisée manuellement.

#### 3.4.2. Désirable

- Disposer d'un document reprenant l'ensemble des exigences synchronisées (exigences répétées) d'un projet.

#### 3.4.3. Optionnelle

- Faire la correspondance entre les composants et fonctions du système avec leurs fichiers relatifs. Bien qu'elle soit optionnelle, il n'est cependant pas encore fixé si une telle réalisation est faisable sur le plan technique.

### 3.5. Classes d'utilisateurs affectées

Les classes d'utilisateurs au sein de GenSpec se distinguent par leurs privilèges en terme d'accès à des fonctionnalités (pour en savoir plus, consulter la rubrique d'aide de GenSpec [3]). Cependant, nous expliquerons ci-après les privilèges de l'administrateur, de l'utilisateur et du concepteur, ces trois types d'acteur étant affectés par la gestion multi-documents.

- **Administrateur** : il a accès à toutes les fonctionnalités de GenSpec. Notamment en ce qui concerne les nouvelles fonctionnalités relatives à la gestion multi-documents.
- **Utilisateur** : Il peut accéder aux mêmes fonctionnalités que l'administrateur, sauf pour les suivantes :
  - Génération de versions officielles ;
  - Ajout d'un usager ;
  - Retrait d'un usager ;
  - Modification du profil d'un autre usager ;
  - Changement du niveau d'accès.

Vis-à-vis de la gestion multi-documents, il peut créer des regroupements, les renommer, les supprimer et effectuer les opérations de déplacement d'exigences entre ces regroupements. Cependant, la fonctionnalité d'attribution des exigences à des composants ne lui est pas accessible.

- **Concepteur** : Ce type d'acteur n'existe actuellement pas au sein de GenSpec. Ce dernier pourra uniquement ajouter des commentaires aux exigences, attribuer des exigences à des composants et des fonctions et enfin générer dans un document le tableau d'attribution des exigences à des composants.

En conclusion, les deux parties de la gestion multi-documents ne sont pas accessibles à tous les types d'utilisateurs. En effet, pour la centralisation des informations, seuls l'utilisateur et l'administrateur y ont accès, alors que pour l'attribution des exigences à des composants, seuls l'administrateur et le concepteur peuvent en bénéficier.

### 3.6. Attribution des exigences à des composants

Elle doit permettre :

- De structurer les exigences en groupes fonctionnels ;

- D'attribuer les exigences à des composants de différents niveaux.

La solution proposée pour permettre une telle attribution est de créer un espace de composants listant tous les composants, et d'y attribuer des exigences à partir des regroupements. Étant donné que des composants peuvent se découper en sous composants, des niveaux de décomposition seront disponibles.

De plus, des regroupements fonctionnels pourront être créés. Excepté qu'ils ne se décomposent pas en niveaux comme les composants, les fonctions pourront également regrouper des exigences.

Une vérification permettra de mettre en évidence les composants ou fonctions ne contenant aucune exigence. Ceci permet alors à l'administrateur ou au concepteur de vérifier la cohérence des regroupements réalisés en composants et en fonctions.

### 3.6.1. Alternatives

- Les composants ou fonctions peuvent faire référence à des documents externes. Il peut être possible de créer des associations entre des composants ou fonctions et des fichiers.
- Une vérification des composants et fonctions permettra de déterminer d'éventuelles incohérences effectuées lors des attributions. Prenons pour exemple, un composant qui fait référence à un fichier qui n'existe pas. Il sera alors possible à l'utilisateur de GenSpec de corriger cette erreur.
- Des indices visuels peuvent être intégrés pour faciliter les attributions. Des indications mettront en évidence si les composants ou les fonctions possèdent des exigences ou des fichiers qui leur sont attribués ou non. Prenons par exemple la possibilité d'afficher les noms de composants et fonctions en gras lorsqu'elles ne contiennent pas d'exigences.
- Enfin, des options de génération permettront de paramétrer la génération du tableau. Ces options permettront de paramétrer entre autre l'ordre des colonnes pour le tableau à générer, le choix de regrouper les exigences générées par composants, de choisir les niveaux de composants à générer. De plus, elles permettront de choisir le format de document de sortie (Microsoft Word ou Excel), mais également la possibilité de nommer les niveaux de composants. Une gestion avancée pourrait fournir une liste de colonne prédéfinie à partir de laquelle il est possible de sélectionner les colonnes à générer dans le tableau.

Notons que les alternatives relatives à l'utilisation de fichiers restent incertaines. En effet, il n'est pas encore établi si une telle gestion des fichiers est faisable techniquement, étant donné le système utilisé pour gérer les fichiers. Par exemple, l'ouverture d'un fichier dans l'explorateur de Windows n'est peut-être pas réalisable de par l'utilisation d'applications spécifiques à Hydro-Québec pour gérer les fichiers.

### 3.7. Centralisation des informations

L'essence même de cette partie du projet de gestion multi-documents réside en la réalisation de regroupement d'exigences. Un projet GenSpec contiendra un arbre global. Cet arbre contiendra l'ensemble des exigences du projet sous forme de hiérarchie d'exigences. L'arbre global pourra contenir plusieurs regroupements d'exigences. C'est à partir de cet arbre qu'il sera possible de générer le document regroupant l'ensemble des exigences du système.

De nouveaux regroupements pourront être créés, un regroupement étant un arbre hiérarchique d'exigences. Le principe des regroupements est qu'une modification d'une exigence appartenant à un regroupement, a pour effet de modifier automatiquement les exigences de l'arbre global qui sont reliées au regroupement en question. De manière semblable, une modification dans l'arbre global, modifie également tous les regroupements présentant les exigences subissant la modification. Concernant les opérations possibles sur les regroupements, il sera bien sûr possible de supprimer des regroupements, de les renommer et enfin de les générer.



Afin de faciliter le processus pour utiliser une même exigence dans plusieurs regroupements, il sera possible de glisser des exigences d'un regroupement à l'autre. A cette fin, plusieurs fenêtres contenant les mêmes regroupements mais affichant des regroupements différents pourront être affichées simultanément pour permettre une consultation en parallèle des exigences. De plus, cette manière de procéder permettra de glisser des exigences d'une fenêtre à l'autre, d'un regroupement à l'autre.

Des exigences possédant toujours les mêmes valeurs sur certains champs pourront être synchronisées. Ce principe est déjà présent dans GenSpec. Cependant, de par l'utilisation d'un arbre global, cette fonction de synchronisation va être utilisée directement entre des exigences faisant partie de spécifications différentes, mais toutes représentées dans l'arbre global. Ceci n'est possible actuellement que par l'utilisation de la section *Autre fonctions*, en recréant à l'intérieur de cette dernière les exigences existantes dans les autres spécifications pour les rendre internes, et enfin en les synchronisant.

La gestion multi-documents permettra également d'améliorer la traçabilité entre les exigences du système et les exigences d'interfaces. En effet, un mécanisme de vérification permettra de mettre en évidence les exigences d'interfaces ne faisant aucun renvoi vers des exigences du système, et vice-versa.

En ce qui concerne la génération, chaque regroupement pourra faire l'objet d'un document séparé. Les options de génération de GenSpec, présentement applicables uniquement pour le projet GenSpec en question, pourront être paramétrées pour tous les regroupements à l'intérieur d'un seul projet GenSpec.

### 3.7.1. Alternatives

Après avoir présenté les fondements de la centralisation des informations, plusieurs alternatives peuvent être retenues afin de répondre aux besoins.

Les alternatives identifiées sont les suivantes :

- La possibilité de créer des regroupements à partir d'une racine de l'arbre global uniquement, ou pour être plus souple, à partir de n'importe quelle exigence.
- La possibilité de créer un regroupement selon différentes manières. Qu'il s'agisse d'un déplacement d'une exigence qui en permette la création automatique, ou encore la création par l'intermédiaire d'un menu contextuel.
- L'ajout automatique d'une section *Autre fonctions* qui regroupe les intrants et les extrants en provenance d'autres spécifications. Lors de la génération de l'arbre global, cette section devra disparaître. En effet, cette dernière n'a plus de rôle à jouer car l'ensemble des exigences est présent dans le même arbre hiérarchique. Par contre, dans un regroupement, cette section permet de lister toutes les exigences utilisées en entrée ou en sortie dans ce regroupement.
- L'ajout d'une vérification qui teste si une exigence de la SES est en rapport avec une exigence de la SEI, et inversement. En effet, des exigences de la SES doivent avoir un lien avec une exigence de la SEI. L'inverse est également vrai, car une exigence d'interface n'a aucune raison d'exister si elle n'est utilisée par aucune exigence de la SES.

## 4. RECOMMANDATIONS

Cette section justifie et motive les choix relatifs à la réalisation d'une gestion multi-documents pour GenSpec.

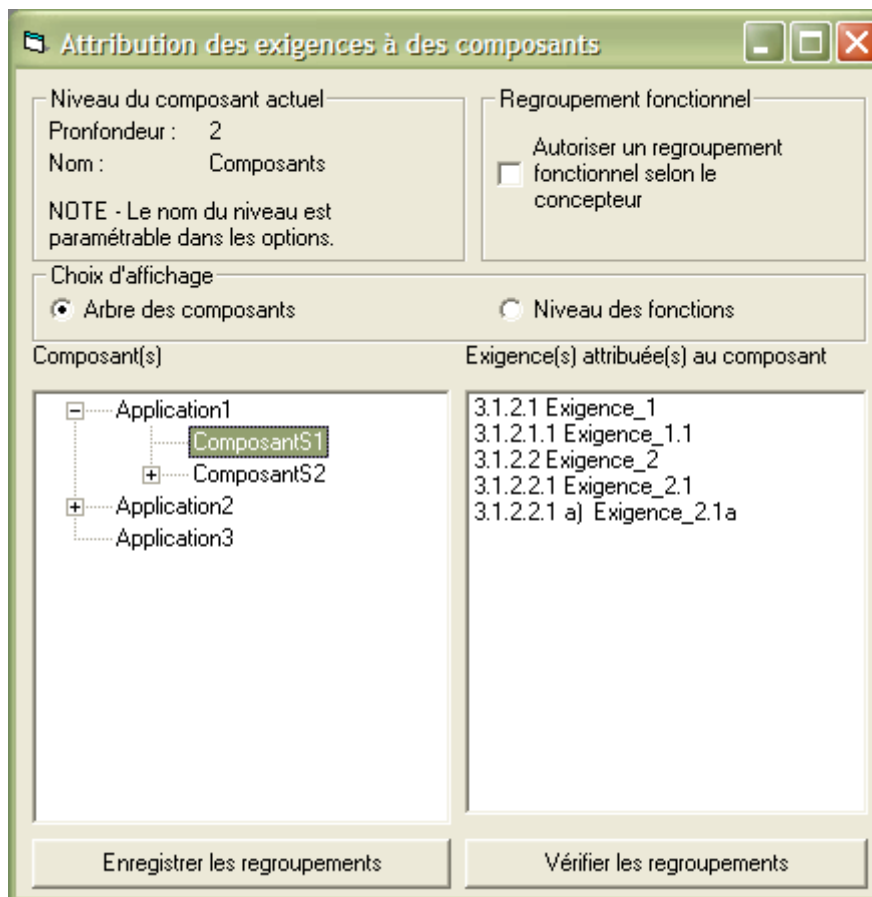
Il existe une multitude de manières de répondre aux besoins décrits ci-dessus. C'est donc dans cette optique, après avoir étudié les différentes alternatives possibles, qu'il faut choisir celles qui semblent les plus adaptées.

Dans les parties suivantes, nous décrivons les choix adoptés pour la réalisation d'une gestion multi-documents. Afin d'être le plus explicite possible avec les choix retenus, les captures d'écrans

retenues vont être systématiquement commentées. Enfin, notons que les alternatives retenues ont fait l'objet d'évolutions successives en accord avec les besoins des utilisateurs de GenSpec. Plusieurs captures d'écrans viennent illustrer la philosophie de l'alternative choisie, bien qu'elles ne soient pas pour autant définitives.

#### 4.1. Attribution des exigences à des composants

##### 4.1.1. Fenêtre des composants avec les exigences qui y sont attribuées



**Figure 1 - Exigences attribuées à ComposantS2 au deuxième niveau de composants**

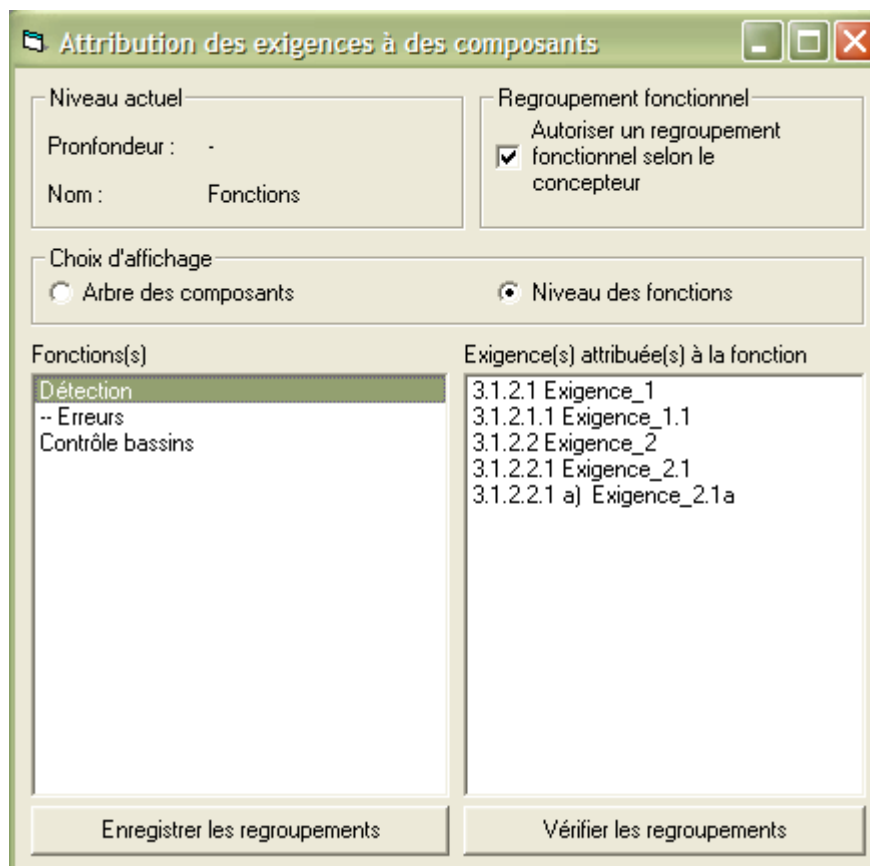
La navigation dans les différents niveaux de composants se fait de manière analogue à celle utilisée dans l'arbre hiérarchique des exigences.

La liste des composants et fonctions avec la liste des exigences attribuées et les fichiers associés fait penser fortement à la manière d'organiser les exigences générées et applicables, fonctionnalité existante de GenSpec. De plus, les opérations sur les composants ou les fonctions telles que l'ajout, la suppression ou la modification du nom, sont accessibles par des menus contextuels, également de manière semblable aux exigences générées et applicables. Ces ressemblances sont appréciées car elles permettent aux utilisateurs de GenSpec de retrouver une interface graphique qui leur est familière.

L'enregistrement des données en base de données permet une réutilisation future des attributions effectuées. La vérification permet de détecter d'éventuels oublis lors de l'attribution

Enfin, un cadre informatif donne un aperçu sur le niveau actuel du composant ainsi que le nom de ce niveau. Le choix d'affichage permet de sélectionner la vue que l'on souhaite montrer au premier plan, c'est-à-dire le niveau des fonctions ou l'arbre des composants.

#### 4.1.2. Fenêtre des fonctions avec les exigences qui y sont attribuées

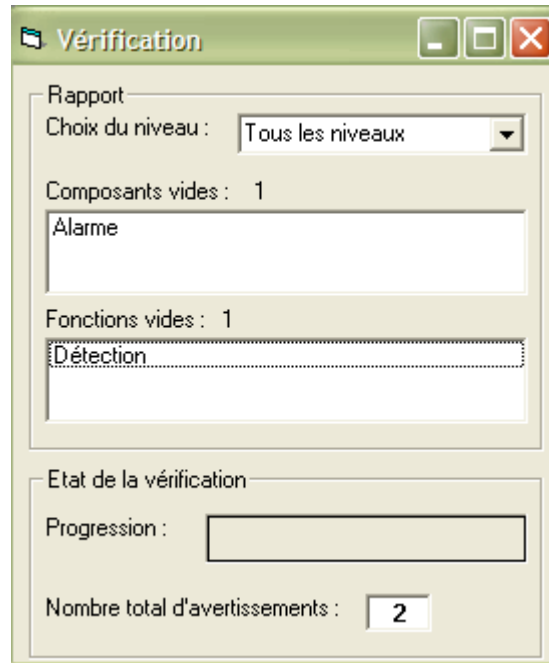


**Figure 2 - Exigences assignées à la fonction Détection**

L'attribution des exigences à des fonctions se réalise de manière similaire à l'attribution à des composants. Notons cependant que les fonctions utilisent une structure en liste et non en hiérarchie. Les fonctions n'ont donc qu'un seul niveau, le niveau des fonctions.

Un regroupement fonctionnel selon le concepteur est disponible, ce dernier indiquant que les fonctions définies (dans l'exemple : Détection, erreurs, contrôles bassins) seront associées aux exigences qui leurs sont attribuées lors de la génération du tableau incluant les exigences en question.

### 4.1.3. Vérification des regroupements



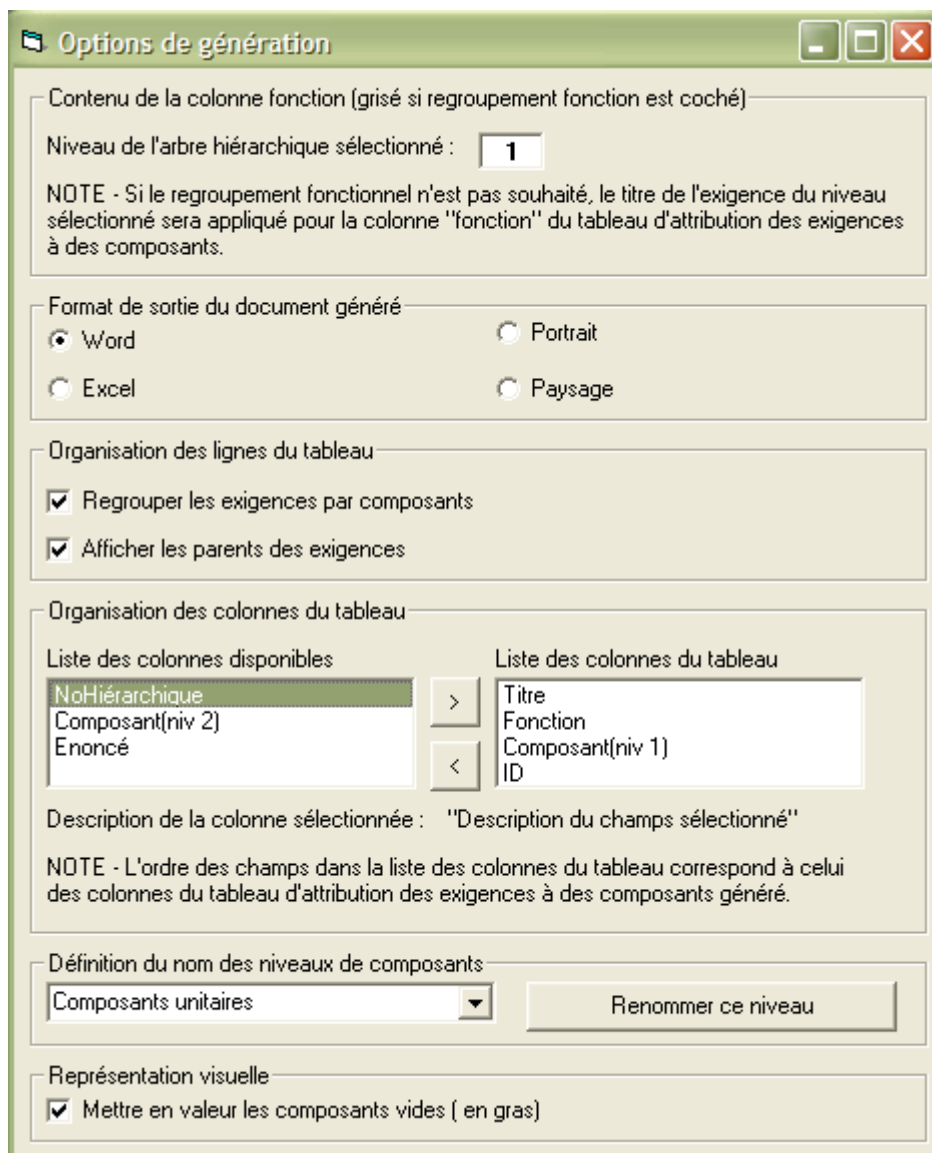
**Figure 3 - Vérification des composants et des fonctions**

Une vérification permet de mettre en évidence d'éventuelles incohérences. Elles sont alors comptées et listées.

La recherche est paramétrable en fonction des niveaux de composants et selon le niveau de fonctions.

Le listage permet d'atteindre les composants ou fonctions qui suscitent les avertissements.

#### 4.1.4. Options de génération



**Options de génération**

Contenu de la colonne fonction (grisé si regroupement fonction est coché)

Niveau de l'arbre hiérarchique sélectionné :

NOTE - Si le regroupement fonctionnel n'est pas souhaité, le titre de l'exigence du niveau sélectionné sera appliqué pour la colonne "fonction" du tableau d'attribution des exigences à des composants.

Format de sortie du document généré

Word  Portrait

Excel  Paysage

Organisation des lignes du tableau

Regrouper les exigences par composants

Afficher les parents des exigences

Organisation des colonnes du tableau

Liste des colonnes disponibles

NoHiérarchique
Composant(niv 2)
Enoncé

Liste des colonnes du tableau

Titre
Fonction
Composant(niv 1)
ID

Description de la colonne sélectionnée : "Description du champs sélectionné"

NOTE - L'ordre des champs dans la liste des colonnes du tableau correspond à celui des colonnes du tableau d'attribution des exigences à des composants généré.

Définition du nom des niveaux de composants

Représentation visuelle

Mettre en valeur les composants vides ( en gras)

**Figure 4 - Options de génération du tableau**

Les options de génération retenues sont les suivantes :

- La définition du niveau d'exigence si un regroupement fonctionnel selon le concepteur n'est pas souhaité, en effet, le titre du parent peut suffire pour définir la fonction que possède une exigence;
- Le format de sortie du document, en fonction des besoins des utilisateurs de GenSpec ;
- Les différentes manières d'organiser les lignes du tableau. Regrouper les exigences par composants permet d'avoir une vue groupée des exigences en composants. Enfin, afficher les parents de l'exigence permet de retrouver immédiatement le contexte d'une exigence en observant le titre de son parent ;
- Le choix des colonnes à afficher pour le tableau. Une liste de colonnes est disponible préalablement, il reste alors à sélectionner les colonnes à générer en fonction des besoins.

Cela donne une grande flexibilité quant à la représentation des informations contenues dans le tableau généré ;

- La définition du nom pour chacun des niveaux de composants permettant de se mettre d'accord sur une terminologie commune ;
- La possibilité de mettre en évidence les composants ou fonctions vides afin de détecter des éventuels oublis lors des attributions.

## 4.2. Centralisation des informations

La solution à adopter pour arriver à ce résultat est décrite dans le paragraphe décrivant la centralisation des informations dans le chapitre "Propositions de solutions".

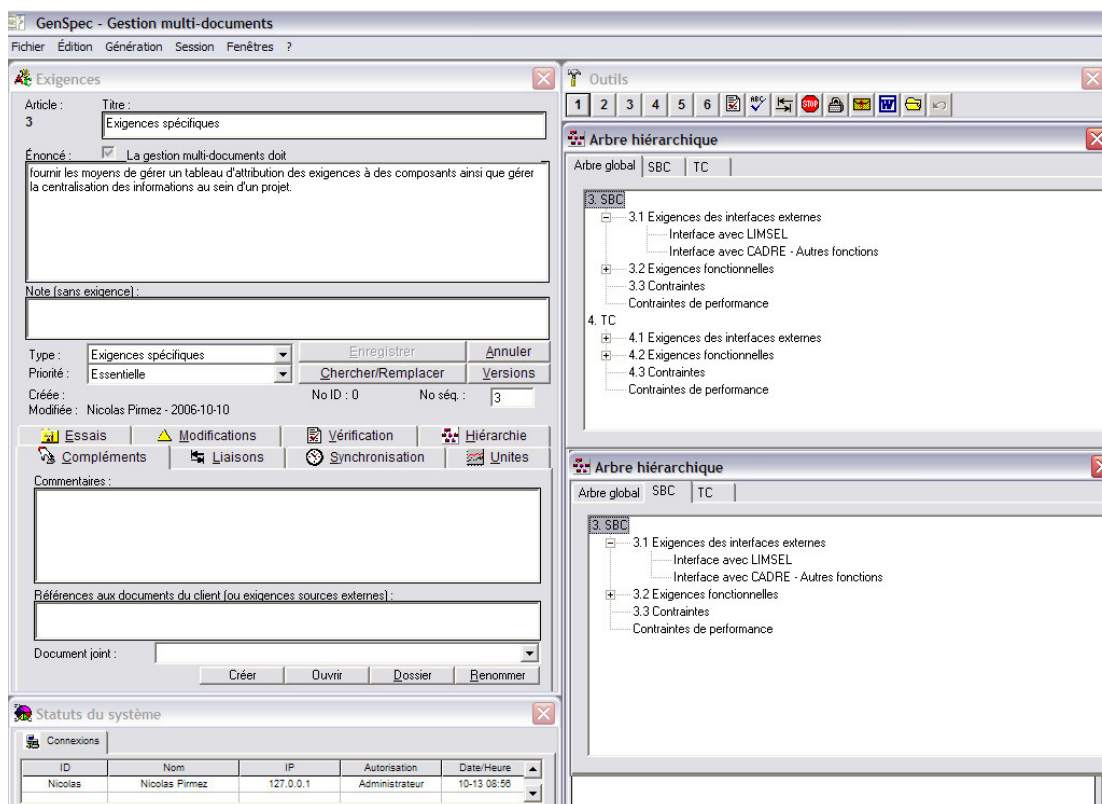


Figure 5 - Exemple d'arbres hiérarchiques

Rappelons que l'essence de la gestion multi-documents et de gérer l'ensemble des exigences dans un seul projet GenSpec et dans une seule base de données. La figure ci-dessous montre un agencement possible des fenêtres reprenant les onglets des différents regroupements.

### 4.2.1. Création et liaisons de regroupements

Dans cette illustration, deux fenêtres de regroupements sont affichées avec deux regroupements d'exigences (en plus de l'arbre global). Cette solution semble être la plus adaptée à la comparaison et au déplacement d'exigences entre les regroupements. Dans l'illustration ci-dessus, l'utilisateur de GenSpec peut travailler sur les exigences relatives à SBC dans le regroupement nommé du même nom. Cela lui permet de ne pas être encombré par des exigences qui n'ont pas de rapport direct avec son travail.

#### **4.2.2. Options de génération**

En plus de l'arbre global, chaque regroupement peut faire l'objet d'une génération de documents. En effet, la génération de l'arbre global permet de représenter l'ensemble des exigences d'un projet dans un seul et même document. Par contre, la génération d'un regroupement permet d'obtenir un document présentant uniquement les exigences contenues dans le regroupement en question.

Dès lors, des options de génération seront disponibles pour chacun des regroupements. Pour ce faire, la création d'un regroupement entrainera automatiquement la création d'un onglet regroupant le panel d'options actuellement disponible dans GenSpec (ce dernier ne s'appliquant alors qu'au regroupement auquel il fait référence). Cela permettra de paramétrer les options de génération pour un regroupement et ce, indépendamment des autres.



## 5. ANALYSE DU SYSTÈME PROPOSÉ

Cette section fournit une analyse des bénéfices, limites, avantages et inconvénients de l'ajout de fonctionnalités relatives à une gestion multi-documents pour GenSpec

### 5.1. Sommaire des améliorations

Un résumé des améliorations résultant de la réalisation de la gestion multi-documents est expliqué ci-dessous.

#### 5.1.1. Attribution des exigences à des composants

- Elle permettra de réaliser une nouvelle tâche, jusqu'alors réalisée manuellement ;
- Elle permettra de générer un nouveau type de document utilisé par le concepteur ;
- Étant donné qu'il sera possible de nommer les différents niveaux de composants, cela fixera un vocabulaire commun sur les types de composants utilisés. Par exemple, si le premier niveau de composant est nommé *Applications*, les utilisateurs qui parleront d'applications sauront qu'il s'agit de composants de premier niveau, et donc de haut niveau en l'occurrence.
- Enfin, elle permettra d'améliorer la traçabilité entre les composants et les exigences qui y sont attribuées.

#### 5.1.2. Centralisation des informations

- Elle donnera la possibilité de gérer l'ensemble des exigences à l'intérieur d'un seul projet GenSpec, qu'il s'agisse des exigences de la SES ou de la SEI ;
- Toutes les informations seront centralisées dans une seule base de données ;
- L'arbre global pourra posséder plusieurs racines, donnant la possibilité de gérer les exigences plus simplement. En effet, contrairement à la situation actuelle, toutes les exigences seront accessibles dans un même arbre hiérarchique, à savoir l'arbre global des exigences ;
- Il sera plus aisé de déplacer ou copier des exigences. En effet, de par la possibilité d'afficher plusieurs fenêtres de regroupements en parallèle, cela permettra d'effectuer des opérations sur les exigences et ce, tout en conservant une vue de la source et de la cible relative au changement ;
- Elle donnera la possibilité de créer, renommer et supprimer des regroupements, ces derniers permettant de travailler sur un groupe d'exigence défini ;
- Elle donnera la possibilité de générer des documents, notamment l'arbre global qui contient l'ensemble des exigences ;
- L'ajout d'une vérification agissant sur les relations entre les exigences de la SES et de la SEI. Cela permettra de mettre en évidence les exigences éventuelles qui seraient source d'incohérences ;
- Elle donnera la possibilité d'importer des exigences existantes d'anciens projets GenSpec dans un nouveau, et ce grâce à la possibilité de coller des exigences au niveau de la racine de l'arbre global, ce qui n'est présentement pas possible.
- Enfin, de par son caractère centralisé, elle donnera la possibilité de réaliser des liens entre des exigences appartenant à des spécifications différentes et ce, de manière dynamique.

### 5.2. Désavantages et limites

Cette sous-section énumère les désavantages et limites auxquels la gestion multi-documents sera confrontée.

### **5.2.1. Attribution des exigences à des composants**

Les colonnes qu'il sera possible de générer dans le tableau sont définies à priori. Le choix est donc restreint à ces colonnes.

### **5.2.2. Centralisation des informations**

Bien que cela ne soit pas lourd de conséquences, l'intégration des nouvelles fonctionnalités va impliquer que les utilisateurs de GenSpec devront intégrer les nouveaux concepts. De plus, la manière dont ils organisent leur projet - en plusieurs projets GenSpec - va être radicalement modifiée.



## **Annexe E**

# **Exigences : Glossaire**

La dernière modification du contenu de ce document date du 9 janvier 2007.

*Exigences du glossaire*

**GenSpec**

**Glossaire**

**OCTOBRE 2006**

## 1. EXIGENCES SPÉCIFIQUES

### 1.1. Exigences fonctionnelles

- 1.1.1. Définition d'un terme
  - 1.1.1.1. Interdictions
    - 1.1.1.1.1. Interdiction d'avoir 1 terme définis de plusieurs manières différentes
    - 1.1.1.1.2. Interdiction de définir un terme apparaissant dans une liste de synonymes
    - 1.1.1.1.3. Interdiction d'avoir 2 termes de définitions identiques
  - 1.1.1.2. Vérifications
    - 1.1.1.2.1. Vérification de la grammaire
    - 1.1.1.2.2. Vérification d'orthographe
    - 1.1.1.2.3. Vérification des termes subjectifs
    - 1.1.1.2.4. Terme subjectif
      - 1.1.1.2.4.a) Ajout d'un terme subjectif
      - 1.1.1.2.4.b) Modification d'un terme subjectif
      - 1.1.1.2.4.c) Suppression d'un terme subjectif
  - 1.1.1.3. Forme d'un terme
    - 1.1.1.3.1. Terme vide
    - 1.1.1.3.2. Terme simple
    - 1.1.1.3.3. Terme composé
  - 1.1.1.4. Définition d'un terme non défini
    - 1.1.1.4.1. Définition du terme
    - 1.1.1.4.2. Version de la définition du terme
      - 1.1.1.4.2.a) Numéro de version de la définition
      - 1.1.1.4.2.b) Auteur de la définition
      - 1.1.1.4.2.c) Date d'ajout/de modification de la définition
    - 1.1.1.4.3. Propriétés du terme
    - 1.1.1.4.4. But(s) du terme pour l'acteur
      - 1.1.1.4.4.a) But(s) pour l'acteur implicite(s)
    - 1.1.1.4.5. But(s) du terme pour le système
      - 1.1.1.4.5.a) But(s) pour le système implicite(s)
    - 1.1.1.4.6. Liste de synonyme(s) du terme
      - 1.1.1.4.6.a) Liste de synonymes vide
      - 1.1.1.4.6.b) Possibilité de définir une liste vide
      - 1.1.1.4.6.c) Composition
    - 1.1.1.4.7. Exigence(s) associée(s) au terme
    - 1.1.1.4.8. Niveau d'ambiguïté de la définition du terme
      - 1.1.1.4.8.a) Première définition de l'ambiguïté
      - 1.1.1.4.8.b) Échelle non numérique
    - 1.1.1.4.9. Information(s) complémentaire(s)
    - 1.1.1.4.10. Historique des modifications de la définition du terme
      - 1.1.1.4.10.a) Initialisation
      - 1.1.1.4.10.b) Informations sur la version d'un historique
      - 1.1.1.4.10.c) Informations sur le terme retenues dans l'historique
      - 1.1.1.4.10.d) Justification de modification
      - 1.1.1.4.10.e) Numéro de version de l'historique
  - 1.1.1.5. Définition d'un terme défini
    - 1.1.1.5.1. Généralisation de modifications
      - 1.1.1.5.1.a) Avertissement de modification
      - 1.1.1.5.1.b) Vérification de la grammaire
    - 1.1.1.5.2. Explication de la modification
      - 1.1.1.5.2.a) Justification de modification pour l'historique
    - 1.1.1.5.3. Modification d'un terme
      - 1.1.1.5.3.a) Utilisation d'un synonyme
    - 1.1.1.5.4. Modification concernant l'ensemble des informations d'un terme
  - 1.1.1.6. Suppression de définition
    - 1.1.1.6.1. Suppression de l'ensemble des informations d'un terme
    - 1.1.1.6.2. Suppression fictive
      - 1.1.1.6.2.a) Informations des termes supprimés non accessibles pour tous les utilisateurs
      - 1.1.1.6.2.b) Informations des termes supprimés accessibles par l'administrateur
      - 1.1.1.6.2.c) Affichage distinctif des termes supprimés
  - 1.1.1.7. Termes prioritaires
    - 1.1.1.7.1. Consultation des termes prioritaires
      - 1.1.1.7.1.a) Filtrage des termes prioritaires
      - 1.1.1.7.1.b) Tri des termes prioritaires
      - 1.1.1.7.1.c) Affichage du nombre d'occurrences
      - 1.1.1.7.1.d) Différentes formes d'un terme prioritaire
    - 1.1.1.7.2. Définition d'un terme prioritaire
      - 1.1.1.7.2.a) Suppression du terme prioritaire
- 1.1.2. Gestion Intrans / Extrant
  - 1.1.2.1. Définition à partir du glossaire
    - 1.1.2.1.1. Définition d'un intrans à partir du glossaire
    - 1.1.2.1.2. Définition d'un extrant à partir du glossaire
  - 1.1.2.2. Définition à partir de l'arbre hiérarchique

- 1.1.2.2.1. Définition d'un intrant à partir de l'arbre hiérarchique
- 1.1.2.2.2. Définition d'un extrant à partir de l'arbre hiérarchique
- 1.1.2.3. Ajout à l'arbre hiérarchique
  - 1.1.2.3.1. Ajout d'intrants à l'arbre hiérarchique
    - 1.1.2.3.1.a) Respect des contraintes de la base de données vis-à-vis des intrants
    - 1.1.2.3.1.b) Terme non défini comme intrant
    - 1.1.2.3.1.c) Définir un terme comme intrant
    - 1.1.2.3.1.d) Énoncé équivalent au titre de l'exigence
  - 1.1.2.3.2. Ajout d'extrants à l'arbre hiérarchique
    - 1.1.2.3.2.a) Respect des contraintes de la base de données vis-à-vis des extrants
    - 1.1.2.3.2.b) Terme non défini comme extrant
    - 1.1.2.3.2.c) Définir le terme comme extrant
    - 1.1.2.3.2.d) Énoncé équivalent au titre de l'exigence
  - 1.1.2.3.3. Synchronisation entre le glossaire et les exigences
    - 1.1.2.3.3.a) Synchronisation vers le glossaire
    - 1.1.2.3.3.b) Synchronisation vers l'arbre hiérarchique
    - 1.1.2.3.3.c) Synchronisation entre les exigences
- 1.1.2.4. Propriétés
  - 1.1.2.4.1. Double propriété : Intrant et extrant
  - 1.1.2.4.2. Aucune propriété : ni intrant ni extrant
- 1.1.2.5. Affichage des exigences
  - 1.1.2.5.1. Exigences où le terme constitue un intrant
  - 1.1.2.5.2. Exigences où le terme constitue un extrant
- 1.1.2.6. Suppression à partir l'arbre hiérarchique
  - 1.1.2.6.1. Suppression d'un intrant à partir de l'arbre hiérarchique
    - 1.1.2.6.1.a) Avertissement de suppression d'intrant
  - 1.1.2.6.2. Suppression d'un extrant à partir de l'arbre hiérarchique
    - 1.1.2.6.2.a) Avertissement de suppression d'extrant
- 1.1.2.7. Non-défini
  - 1.1.2.7.1. Intrant non-défini au sein de l'arbre hiérarchique
  - 1.1.2.7.2. Extrant non-défini au sein de l'arbre hiérarchique
  - 1.1.2.7.3. Termes définis soulignés
- 1.1.3. Gestion des informations
  - 1.1.3.1. Consultation des informations
    - 1.1.3.1.1. Consultation d'un terme à partir du glossaire
    - 1.1.3.1.2. Consultation d'un terme à partir d'une exigence
    - 1.1.3.1.3. Consultation de la liste des termes définis
    - 1.1.3.1.4. Consultation de l'ensemble des informations associées à un terme
  - 1.1.3.2. Affichage des informations
    - 1.1.3.2.1. Affichage des informations au sein de la fenêtre du glossaire
    - 1.1.3.2.2. Affichage de l'historique
      - 1.1.3.2.2.a) Liste des versions de la définition
      - 1.1.3.2.2.b) Justification de la modification
      - 1.1.3.2.2.c) Filtre
    - 1.1.3.2.3. Affichage des exigences
    - 1.1.3.2.4. Affichage des termes définis dans une exigence
    - 1.1.3.2.5. Affichage de la définition d'un terme directement à partir de la fenêtre des exigences
    - 1.1.3.2.6. Affichage des termes ambigus
  - 1.1.3.3. Recherche au sein du glossaire
    - 1.1.3.3.1. Recherche selon le terme
    - 1.1.3.3.2. Recherche selon un ensemble de critères
    - 1.1.3.3.3. Gestion des différentes formes des termes
- 1.1.4. Utilisation
  - 1.1.4.1. Utilisation dans une exigence
    - 1.1.4.1.1. Utiliser un terme défini dans une exigence
    - 1.1.4.1.2. Interdiction d'utiliser un synonyme
      - 1.1.4.1.2.a) Message informatif
      - 1.1.4.1.2.b) Choix direct
      - 1.1.4.1.2.c) Vérification à postériori
    - 1.1.4.1.3. Terme définis soulignés
- 1.1.5. Gestion courante
  - 1.1.5.1. Base de données
    - 1.1.5.1.1. Base de données précédentes
  - 1.1.5.2. Importation de glossaire(s)
    - 1.1.5.2.1. Choix entre synonymes
      - 1.1.5.2.1.a) Choix pour tous les synonymes
      - 1.1.5.2.1.b) Comparaison
      - 1.1.5.2.1.c) Choix unique
    - 1.1.5.2.2. Termes supprimés
  - 1.1.5.3. Génération d'un document
    - 1.1.5.3.1. Tableau
    - 1.1.5.3.2. Informations reprises

- 1.1.5.3.2.a) Définition
- 1.1.5.3.2.b) But pour l'acteur
- 1.1.5.3.2.c) But pour le système
- 1.1.5.3.2.d) Champs implicites
- 1.1.5.3.3. Ordre
- 1.1.5.3.4. Glossaire partiel
- 1.1.5.3.5. Informations partielles
- 1.1.5.3.6. Options de génération
  - 1.1.5.3.6.a) Totalité du glossaire
  - 1.1.5.3.6.b) Termes définis distincts
  - 1.1.5.3.6.c) Glossaire au sein du document des exigences
  - 1.1.5.3.6.d) Affichage du quadrillage
  - 1.1.5.3.6.e) Affichage du nom des colonnes
- 1.1.5.4. Gestion générale du glossaire
  - 1.1.5.4.1. Choix de fonctionnalités par l'administrateur
    - 1.1.5.4.1.a) Obligation d'utiliser le glossaire
    - 1.1.5.4.1.b) Informations obligatoires
  - 1.1.5.4.2. Menu d'aide
    - 1.1.5.4.2.a) Ajout au menu d'aide actuel
    - 1.1.5.4.2.b) Consultation
- 1.1.5.5. Propriétés
  - 1.1.5.5.1. Affichage des termes définis
  - 1.1.5.5.2. Un glossaire par projet
    - 1.1.5.5.2.a) Maximum
    - 1.1.5.5.2.b) Glossaire vide
- 1.1.5.6. Contrainte(s)
  - 1.1.5.6.1. Langage



## 1. EXIGENCES SPÉCIFIQUES

Le Glossaire de GenSpec doit permettre de réduire l'ambiguïté des spécifications, une meilleure gestion des intrants et des extrants et de définir un ensemble d'informations permettant une meilleure compréhension des spécifications réalisées par l'utilisation de GenSpec. (exig. # 0)

Il en découle les exigences suivantes :

### 1) Exigences fonctionnelles

Réf. besoin client : Modèle d'objectifs et Cas d'utilisation – Glossaire pour GenSpec – (Diagramme de Cas d'utilisation – page 7).

*Commentaire : Au sein de des exigences, lorsque l'on fera référence à une catégorie d'utilisateur, on fera également référence aux catégories supérieures conformément à la hiérarchie des catégories d'utilisateurs. De cette manière, cela permet de définir de manière implicite les droits d'accès des différents utilisateurs vis-à-vis de l'ensemble des exigences.*

### **1.1. Exigences fonctionnelles**

Le Glossaire de GenSpec doit exécuter les fonctions requises, en réaction aux intrants et pour soutenir ses extrants. (exig. # 146)

Il en découle les exigences suivantes :

- 1) Définition d'un terme
- 2) Gestion Intrans / Extrans
- 3) Gestion des informations
- 4) Utilisation
- 5) Gestion courante

### 1.1.1. Définition d'un terme

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un terme au sein du glossaire. (exig. # 1844)

Il en découle les exigences suivantes :

- 1) Interdictions
- 2) Vérifications
- 3) Forme d'un terme
- 4) Définition d'un terme non défini
- 5) Définition d'un terme défini
- 6) Suppression de définition
- 7) Termes prioritaires

NOTE – La définition correspondant ici à l'ensemble des informations relatives à un terme et non pas à une définition succincte du sens/concept du terme employé (sauf exception précisée).

#### 1.1.1.1. Interdictions

Le Glossaire de GenSpec doit respecter plusieurs propriétés vis-à-vis de la définition des termes au sein du glossaire. (exig. # 1962)

##### 1.1.1.1.1. Interdiction d'avoir 1 terme définis de plusieurs manières différentes

Le Glossaire de GenSpec doit limiter le nombre de définitions d'un terme à une seule occurrence. (exig. # 1879)

NOTE – 1 mot = 1 sens.

##### 1.1.1.1.2. Interdiction de définir un terme apparaissant dans une liste de synonymes

Le Glossaire de GenSpec doit interdire la définition d'un terme au sein du glossaire lorsque ce terme apparaît déjà dans une liste de synonymes d'un autre terme du glossaire. (exig. # 1875)

NOTE – Il est impossible d'avoir un terme qui soit présent au sein de plus d'une liste de synonymes. Sinon, cela impliquerait que 2 termes définis au sein du glossaire soit identiques.

##### 1.1.1.1.3. Interdiction d'avoir 2 termes de définitions identiques

Le Glossaire de GenSpec doit assurer l'unicité de la définition d'un terme au sein du glossaire. (exig. # 1876)

NOTE – La définition correspondant à l'ensemble des informations concernant un terme.

1 sens = 1 mot.

#### 1.1.1.2. Vérifications

Le Glossaire de GenSpec doit permettre d'effectuer, si l'utilisateur le souhaite, un ensemble de vérifications. Ces vérifications doivent être associée au menu actuel de vérification. (exig. # 2029)

#### **1.1.1.2.1. Vérification de la grammaire [C]**

Le Glossaire de GenSpec doit, si l'utilisateur le souhaite, vérifier la grammaire vis-à-vis de l'ensemble des informations d'un terme au sein du glossaire. (exig. # 2030)

#### **1.1.1.2.2. Vérification d'orthographe [C]**

Le Glossaire de GenSpec doit, si l'utilisateur le souhaite, vérifier l'orthographe d'un terme et de l'ensemble des informations de ce terme au sein du glossaire. (exig. # 1881)

#### **1.1.1.2.3. Vérification des termes subjectifs**

Le Glossaire de GenSpec doit permettre, si l'utilisateur le souhaite, de vérifier la présence, au sein des exigences, de termes subjectifs. (exig. # 2053)

NOTE – Un terme subjectif correspond à un terme provoquant à lui seul de l'ambiguïté au sein de n'importe quel exigence. Exemple : correct, fiable, etc.

#### **1.1.1.2.4. Terme subjectif**

Le Glossaire de GenSpec doit permettre la gestion des termes subjectifs par le biais d'une fenêtre associée à cette fonctionnalité : (exig. # 2054)

- a) Ajout d'un terme subjectif. (exig. # 2055) Le glossaire de GenSpec doit permettre à l'utilisateur, par le biais de fenêtre des termes subjectifs, d'ajouter un terme au sein de la liste des termes subjectifs ; (exig. # 2055)
- b) Modification d'un terme subjectif. (exig. # 2057) Le glossaire de GenSpec doit permettre la modification, par l'utilisateur, par le biais de fenêtre des termes subjectifs, d'un terme dans la liste des termes subjectifs ; (exig. # 2057)
- c) Suppression d'un terme subjectif. (exig. # 2056) Le glossaire de GenSpec doit permettre la suppression, par l'utilisateur, par le biais de fenêtre des termes subjectifs, d'un terme au sein de la liste des termes subjectifs. (exig. # 2056)

NOTE – Un terme subjectif correspond à un terme provoquant à lui seul de l'ambiguïté au sein de n'importe quel exigence. Exemple : correct, fiable, etc.

#### **1.1.1.3. Forme d'un terme**

Le Glossaire de GenSpec doit permettre à l'utilisateur d'ajouter des termes sous différentes formes – simple et composé – au sein du glossaire. (exig. # 2067)

##### **1.1.1.3.1. Terme vide**

Le Glossaire de GenSpec doit interdire à l'utilisateur de fournir une définition au glossaire sans aucun terme. (exig. # 2070)

##### **1.1.1.3.2. Terme simple**

Le Glossaire de GenSpec doit permettre à l'utilisateur d'ajouter un terme au sein du glossaire qui ne soit pas composé. (exig. # 2068)

NOTE – Un terme composé est un terme au sein du glossaire qui est composé de plusieurs mots. Exemple : "bassin de délestage".

#### 1.1.1.3.3. Terme composé

Le Glossaire de GenSpec doit permettre à l'utilisateur d'ajouter un terme au sein du glossaire qui soit composé. (exig. # 2069)

NOTE – Un terme composé est un terme au sein du glossaire qui est composé de plusieurs mots. Exemple : "bassin de délestage".

#### 1.1.1.4. Définition d'un terme non défini

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un terme non défini au sein du glossaire à partir d'une exigence ou directement à partir du glossaire. Voir références internes ci après identifiées pour le respect des propriétés ainsi que pour établir les vérifications à effectuer. (exig. # 1850)

Réf. interne : 1.1.1.1, 1.1.1.2.

##### 1.1.1.4.1. Définition du terme

Le Glossaire de GenSpec doit demander à l'utilisateur de fournir la définition du terme. (exig. # 1864)

NOTE – La définition du terme correspondant à la sémantique qui est associé au terme employé.

##### 1.1.1.4.2. Version de la définition du terme

Le Glossaire de GenSpec doit générer automatiquement les informations de version de la définition d'un terme : (exig. # 1865)

- a) Numéro de version de la définition. (exig. # 2005) Le glossaire de GenSpec doit générer automatiquement le numéro de version d'une définition d'un terme ; (exig. # 2005)
- b) Auteur de la définition. (exig. # 2006) Le glossaire de GenSpec doit retenir qui est l'auteur de la définition (sans retenir le statut de l'auteur) ; (exig. # 2006) NOTE – Le statut correspond au type d'utilisateur vis-à-vis de GenSpec. Exemple : Administrateur, utilisateur, etc.
- c) Date d'ajout/de modification de la définition. (exig. # 2007) Le glossaire de GenSpec doit retenir quand la définition a été ajoutée/modifiée. (exig. # 2007)

NOTE – L'ajout au sein du glossaire correspond à la première version.

##### 1.1.1.4.3. Propriétés du terme

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir les propriétés du terme. Voir référence interne ci après identifiée pour voir quelles sont les propriétés à établir. (exig. # 1866)

Réf. interne : 1.1.2.1.

##### 1.1.1.4.4. But(s) du terme pour l'acteur

Le Glossaire de GenSpec doit permettre à l'utilisateur d'indiquer le(s) but(s) du terme pour l'acteur (humain ou autre système) : (exig. # 1867)

- a) But(s) pour l'acteur implicite(s). (exig. # 2008) Le glossaire de GenSpec doit permettre à l'utilisateur de définir un but pour un acteur comme étant implicite. (exig. # 2008)

#### 1.1.1.4.5. But(s) du terme pour le système

Le Glossaire de GenSpec doit permettre à l'utilisateur d'indiquer le(s) but(s) du terme pour le système : (exig. # 1868)

- a) But(s) pour le système implicite(s). (exig. # 2009) Le glossaire de GenSpec doit permettre à l'utilisateur de définir un but pour le système comme étant implicite. (exig. # 2009)

#### 1.1.1.4.6. Liste de synonyme(s) du terme

Le Glossaire de GenSpec doit permettre à l'utilisateur de fournir une liste de synonymes au terme défini : (exig. # 1869)

- a) Liste de synonymes vide. (exig. # 2010) La liste de synonymes peut être vide dans le cas où l'utilisateur ne connaît pas de synonymes du terme défini ou s'il n'existe pas de synonymes de ce terme ; (exig. # 2010)
- b) Possibilité de définir une liste vide. (exig. # 2083) Le glossaire de GenSpec doit permettre à l'utilisateur de marquer le fait qu'un terme ne possède pas de synonymes – onglet à cocher : "Aucun synonymes correspondants" – ; (exig. # 2083)
- c) Composition. (exig. # 2066) Un synonyme ne peut être qu'un seul terme ou il peut être composé de plusieurs termes. (exig. # 2066)

NOTE – Cette liste contribuera à la réalisation de la propriété 1 terme = 1 sens ; 1 sens = 1 terme.

#### 1.1.1.4.7. Exigence(s) associée(s) au terme

Le Glossaire de GenSpec doit permettre d'afficher automatiquement à l'utilisateur, au sein de la fenêtre du glossaire, l'ensemble des exigences où le terme défini apparaît. Voir référence interne ci après identifiée pour les cas intrants/extrants. (exig. # 1870)

Réf. interne : 1.1.2.5.

#### 1.1.1.4.8. Niveau d'ambiguïté de la définition du terme

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir le niveau d'ambiguïté ainsi que la justification d'une telle évaluation au sein de la définition d'un terme du glossaire : (exig. # 1873)

- a) Première définition de l'ambiguïté. (exig. # 2017) Le glossaire de GenSpec doit permettre à l'auteur de la définition de définir l'ambiguïté de la définition qu'il vient de fournir lors de l'ajout du terme au glossaire ; (exig. # 2017)
- b) Échelle non numérique. (exig. # 2047) Le glossaire de GenSpec doit fournir une échelle non numérique au visiteur afin d'évaluer l'ambiguïté d'un terme défini au sein du glossaire. (exig. # 2047)

#### 1.1.1.4.9. Information(s) complémentaire(s)

Le Glossaire de GenSpec doit permettre à l'utilisateur de fournir l'ensemble des informations complémentaires qu'il juge comme étant importantes/nécessaires. (exig. # 1874)

#### 1.1.1.4.10. Historique des modifications de la définition du terme [O]

Le Glossaire de GenSpec doit permettre de fournir à l'utilisateur l'historique de l'ensemble des modifications réalisées sur une définition d'un terme : (exig. # 1872)

- a) Initialisation [O]. (exig. # 2015) Le glossaire de GenSpec doit établir comme premier historique de la définition d'un terme l'ajout de cette définition au glossaire ; (exig. # 2015)
- b) Informations sur la version d'un historique [O]. (exig. # 2012) Le glossaire de GenSpec doit retenir le numéro de version, la date et l'heure de modification, l'auteur de la modification et

son statut et le projet dans lequel la modification a été réalisée pour identifier un élément de l'historique ; (exig. # 2012)

- c) Informations sur le terme retenues dans l'historique [O]. (exig. # 2038) Le glossaire de GenSpec doit retenir, pour chaque version de l'historique d'un terme défini au sein du glossaire, la définition, les propriétés, le but pour l'acteur, le but pour le système, la liste de synonymes, le niveau d'ambiguïté et le commentaire associé au terme ; (exig. # 2038)
- d) Justification de modification [O]. (exig. # 2013) Le glossaire de GenSpec doit demander – et retenir- la justification de changement sur la définition d'un terme fournie par l'utilisateur ; (exig. # 2013)
- e) Numéro de version de l'historique [O]. (exig. # 2014) Le glossaire de GenSpec doit incrémenter automatiquement le numéro de version de l'historique à toute nouvelle modification apportée à la définition du terme au sein du glossaire. (exig. # 2014)

#### **1.1.1.5. Définition d'un terme défini**

Le Glossaire de GenSpec doit permettre à l'utilisateur de modifier les informations d'un terme défini au sein du glossaire. Voir référence interne ci après identifiée pour établir quelles sont les informations modifiables et établir quelles sont les vérifications à effectuer. (exig. # 2016)

Réf. interne : 1.1.1.1, 1.1.1.2.

##### **1.1.1.5.1. Généralisation de modifications**

Le Glossaire de GenSpec doit permettre de propager une quelconque modification d'un terme effectuée au sein du glossaire par l'utilisateur sur l'ensemble des exigences dans lesquelles il se trouve : (exig. # 2020)

- a) Avertissement de modification. (exig. # 2022) Le glossaire de GenSpec doit avertir l'utilisateur que les exigences où le terme apparaît peuvent être dans un état incohérent suite à une modification apportée au terme défini au sein du glossaire ; (exig. # 2022)
- b) Vérification de la grammaire [C]. (exig. # 1924) Aucune vérification de la grammaire ne doit être effectuée – excepté une vérification de base de "Microsoft Word" si l'utilisateur le désire – par le glossaire après une modification apportée à un terme au sein des exigences où le terme apparaît. (exig. # 1924)

NOTE – Exemple : une modification de l'orthographe d'un terme au sein du glossaire entraîne une modification de l'orthographe du terme dans la totalité des exigences où il apparaît.

##### **1.1.1.5.2. Explication de la modification [O]**

Le Glossaire de GenSpec doit demander à l'utilisateur de fournir une justification concernant la raison du changement réalisé sur l'ensemble des informations d'un terme lors d'une modification : (exig. # 1948)

- a) Justification de modification pour l'historique [O]. (exig. # 2021) L'explication de la modification fournie par l'utilisateur (le visiteur et acteurs plus haut d'un point de vue hiérarchie également car ils peuvent modifier le niveau d'ambiguïté) doit être associée à la version de l'historique réalisée suite à la modification. (exig. # 2021)

##### **1.1.1.5.3. Modification d'un terme**

Le Glossaire de GenSpec doit permettre à l'utilisateur de modifier la façon d'écrire – orthographe ou remplacer le terme par un autre – d'un terme défini au sein du glossaire. Voir les références internes ci après identifiée pour la généralisation de la modification et pour la gestion de l'historique : (exig. # 2023)

- a) Utilisation d'un synonyme. (exig. # 1953) Le glossaire de GenSpec doit permettre de remplacer un terme défini au sein du glossaire par un de ses synonymes. Voir référence interne ci après identifiée pour la généralisation de la modification. (exig. # 1953)

#### **1.1.1.5.4. Modification concernant l'ensemble des informations d'un terme**

Le Glossaire de GenSpec doit permettre à l'utilisateur de modifier les informations comprises au sein du glossaire concernant un terme. Correspond, d'un point de vue modification et non ajout au glossaire, à la référence interne ci après identifiée et ses enfants. (exig. # 2019)

Réf. interne : 1.1.1.4.

#### **1.1.1.6. Suppression de définition**

Le Glossaire de GenSpec doit permettre à l'utilisateur de supprimer l'ensemble des informations associées à un terme au sein du glossaire. Voir référence interne ci après identifiée pour les interdictions à respecter. (exig. # 1861)

Réf. interne : 1.1.1.1.

#### **1.1.1.6.1. Suppression de l'ensemble des informations d'un terme**

Le Glossaire de GenSpec doit permettre à l'utilisateur de supprimer un terme et l'ensemble de ses informations au sein du glossaire. (exig. # 2024)

#### **1.1.1.6.2. Suppression fictive**

Le Glossaire de GenSpec doit retenir l'ensemble des informations concernant un terme qu'il a en sa possession après suppression d'un terme par l'utilisateur au sein du glossaire : (exig. # 1950)

- a) Informations des termes supprimés non accessibles pour tous les utilisateurs. (exig. # 2025)  
Le glossaire de GenSpec ne doit pas permettre l'affichage à tous les utilisateurs l'ensemble des informations concernant les termes supprimés au sein du glossaire ; (exig. # 2025)
- b) Informations des termes supprimés accessibles par l'administrateur [O]. (exig. # 2026) Le glossaire de GenSpec doit permettre à l'administrateur d'accéder, au sein du glossaire, aux informations des termes supprimés ; (exig. # 2026)
- c) Affichage distinctif des termes supprimés [O]. (exig. # 2027) Le glossaire de GenSpec doit afficher de manière distinctive au sein de la liste des termes définis dans le glossaire pour l'administrateur, les termes qui ont été supprimés. (exig. # 2027)

#### **1.1.1.7. Termes prioritaires [O]**

Le Glossaire de GenSpec doit permettre la gestion des termes prioritaires par l'utilisateur au sein de la fenêtre du glossaire. (exig. # 2058)

NOTE – Un terme prioritaire est un terme apparaissant de nombreuses fois au sein des exigences et qui ne possède pas de définition au sein du glossaire.

#### **1.1.1.7.1. Consultation des termes prioritaires**

Le Glossaire de GenSpec doit permettre la consultation, par le visiteur, des termes prioritaires à définir au sein du glossaire : (exig. # 2059)

- a) Filtrage des termes prioritaires. (exig. # 2062) La liste des termes prioritaires ne doit pas comporter tous les termes courants de formations de phrase tel que les articles (le, la, les, etc.), les éléments de liaison, etc ; (exig. # 2062)



- b) Tri des termes prioritaires. (exig. # 2063) Les termes prioritaires doivent être affichés au sein d'une liste -au sein de la fenêtre du glossaire- à l'utilisateur par ordre décroissant d'occurrence au sein des exigences ; (exig. # 2063)
- c) Affichage du nombre d'occurrences. (exig. # 2064) Le glossaire de GenSpec doit afficher au visiteur le nombre d'occurrences d'un terme prioritaire au sein de la liste des termes prioritaires ; (exig. # 2064)
- d) Différentes formes d'un terme prioritaire. (exig. # 2065) Le glossaire de GenSpec ne doit pas prendre en compte les différentes formes d'un terme prioritaire au sein de la liste des termes prioritaires. (exig. # 2065) NOTE – Forme renvoyant au fait qu'un terme peut être au pluriel, les éléments faisant la liaison au sein d'un ensemble de termes composés, etc.

#### 1.1.1.7.2. Définition d'un terme prioritaire

Le Glossaire de GenSpec doit permettre, à l'utilisateur, de définir un terme prioritaire – Voir référence interne ci après identifiée pour les informations à fournir sur un terme au sein du glossaire – directement à partir du menu lié aux termes prioritaires de la fenêtre du glossaire : (exig. # 2060)

- a) Suppression du terme prioritaire. (exig. # 2061) Une fois qu'un terme prioritaire a été défini, celui-ci ne doit plus apparaître au sein de la liste des termes prioritaires. (exig. # 2061)

### **1.1.2. Gestion Intrans / Extrants**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir des intrants/extrants directement à partir du glossaire et de pouvoir les inclure directement au sein de l'arbre hiérarchique des exigences. (exig. # 1958)

Il en découle les exigences suivantes :

- 1) Définition à partir du glossaire
- 2) Définition à partir de l'arbre hiérarchique
- 3) Ajout à l'arbre hiérarchique
- 4) Propriétés
- 5) Affichage des exigences
- 6) Suppression à partir l'arbre hiérarchique
- 7) Non-défini

#### **1.1.2.1. Définition à partir du glossaire**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un terme comme étant intrant ou extrant au sein du glossaire. (exig. # 1985)

##### **1.1.2.1.1. Définition d'un intrant à partir du glossaire**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un intrant directement à partir du glossaire sans nécessairement le lier à une exigence. (exig. # 1959)

##### **1.1.2.1.2. Définition d'un extrant à partir du glossaire**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un extrant directement à partir du glossaire sans nécessairement le lier à une exigence. (exig. # 1978)

#### **1.1.2.2. Définition à partir de l'arbre hiérarchique**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un terme comme étant intrant ou extrant directement à partir de l'arbre hiérarchique. (exig. # 1989)

##### **1.1.2.2.1. Définition d'un intrant à partir de l'arbre hiérarchique**

Le Glossaire de GenSpec doit permettre à l'utilisateur d'ajouter au glossaire la définition d'un terme correspondant à un intrant à partir de l'arbre hiérarchique. (exig. # 1990)

##### **1.1.2.2.2. Définition d'un extrant à partir de l'arbre hiérarchique**

Le Glossaire de GenSpec doit permettre à l'utilisateur d'ajouter au glossaire la définition d'un terme correspondant à un extrant à partir de l'arbre hiérarchique. (exig. # 1991)

#### **1.1.2.3. Ajout à l'arbre hiérarchique**

Le Glossaire de GenSpec doit permettre à l'utilisateur d'ajouter des termes définis comme intrant/extrant directement au sein de l'arbre hiérarchique. (exig. # 1986)

#### 1.1.2.3.1. Ajout d'intrants à l'arbre hiérarchique

Le Glossaire de GenSpec doit permettre à l'utilisateur de faire glisser un terme défini comme intrant au sein du glossaire dans l'arbre hiérarchique : (exig. # 1976)

- a) Respect des contraintes de la base de données vis-à-vis des intrants. (exig. # 1979) L'introduction d'un intrant au sein de l'arbre hiérarchique doit répondre aux contraintes définies au sein de la base données ; (exig. # 1979) NOTE – Un intrant ne peut pas se retrouver n'importe où au sein d'un arbre hiérarchique.
- b) Terme non défini comme intrant. (exig. # 2075) Le glossaire de GenSpec doit avertir l'utilisateur lorsqu'il souhaite introduire un terme au sein de l'arbre hiérarchique comme intrant alors que cette propriété n'est pas établie au sein du glossaire ; (exig. # 2075)
- c) Définir un terme comme intrant. (exig. # 2078) Le glossaire de GenSpec doit permettre à l'utilisateur de définir un terme comme intrant – s'il n'était pas défini comme intrant avant l'insertion dans l'arbre – lors de l'insertion du terme comme intrant au sein de l'arbre hiérarchique ; (exig. # 2078)
- d) Énoncé équivalent au titre de l'exigence. (exig. # 2101) L'énoncé d'un intrant doit être identique à son titre et inséré automatiquement lors de la définition d'un intrant. (exig. # 2101)

#### 1.1.2.3.2. Ajout d'extrants à l'arbre hiérarchique

Le Glossaire de GenSpec doit permettre à l'utilisateur de faire glisser un terme défini comme intrant au sein du glossaire dans l'arbre hiérarchique : (exig. # 1977)

- a) Respect des contraintes de la base de données vis-à-vis des extrants. (exig. # 1980) L'introduction d'un extrant au sein de l'arbre hiérarchique doit répondre aux contraintes définies au sein de la base données ; (exig. # 1980) NOTE – Un extrant ne peut pas se retrouver n'importe où au sein d'un arbre hiérarchique.
- b) Terme non défini comme extrant. (exig. # 2076) Le glossaire de GenSpec doit avertir l'utilisateur lorsqu'il souhaite introduire un terme au sein de l'arbre hiérarchique comme extrant alors que cette propriété n'est pas établie au sein du glossaire ; (exig. # 2076)
- c) Définir le terme comme extrant. (exig. # 2077) Le glossaire de GenSpec doit permettre à l'utilisateur de définir un terme comme extrant – s'il n'était pas défini comme extrant avant l'insertion dans l'arbre – lors de l'insertion du terme comme extrant au sein de l'arbre hiérarchique ; (exig. # 2077)
- d) Énoncé équivalent au titre de l'exigence. (exig. # 2102) L'énoncé d'un extrant doit être identique à son titre et inséré automatiquement lors de la définition d'un extrant. (exig. # 2102)

#### 1.1.2.3.3. Synchronisation entre le glossaire et les exigences [O]

Le Glossaire de GenSpec doit synchroniser les exigences établies au sein de l'arbre hiérarchique à partir du glossaire : (exig. # 2097)

- a) Synchronisation vers le glossaire. (exig. # 2098) Le glossaire de GenSpec doit permettre la modification automatique des informations associées au terme, au sein du glossaire, lorsque des modifications sont apportées à une exigence qui a été établie directement à partir de ce terme au sein de l'arbre hiérarchique ; (exig. # 2098)
- b) Synchronisation vers l'arbre hiérarchique. (exig. # 2099) Le glossaire de GenSpec doit permettre la modification automatique, au sein des exigences de l'arbre hiérarchique, des exigences qui ont été établies à partir d'un terme du glossaire lorsque qu'une modification est apportée par l'utilisateur aux informations de ce terme au sein du glossaire ; (exig. # 2099)

- c) Synchronisation entre les exigences. (exig. # 2100) Le glossaire de GenSpec doit permettre la synchronisation directe entre les exigences établies au sein de l'arbre hiérarchique à partir d'un même terme au sein du glossaire. (exig. # 2100)

#### **1.1.2.4. Propriétés**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir des termes qui sont à la fois intrants et extrants ou aucun des deux au sein du glossaire. (exig. # 1987)

##### **1.1.2.4.1. Double propriété : Intrant et extrant**

Le Glossaire de GenSpec doit permettre à l'utilisateur définir un terme comme étant un intrant et un extrant. (exig. # 1981)

##### **1.1.2.4.2. Aucune propriété : ni intrant ni extrant**

Le Glossaire de GenSpec doit permettre à l'utilisateur de définir un terme qui ne soit ni un intrant ni un extrant. (exig. # 1982)

#### **1.1.2.5. Affichage des exigences [O]**

Le Glossaire de GenSpec doit permettre d'afficher au visiteur, au sein de la fenêtre du glossaire de GenSpec, les exigences où le terme constitue un intrant ou un extrant. (exig. # 1988)

##### **1.1.2.5.1. Exigences où le terme constitue un intrant**

Le Glossaire de GenSpec doit permettre d'afficher au visiteur, au sein de la liste des exigences dans la fenêtre du glossaire, les exigences pour lesquelles le terme constitue un intrant. (exig. # 1983)

##### **1.1.2.5.2. Exigences où le terme constitue un extrant**

Le Glossaire de GenSpec doit permettre d'afficher au visiteur, au sein de la liste des exigences dans la fenêtre du glossaire, les exigences pour lesquelles le terme constitue un extrant. (exig. # 1984)

#### **1.1.2.6. Suppression à partir l'arbre hiérarchique [O]**

Le Glossaire de GenSpec doit avertir l'utilisateur lorsque la propriété intrant ou extrant n'est plus vérifiée au sein de l'arbre hiérarchique suite à la suppression du dernier élément correspondant dans l'arbre. (exig. # 1992)

##### **1.1.2.6.1. Suppression d'un intrant à partir de l'arbre hiérarchique**

Le Glossaire de GenSpec doit avertir l'utilisateur lorsqu'un terme défini comme intrant au sein du glossaire voit sa dernière occurrence en tant qu'intrant supprimée au sein de l'arbre hiérarchique : (exig. # 1997)

- a) Avertissement de suppression d'intrant. (exig. # 1998) Il faut fournir à l'utilisateur le choix de ne pas supprimer la dernière occurrence ou de changer la définition du terme – dire qu'il n'est plus intrant – ou supprimer l'élément et conserver la propriété. (exig. # 1998)

#### **1.1.2.6.2. Suppression d'un extrait à partir de l'arbre hiérarchique**

Le Glossaire de GenSpec doit avertir l'utilisateur lorsqu'un terme défini comme extrait au sein du glossaire voit sa dernière occurrence en tant qu'intrant supprimée au sein de l'arbre hiérarchique : (exig. # 1999)

- a) Avertissement de suppression d'extrait. (exig. # 2000) Il faut fournir à l'utilisateur le choix de ne pas supprimer la dernière occurrence ou de changer la définition du terme – dire qu'il n'est plus intrant – ou supprimer l'élément et conserver la propriété. (exig. # 2000)

#### **1.1.2.7. Non-défini [O]**

Le Glossaire de GenSpec doit permettre au visiteur de distinguer les intrants et extraits définis au sein du glossaire de ceux non définis dans l'arbre hiérarchique. (exig. # 1994)

##### **1.1.2.7.1. Intrant non-défini au sein de l'arbre hiérarchique**

Le Glossaire de GenSpec doit permettre au visiteur de différencier, dans l'arbre hiérarchique, les intrants non définis des intrants définis au sein du glossaire. (exig. # 1995)

##### **1.1.2.7.2. Extrait non-défini au sein de l'arbre hiérarchique**

Le Glossaire de GenSpec doit permettre au visiteur de différencier, dans l'arbre hiérarchique, les extraits non définis des extraits définis au sein du glossaire. (exig. # 1996)

##### **1.1.2.7.3. Termes définis soulignés**

Le Glossaire de GenSpec doit afficher au sein de l'arbre hiérarchique, les intrants et les extraits qui sont définis au sein du glossaire de manière soulignée. (exig. # 2079)

### **1.1.3. Gestion des informations**

Le Glossaire de GenSpec doit permettre au visiteur d'afficher et de consulter l'ensemble des informations contenues au sein du glossaire. (exig. # 1845)

Il en découle les exigences suivantes :

- 1) Consultation des informations
- 2) Affichage des informations
- 3) Recherche au sein du glossaire

#### **1.1.3.1. Consultation des informations**

Le Glossaire de GenSpec doit permettre à l'utilisateur de consulter l'ensemble des informations concernant un terme au sein du glossaire. (exig. # 2032)

##### **1.1.3.1.1. Consultation d'un terme à partir du glossaire**

Le Glossaire de GenSpec doit permettre au visiteur de consulter l'ensemble des informations associées à un terme au sein du glossaire. Voir référence interne ci après identifiée pour établir la manière d'afficher les informations. (exig. # 2028)

Réf. interne : 1.1.3.2.1.

##### **1.1.3.1.2. Consultation d'un terme à partir d'une exigence**

Le Glossaire de GenSpec doit permettre au visiteur de consulter l'ensemble des informations – Voir référence interne ci après identifiée pour établir la manière d'afficher les informations – concernant un terme défini au sein du glossaire à partir d'une exigence affichée dans la fenêtre des exigences où le terme est utilisé. (exig. # 2031)

Réf. interne : 1.1.3.2.1.

##### **1.1.3.1.3. Consultation de la liste des termes définis**

Le Glossaire de GenSpec doit permettre au visiteur de consulter la liste des termes définis au sein du glossaire dans la fenêtre du glossaire. (exig. # 2033)

##### **1.1.3.1.4. Consultation de l'ensemble des informations associées à un terme**

Le Glossaire de GenSpec doit permettre d'afficher au visiteur, pour tout termes définis au sein du glossaire, l'ensemble des informations qui lui sont associées – Voir référence interne ci après identifiée pour l'ensemble des informations contenues dans la définition d'un terme -. (exig. # 2035)

Réf. interne : 1.1.1.4.

#### **1.1.3.2. Affichage des informations**

Le Glossaire de GenSpec doit permettre d'afficher au visiteur un ensemble d'informations correspondant au glossaire. (exig. # 1956)

#### 1.1.3.2.1. Affichage des informations au sein de la fenêtre du glossaire

Le Glossaire de GenSpec doit afficher au visiteur l'ensemble des informations associées à un terme au sein de la fenêtre du glossaire dans GenSpec. (exig. # 2034)

#### 1.1.3.2.2. Affichage de l'historique [O]

Le Glossaire de GenSpec doit permettre au visiteur de consulter la liste des modifications qu'a subi l'ensemble des informations associées à un terme défini au sein du glossaire : (exig. # 1949)

- a) Liste des versions de la définition [O]. (exig. # 2036) Le glossaire de GenSpec doit afficher au visiteur une liste des différentes versions de l'ensemble des informations associées à un terme au sein du glossaire dans la fenêtre du glossaire ; (exig. # 2036)
- b) Justification de la modification [O]. (exig. # 2037) Pour chaque version, il doit être possible de consulter la raison de la modification par rapport à la version de la définition précédente au sein de la fenêtre du glossaire associée à l'historique ; (exig. # 2037)
- c) Filtre [O]. (exig. # 1964) Le glossaire de GenSpec doit permettre de filtrer l'historique d'un terme selon une période temps un auteur (ou son statut) et un projet et ce, au sein de la fenêtre du glossaire liée à l'historique. (exig. # 1964)

#### 1.1.3.2.3. Affichage des exigences [O]

Le Glossaire de GenSpec doit permettre au visiteur d'afficher, dans la fenêtre dédiée aux exigences, l'exigence sélectionnée – au sein de la fenêtre du glossaire – dans la liste des exigences d'un terme défini au sein du glossaire. (exig. # 1927)

#### 1.1.3.2.4. Affichage des termes définis dans une exigence

Le Glossaire de GenSpec doit permettre d'afficher au visiteur de manière distincte par rapport à l'ensemble du texte de l'exigence, dans une quelconque exigence dans la fenêtre des exigences, les termes qui sont définis au sein du glossaire. (exig. # 2039)

#### 1.1.3.2.5. Affichage de la définition d'un terme directement à partir de la fenêtre des exigences

Le Glossaire de GenSpec doit permettre d'afficher au visiteur la définition d'un terme juste en plaçant le curseur de la souris sur ce terme. (exig. # 1955)

NOTE – Définition réfère ici juste à la définition du concept et non l'ensemble des informations associées au terme au sein du glossaire.

#### 1.1.3.2.6. Affichage des termes ambigus [O]

Le Glossaire de GenSpec doit permettre d'afficher au visiteur, de manière distincte, dans la liste des termes au sein de la fenêtre du glossaire les termes ayant un niveau d'ambiguïté élevé. (exig. # 2018)

#### 1.1.3.3. Recherche au sein du glossaire [O]

Le Glossaire de GenSpec doit permettre au visiteur d'effectuer une recherche au sein des termes définis dans le glossaire. (exig. # 1926)

##### 1.1.3.3.1. Recherche selon le terme

Le Glossaire de GenSpec doit permettre au visiteur de consulter l'ensemble des informations associées à un terme en introduisant ce terme au sein du glossaire. (exig. # 1974)

#### **1.1.3.3.2. Recherche selon un ensemble de critères**

Le Glossaire de GenSpec doit ajouter à la fonction de recherche déjà présente au sein de GenSpec, la possibilité au visiteur de rechercher un mot dans l'ensemble des informations concernant un terme défini au sein du glossaire. (exig. # 1975)

NOTE – Ici, le terme "mot" est employé au sein de l'exigence pour référer à la terminologie employée vis-à-vis du menu de recherche déjà présent au sein de GenSpec.

#### **1.1.3.3.3. Gestion des différentes formes des termes**

Le Glossaire de GenSpec doit pouvoir gérer les différentes formes possibles que peut posséder un termes. (exig. # 2052)

NOTE – La notion de "formes" renvoie au pluriel, à l'emploi d'articles différents au sein d'un groupe de mots, etc.



#### 1.1.4. Utilisation

Le Glossaire de GenSpec doit permettre à l'utilisateur d'utiliser les termes définis au sein du glossaire pour l'élaboration des exigences. (exig. # 1855)

Il en découle les exigences suivantes :

- 1) Utilisation dans une exigence

##### 1.1.4.1. Utilisation dans une exigence

Le Glossaire de GenSpec doit permettre à l'utilisateur d'utiliser un terme défini au sein du glossaire dans d'une exigence. (exig. # 1920)

##### 1.1.4.1.1. Utiliser un terme défini dans une exigence

Le Glossaire de GenSpec doit permettre à l'utilisateur d'utiliser un terme défini au sein du glossaire dans plusieurs énoncés d'exigences distinctes ainsi que dans plusieurs définitions d'intrants et d'extrants. (exig. # 2040)

##### 1.1.4.1.2. Interdiction d'utiliser un synonyme

Le Glossaire de GenSpec doit proposer à l'utilisateur, lorsqu'il souhaite employer un terme présent au sein d'une liste de synonymes d'un terme défini au sein d'une exigence, le terme défini dans le glossaire contenant cette liste : (exig. # 1921)

- a) Message informatif. (exig. # 1973) Le glossaire de GenSpec doit fournir à l'utilisateur un message expliquant pourquoi il ne peut utiliser un terme se trouvant dans une liste de synonymes définie au sein du glossaire ; (exig. # 1973)
- b) Choix direct. (exig. # 2071) Le glossaire de GenSpec doit fournir à l'utilisateur lors de la rédaction de l'exigence, directement après l'introduction d'un synonyme au sein de cette exigence, le terme correspondant à ce synonyme au sein du glossaire. L'utilisateur doit valider, à long terme ou non, le terme fournit par le glossaire avant l'utilisation au sein de l'exigence ; (exig. # 2071)
- c) Vérification à posteriori [C]. (exig. # 2095) Le glossaire de GenSpec doit permettre à l'utilisateur de réaliser une vérification globale – de la totalité du projet – des exigences afin de vérifier si l'on utilise des synonymes au sein des exigences. (exig. # 2095)

NOTE – 1 mot = 1 sens ; 1 sens = 1 mot.

##### 1.1.4.1.3. Terme définis soulignés

Le Glossaire de GenSpec doit afficher de manière soulignée, au sein des exigences et du titre de l'exigence, les termes possédant une définition non implicite au sein du glossaire. (exig. # 2096)

### **1.1.5. Gestion courante**

Le Glossaire de GenSpec doit fournir à l'utilisateur un ensemble de services complémentaires nécessaires pour la gestion du glossaire. (exig. # 1863)

Il en découle les exigences suivantes :

- 1) Base de données
- 2) Importation de glossaire(s)
- 3) Génération d'un document
- 4) Gestion générale du glossaire
- 5) Propriétés
- 6) Contrainte(s)

#### **1.1.5.1. Base de données**

Le Glossaire de GenSpec doit permettre une gestion de l'ensemble des bases de données associées à GenSpec. (exig. # 2074)

##### **1.1.5.1.1. Base de données précédentes**

Le Glossaire de GenSpec doit permettre une gestion des versions de base de données précédant l'ajout de la fonction glossaire au sein de GenSpec. (exig. # 2086)

#### **1.1.5.2. Importation de glossaire(s) [C]**

Le Glossaire de GenSpec doit permettre à l'utilisateur d'importer un ou plusieurs glossaires provenant d'autres projets GenSpec au sein d'un glossaire d'un autre projet. Voir référence interne ci après identifiée pour établir les propriétés à respecter. (exig. # 1916)

Réf. interne : 1.1.1.1.

##### **1.1.5.2.1. Choix entre synonymes**

Le Glossaire de GenSpec doit permettre à l'utilisateur de choisir entre 2 définitions de termes synonymes lors de l'importation d'un glossaire au sein d'un autre glossaire : (exig. # 1944)

- a) Choix pour tous les synonymes. (exig. # 1945) Le glossaire fournit à l'utilisateur la possibilité d'établir un choix général pour l'ensemble des synonymes ; (exig. # 1945) NOTE – Accepter/Refuser de les remplacer tous.
- b) Comparaison [O]. (exig. # 1946) Le glossaire fournit à l'utilisateur la possibilité de comparer chaque paire de synonymes vis-à-vis de leurs définitions ; (exig. # 1946) NOTE – Définition correspondant à l'ensemble des informations sur un terme.
- c) Choix unique [O]. (exig. # 1947) Le glossaire fournit à l'utilisateur la possibilité de choisir de manière unique pour chaque paire de synonyme (sans la comparaison). (exig. # 1947)

NOTE – Définition ici se rapportant à la totalité des informations concernant un terme au sein du glossaire.

##### **1.1.5.2.2. Termes supprimés**

Le Glossaire de GenSpec doit importer tous les termes d'un glossaire au sein d'un autre glossaire lors d'une importation excepté les termes supprimés. Voir référence interne ci après identifiée pour respecter les propriétés liées à la suppression d'un terme. (exig. # 1952)

Réf. interne : 1.1.1.6.

### **1.1.5.3. Génération d'un document**

Le Glossaire de GenSpec doit permettre au visiteur de générer un document "Microsoft Word" spécifique au contenu du glossaire. (exig. # 1911)

#### **1.1.5.3.1. Tableau [C]**

Le Glossaire de GenSpec doit générer un document concernant les informations du glossaire sous forme de tableaux à 2 colonnes (exig. # 2089)

#### **1.1.5.3.2. Informations reprises**

Le Glossaire de GenSpec doit reprendre 3 types d'informations au sein du document généré : (exig. # 2090)

- a) Définition. (exig. # 2091) Le glossaire de GenSpec doit reprendre la définition du terme lors de la génération du document ; (exig. # 2091)
- b) But pour l'acteur. (exig. # 2092) Le glossaire de GenSpec doit reprendre le but pour l'acteur du terme lors de la génération du document ; (exig. # 2092)
- c) But pour le système. (exig. # 2093) Le glossaire de GenSpec doit reprendre le but pour le système du terme lors de la génération du document ; (exig. # 2093)
- d) Champs implicites. (exig. # 2094) Le glossaire de GenSpec doit intégrer au sein du document généré l'ensemble des définitions de terme si au moins un des champs – définition, but pour le système, but pour acteur – est défini. (exig. # 2094)

#### **1.1.5.3.3. Ordre [C]**

Le Glossaire de GenSpec doit permettre au visiteur de générer un document où les termes du glossaire sont organisés dans un ordre alphanumérique. (exig. # 1971)

#### **1.1.5.3.4. Glossaire partiel [O]**

Le Glossaire de GenSpec doit permettre au visiteur de générer un document contenant toutes les informations concernant une sélection de termes du glossaire. (exig. # 1913)

#### **1.1.5.3.5. Informations partielles [O]**

Le Glossaire de GenSpec doit permettre au visiteur de définir les informations qu'il souhaite obtenir et ce, que soit pour une sélection de termes ou que ce soit pour la totalité des termes. (exig. # 1914)

#### **1.1.5.3.6. Options de génération**

Le Glossaire de GenSpec doit ajouter les options nécessaires à la génération de documents concernant le glossaire dans la fenêtre d'option de génération : (exig. # 2082)

- a) Totalité du glossaire [C]. (exig. # 1912) Le glossaire de GenSpec doit permettre au visiteur de générer un document contenant la totalité des informations concernant chaque terme contenu au sein du glossaire ; (exig. # 1912)
- b) Termes définis distincts [C]. (exig. # 2081) Le glossaire de GenSpec doit permettre au visiteur d'afficher de manière distincte (styles paramétrables Word) chaque occurrence des termes définis (définition non implicite) au sein du glossaire dans les documents résultant de la génération de document ; (exig. # 2081)

- c) Glossaire au sein du document des exigences. (exig. # 2072) Le glossaire de GenSpec doit permettre selon le choix du visiteur, lors de la génération du document des exigences, de définir un terme après sa première apparition au sein d'une exigence dans le document des exigences ; (exig. # 2072)
- d) Affichage du quadrillage [C]. (exig. # 2087) Le glossaire de GenSpec doit fournir la possibilité au visiteur de choisir de générer le document avec ou sans quadrillage correspondant au tableau ; (exig. # 2087)
- e) Affichage du nom des colonnes [C]. (exig. # 2088) Le glossaire de GenSpec doit fournir à l'utilisateur la possibilité d'ajouter ou supprimer le nom des colonnes du tableau reprenant les informations sur les termes émanant du glossaire. (exig. # 2088)

#### **1.1.5.4. Gestion générale du glossaire**

Le Glossaire de GenSpec doit permettre de fournir au visiteur des explications sur son fonctionnement et des options de contrôle à l'administrateur. (exig. # 2041)

##### **1.1.5.4.1. Choix de fonctionnalités par l'administrateur [O]**

Le Glossaire de GenSpec doit permettre à l'administrateur de définir l'utilisation du glossaire pour l'ensemble des participants au sein d'un projet : (exig. # 1941)

- a) Obligation d'utiliser le glossaire. (exig. # 2043) L'administrateur doit pouvoir spécifier, par le biais de la fenêtre d'option associée au glossaire, s'il est obligatoire d'employer le glossaire ou non au sein d'un projet ; (exig. # 2043)
- b) Informations obligatoires. (exig. # 2044) L'administrateur doit pouvoir spécifier, par le biais de la fenêtre d'option associée au glossaire, quelles sont les informations obligatoires que les différents utilisateurs devront fournir lors de la définition d'un terme au sein du glossaire. (exig. # 2044)

##### **1.1.5.4.2. Menu d'aide**

Le Glossaire de GenSpec doit permettre au visiteur de consulter un menu d'aide pour lui fournir l'ensemble des informations nécessaires à l'utilisation du glossaire : (exig. # 1915)

- a) Ajout au menu d'aide actuel. (exig. # 2045) Le menu d'aide associé au glossaire doit être ajouté au menu d'aide existant actuellement au sein de GenSpec ; (exig. # 2045)
- b) Consultation. (exig. # 2049) Le menu d'aide du glossaire de GenSpec doit être accessible par tous les types différents d'utilisateurs de GenSpec. (exig. # 2049)

#### **1.1.5.5. Propriétés**

Le Glossaire de GenSpec doit posséder au maximum un glossaire par projet. (exig. # 2042)

##### **1.1.5.5.1. Affichage des termes définis**

Le Glossaire de GenSpec doit afficher au visiteur les termes définis au sein du glossaire et ce, quelque soit le texte concerné, de manière soulignée – excepté au sein du glossaire où l'on définit la syntaxe du terme -. (exig. # 2080)

##### **1.1.5.5.2. Un glossaire par projet**

Le Glossaire de GenSpec doit permettre à l'utilisateur l'élaboration d'un glossaire par projet : (exig. # 1880)

- a) Maximum. (exig. # 2046) Le glossaire de GenSpec doit permettre l'élaboration au maximum d'un glossaire par projet ; (exig. # 2046)
- b) Glossaire vide. (exig. # 2048) Un glossaire peut être vide si l'utilisateur n'a défini aucun terme ni importé aucun autre glossaire au sein du glossaire du projet. (exig. # 2048)

NOTE – 1 projet = 1 glossaire.

#### **1.1.5.6. Contrainte(s)**

Le Glossaire de GenSpec doit respecter un ensemble de contraintes concernant sa réalisation. (exig. # 2085)

##### **1.1.5.6.1. Langage**

Le Glossaire de GenSpec doit être réalisé, comme la totalité du logiciel GenSpec, en Visual Basic 6. (exig. # 2084)

## **Annexe F**

# **Exigences : GMDE et AECF**

La dernière modification du contenu de ce document date du 8 janvier 2007.

*Exigences de la GMDE et l'AECF*

**GenSpec**

**Gestion multi-documents**

**OCTOBRE 2006**

## 1. EXIGENCES SPÉCIFIQUES INITIALES

### 1.1. Exigences fonctionnelles

#### 1.1.1. Centralisation des informations

##### 1.1.1.1. Gestion des regroupements

###### 1.1.1.1.1. Création d'un regroupement

- 1.1.1.1.1.a) Nom unique
- 1.1.1.1.1.b) Contenu du regroupement
- 1.1.1.1.1.c) Numéro identifiant
- 1.1.1.1.1.d) Création des paramètres de génération

###### 1.1.1.1.2. Modification du nom d'un regroupement

- 1.1.1.1.2.a) Nom unique

###### 1.1.1.1.3. Suppression d'un regroupement

##### 1.1.1.1.4. Propriétés des arbres hiérarchiques

- 1.1.1.1.4.a) Racines de l'arbre global
- 1.1.1.1.4.b) Racine d'un regroupement
- 1.1.1.1.4.c) Correspondance entre un regroupement et l'arbre global
- 1.1.1.1.4.d) Correspondance entre l'arbre global et un regroupement
- 1.1.1.1.4.e) Pas de limitation quant aux nombre de regroupements à créer

##### 1.1.1.2. Gestion de la vérification

- 1.1.1.2.1. Comptage et listage des exigences de format ne faisant aucun renvoi vers une exigence

- 1.1.1.2.2. Comptage et listage des exigences ne faisant aucun renvoi vers une exigence de format

- 1.1.1.2.3. Raccourci vers une exigence faisant l'objet d'une erreur

- 1.1.1.2.4. Raccourci vers une exigence faisant l'objet d'un avertissement

##### 1.1.1.3. Opérations de déplacement d'une exigence

- 1.1.1.3.1.a) Couper avec la souris
- 1.1.1.3.1.b) Ajout dans le menu contextuel au même niveau
- 1.1.1.3.1.c) Couper avec le raccourci clavier
- 1.1.1.3.1.d) Couper avec le menu contextuel
- 1.1.1.3.1.e) Copier avec le raccourci clavier
- 1.1.1.3.1.f) Copier avec le menu contextuel
- 1.1.1.3.1.g) Coller
- 1.1.1.3.1.h) Coller avec synchronisation
- 1.1.1.3.1.i) Coller au même niveau
- 1.1.1.3.1.j) Coller au même niveau avec synchronisation
- 1.1.1.3.1.k) Coller avec liaison et synchronisation au même niveau

##### 1.1.1.4. Gestion de la synchronisation

- 1.1.1.4.1. Atteinte d'une exigence à partir de l'arbre global

- 1.1.1.4.2. Atteinte d'une exigence à partir d'un regroupement

- 1.1.1.4.2.a) L'exigence cible est interne au regroupement

- 1.1.1.4.2.b) L'exigence cible est externe au regroupement et aucune autre fenêtre n'est affichée

- 1.1.1.4.2.c) L'exigence cible est externe au regroupement et une autre fenêtre est affichée

- 1.1.1.4.3. Synchronisation de tous les types

##### 1.1.1.5. Gestion de la génération

- 1.1.1.5.1. Génération d'un document regroupant l'ensemble des exigences

- 1.1.1.5.2. Génération d'un document regroupant l'ensemble des exigences synchronisées

- 1.1.1.5.3. Génération d'un document regroupant les exigences d'un regroupement

- 1.1.1.5.4. Réutilisation des options de génération existantes

- 1.1.1.5.5. Omission des sections Autres fonctions dans l'arbre global

- 1.1.1.5.6. Un onglet par regroupement

##### 1.1.1.6. Modification de la fonctionnalité Exigences générées et applicables

- 1.1.1.6.1. Création d'un fichier par groupe [O]

- 1.1.1.6.2. Gestion des liaisons [O]

- 1.1.1.6.2.a) Intrants [O]

- 1.1.1.6.2.b) Extrants [O]

- 1.1.1.6.3. Avertissement d'un changement automatique dans un groupe

##### 1.1.1.7. Gestion des fenêtres de regroupements

- 1.1.1.7.1. Affichage d'autant de fenêtres que l'on souhaite

- 1.1.1.7.2. Affichage des fenêtres

- 1.1.1.7.2.a) Aucun regroupement créé

- 1.1.1.7.2.b) Au moins un regroupement créé

- 1.1.1.7.2.c) Une seule fenêtre affichée

- 1.1.1.7.2.d) Plusieurs fenêtres affichées

##### 1.1.1.8. Gestion des exigences entre les regroupements

- 1.1.1.8.1. Numéro hiérarchique des exigences d'un regroupement

- 1.1.1.8.2. Création de la section Autres fonctions

- 1.1.1.8.2.a) Création automatique de la section Autres fonctions

- 1.1.1.8.2.b) Interdiction de la création de la section Autres fonctions si elle est déjà présente

## 2. EXIGENCES SPÉCIFIQUES RELATIVES À LA PREMIÈRE ITÉRATION

### 2.1. Exigences fonctionnelles



- 2.1.1. Tableau d'attribution des exigences à des composants
  - 2.1.1.1. Gestion des composants
    - 2.1.1.1.1. Création d'un composant
      - 2.1.1.1.1.a) Nom unique
      - 2.1.1.1.1.b) Contenu du composant
    - 2.1.1.1.2. Modification du nom d'un composant
    - 2.1.1.1.3. Suppression d'un composant
    - 2.1.1.1.4. Attribution des exigences
      - 2.1.1.1.4.a) Déplacement vers l'espace réservé aux attributions
      - 2.1.1.1.4.b) Cohérence avec les composants parents
    - 2.1.1.1.5. Suppression d'une attribution
      - 2.1.1.1.5.a) Suppression de l'attribution seule
      - 2.1.1.1.5.b) Suppression de l'attribution avec ses enfants et parents
    - 2.1.1.1.6. Organisation des composants en niveaux
      - 2.1.1.1.6.a) Arbre hiérarchique de composants
      - 2.1.1.1.6.b) Niveaux des composants
  - 2.1.1.2. Gestion des fonctions
    - 2.1.1.2.1. Même exigences pour la gestion des fonctions que pour la gestion des composants
    - 2.1.1.2.2. Création d'une fonction
      - 2.1.1.2.2.a) Nom unique
      - 2.1.1.2.2.b) Contenu de la fonction
      - 2.1.1.2.2.c) Niveau des fonctions
  - 2.1.1.3. Options et génération du tableau
    - 2.1.1.3.1. Format de sortie du document
      - 2.1.1.3.1.a) Microsoft Excel
    - 2.1.1.3.2. Organisation automatique des colonnes du tableau
      - 2.1.1.3.2.a) Numéro hiérarchique – Titre – ID – Fonction – Composant (niveau 1) – ... – Composant (niveau n)
      - 2.1.1.3.2.b) ID – Numéro hiérarchique – Titre – Fonction – Composant (niveau 1) – ... – Composant (niveau n)
    - 2.1.1.3.3. Regroupement fonctionnel
      - 2.1.1.3.3.a) Un regroupement fonctionnel est souhaité
      - 2.1.1.3.3.b) Un regroupement fonctionnel n'est pas souhaité
      - 2.1.1.3.3.c) Génération si un regroupement fonctionnel est souhaité
      - 2.1.1.3.3.d) Génération si un regroupement fonctionnel n'est pas souhaité
  - 2.1.1.4. Gestion de l'interface graphique
    - 2.1.1.4.1. Menu contextuel pour les attributions
      - 2.1.1.4.1.a) Valider cette attribution
      - 2.1.1.4.1.b) Suppression de l'attribution seule
      - 2.1.1.4.1.c) Suppression de l'attribution avec ses enfants et parents
    - 2.1.1.4.2. Menu contextuel pour les composants
      - 2.1.1.4.2.a) Ajout sous un composant
      - 2.1.1.4.2.b) Ajout d'un composant au même niveau
      - 2.1.1.4.2.c) Modification du nom
      - 2.1.1.4.2.d) Suppression
    - 2.1.1.4.3. Menu contextuel pour les fonctions
      - 2.1.1.4.3.a) Ajout d'une fonction
      - 2.1.1.4.3.b) Modification du nom
      - 2.1.1.4.3.c) Suppression
    - 2.1.1.4.4. Choix d'affichage
      - 2.1.1.4.4.a) Les composants et les attributions
      - 2.1.1.4.4.b) Les fonctions et les attributions
  - 2.1.1.5. Gestion des accès avec l'interface graphique de GenSpec
    - 2.1.1.5.1. Accès à la fenêtre des attributions
      - 2.1.1.5.1.a) Menu Outils
      - 2.1.1.5.1.b) Position du menu Outils
    - 2.1.1.5.2. Accès à la fenêtre 'Progression de la génération du document Excel'
      - 2.1.1.5.2.a) Nouvelle fenêtre
      - 2.1.1.5.2.b) Contenu de la liste 'Documents à générer'
      - 2.1.1.5.2.c) Bouton Options de génération
      - 2.1.1.5.2.d) Position du bouton des Options de génération
    - 2.1.1.5.3. Accès à la fenêtre des options de génération du tableau
    - 2.1.1.5.4. Réorganisation des icônes
      - 2.1.1.5.4.a) Bouton Excel
      - 2.1.1.5.4.b) Icône Excel
  - 2.1.1.6. Conversion des attributions existantes
    - 2.1.1.6.1. Intrants de l'application
      - 2.1.1.6.1.a) La table contenant les attributions existantes
      - 2.1.1.6.1.b) Les tables listant les fonctions, les applications et les composants existants
    - 2.1.1.6.2. Méthodologie d'importation
      - 2.1.1.6.2.a) Convertir la base de données
      - 2.1.1.6.2.b) Choisir la base de données contenant les fonctions, applications et composants
      - 2.1.1.6.2.c) Choisir la base de données contenant les attributions existantes
      - 2.1.1.6.2.d) Intégrer les attributions dans la gestion multi-documents

- 2.1.1.6.2.e) Supprimer les tables relatives aux attributions existantes
  - 2.1.1.6.3. Règles de conversion
    - 2.1.1.6.3.a) Fonction
    - 2.1.1.6.3.b) Application
    - 2.1.1.6.3.c) Composant
  - 2.1.1.6.4. Génération du tableau de conception
    - 2.1.1.6.4.a) Sans la centralisation des informations au sein de GenSpec
    - 2.1.1.6.4.b) Avec la centralisation des informations au sein de GenSpec
  - 2.1.1.7. Gestion de la validation
    - 2.1.1.7.1. Modification d'une exigence
      - 2.1.1.7.1.a) Libellé des attributions
    - 2.1.1.7.2. Suppression d'une exigence
    - 2.1.1.7.3. Validation des attributions
      - 2.1.1.7.3.a) Libellé du menu contextuel
  - 2.1.2. Gestion de la cohérence avec l'environnement extérieur
    - 2.1.2.1. Aide
    - 2.1.2.2. Mise à jour de la base de données
    - 2.1.2.3. Usager concepteur
    - 2.1.2.4. Cohésion avec les fonctionnalités existantes
      - 2.1.2.4.1.a) Menus contextuels pour les fonctions et les composants
      - 2.1.2.4.1.b) Menus contextuels pour les attributions
- 3. EXIGENCES SPÉCIFIQUES RELATIVES À LA DEUXIÈME ITÉRATION**
- 3.1. Exigences fonctionnelles**
- 3.1.1. Centralisation des informations
    - 3.1.1.1. Gestion des regroupements
      - 3.1.1.1.1. Propriétés
        - 3.1.1.1.1.a) Racines de l'arbre global
    - 3.1.1.2. Opérations de déplacement d'une exigence
      - 3.1.1.2.1. Coller au même niveau
        - 3.1.1.2.1.a) Ajout dans le menu contextuel
      - 3.1.1.2.2. Ajouter au même niveau que cette exigence
        - 3.1.1.2.2.a) Ajout dans le menu contextuel
    - 3.1.1.3. Gestion de la génération
      - 3.1.1.3.1. Génération d'un document regroupant l'ensemble des exigences
    - 3.1.1.4. Cohérence avec les fonctionnalités existantes
      - 3.1.1.4.1. Fonctionnalités présentes dans le menu Outils
- 4. EXIGENCES SPÉCIFIQUES RÉALISÉES**
- 4.1. Exigences fonctionnelles**
- 4.1.1. Centralisation des informations
    - 4.1.1.1. Gestion des regroupements
      - 4.1.1.1.1. Propriétés
        - 4.1.1.1.1.a) Racines de l'arbre global
    - 4.1.1.2. Opérations de déplacement d'une exigence
      - 4.1.1.2.1. Coller au même niveau
        - 4.1.1.2.1.a) Ajout dans le menu contextuel
      - 4.1.1.2.2. Ajouter au même niveau que cette exigence
        - 4.1.1.2.2.a) Ajout dans le menu contextuel
    - 4.1.1.3. Gestion de la génération
      - 4.1.1.3.1. Génération d'un document regroupant l'ensemble des exigences
    - 4.1.1.4. Cohérence avec les fonctionnalités existantes
      - 4.1.1.4.1. Fonctionnalités présentes dans le menu Outils
- 5. EXIGENCES SPÉCIFIQUES NON RÉALISÉES**
- 5.1. Exigences fonctionnelles**
- 5.1.1. Centralisation des informations
    - 5.1.1.1. Gestion de la vérification
      - 5.1.1.1.1. Comptage et listage des exigences de format ne faisant aucun renvoi vers une exigence
      - 5.1.1.1.2. Comptage et listage des exigences ne faisant aucun renvoi vers une exigence de format
      - 5.1.1.1.3. Raccourci vers une exigence faisant l'objet d'une erreur
      - 5.1.1.1.4. Raccourci vers une exigence faisant l'objet d'un avertissement
    - 5.1.1.2. Gestion de la synchronisation
      - 5.1.1.2.1. Atteinte d'une exigence à partir de l'arbre global
      - 5.1.1.2.2. Atteinte d'une exigence à partir d'un regroupement
        - 5.1.1.2.2.a) L'exigence cible est interne au regroupement
        - 5.1.1.2.2.b) L'exigence cible est externe au regroupement et aucune autre fenêtre n'est affichée
        - 5.1.1.2.2.c) L'exigence cible est externe au regroupement et une autre fenêtre est affichée
      - 5.1.1.2.3. Synchronisation de tous les types
    - 5.1.1.3. Gestion de la génération
      - 5.1.1.3.1. Génération d'un document regroupant l'ensemble des exigences synchronisées
      - 5.1.1.3.2. Omission des sections Autres fonctions dans l'arbre global
    - 5.1.1.4. Gestion des fenêtres de regroupements

- 5.1.1.4.1. Affichage d'autant de fenêtres que l'on souhaite
- 5.1.1.4.2. Affichage des fenêtres
  - 5.1.1.4.2.a) Aucun regroupement créé
  - 5.1.1.4.2.b) Au moins un regroupement créé
  - 5.1.1.4.2.c) Une seule fenêtre affichée
  - 5.1.1.4.2.d) Plusieurs fenêtres affichées
- 5.1.1.5. Gestion des exigences entre les regroupements
  - 5.1.1.5.1. Numéro hiérarchique des exigences d'un regroupement
  - 5.1.1.5.2. Création de la section Autre fonctions
    - 5.1.1.5.2.a) Création automatique de la section Autres fonctions
    - 5.1.1.5.2.b) Interdiction de la création de la section Autres fonctions si elle est déjà présente

## 1. EXIGENCES SPÉCIFIQUES INITIALES

La gestion multi-documents doit fournir les moyens de gérer un tableau d'attribution des exigences à des composants ainsi que gérer la centralisation des informations au sein d'un projet.

Il en découle les exigences suivantes :

- 1) Exigences fonctionnelles

Réf. besoin client : [2] 03 p.01.

### **1.1. Exigences fonctionnelles**

La gestion multi-documents doit être capable de réaliser un tableau d'attribution des exigences à des composants et de centraliser les informations d'un projet au sein d'une seule base de données.

Il en découle les exigences suivantes :

- 1) Tableau d'attribution des exigences à des composants
- 2) Centralisation des informations
- 3) Gestion de la cohérence avec l'environnement extérieur

### 1.1.1. Centralisation des informations

La gestion multi-documents doit permettre de centraliser les informations d'un projet dans une seule base de données.

Il en découle les exigences suivantes :

- 1) Gestion des regroupements
- 2) Gestion de la vérification
- 3) Opérations de déplacement d'une exigence
- 4) Gestion de la synchronisation
- 5) Gestion de la génération
- 6) Modification de la fonctionnalité Exigences générées et applicables
- 7) Gestion des fenêtres de regroupements
- 8) Gestion des exigences entre les regroupements

#### 1.1.1.1. Gestion des regroupements

NOTE – Un regroupement d'exigences consiste en un arbre hiérarchique représentant une ou plusieurs exigences.

##### 1.1.1.1.1. Création d'un regroupement

La gestion multi-documents doit permettre de créer un composant dans le niveau sélectionné :

- a) Nom unique. Il faut donner un nom unique pour le regroupement ;
- b) Contenu du regroupement. Le regroupement ne contient aucune exigence ;
- c) Numéro identifiant. Le regroupement possède un numéro identifiant qui permet de l'identifier afin d'effectuer des renvois vers ce regroupement ; NOTE – Un renvoi vers une exigence appartenant à un autre regroupement que celui de l'exigence qui effectue le renvoi est réalisé en donnant à l'exigence source du renvoi l'identifiant du regroupement possédant l'exigence cible du renvoi.
- d) Création des paramètres de génération. Lors de la création d'un regroupement, il faut créer un nouvel onglet qui reprend la fenêtre 'Progression de la génération du document Word' existante dans GenSpec. Il y a donc autant d'onglets présentant cette fenêtre qu'il y a de regroupements (l'arbre global étant compris comme un regroupement).

##### 1.1.1.1.2. Modification du nom d'un regroupement

La gestion multi-documents doit permettre de renommer un regroupement. Le numéro identifiant reste néanmoins identique :

- a) Nom unique. Il faut donner un nom unique pour le regroupement.

##### 1.1.1.1.3. Suppression d'un regroupement

La gestion multi-documents doit permettre de supprimer un regroupement.

NOTE – Supprimer un regroupement ne supprime pas les exigences qui y sont associées, mais uniquement les liens vers ces dernières.

Les exigences appartiennent toujours à l'arbre global.

#### 1.1.1.1.4. Propriétés des arbres hiérarchiques

La gestion multi-documents doit assurer qu'à tout moment, l'arbre global représente l'ensemble des informations présentes dans tous les regroupements :

- a) Racines de l'arbre global. L'arbre global peut posséder plusieurs racines distinctes ;
- b) Racine d'un regroupement. Un regroupement possède au plus une racine ;
- c) Correspondance entre un regroupement et l'arbre global. Toute information enregistrée dans un regroupement doit être enregistrée dans l'arbre global ; NOTE – Le regroupement en question étant un raccourci vers une racine de l'arbre global et ses enfants.
- d) Correspondance entre l'arbre global et un regroupement. Toute information enregistrée dans l'arbre global est enregistrée dans l'éventuel regroupement qui est associé à l'exigence enregistrée ; NOTE – Le regroupement en question étant un raccourci vers une racine de l'arbre global et ses enfants.
- e) Pas de limitation quant aux nombre de regroupements à créer. Le nombre de regroupements autorisé n'est pas borné.

NOTE – L'arbre global est le regroupement sous forme d'arbre hiérarchique des autres regroupements du projet.

Un regroupement est considéré comme une vue d'une racine de l'arbre global avec l'ensemble de ses enfants.

#### 1.1.1.2. Gestion de la vérification

NOTE – Les moyens peuvent être une interface graphique rendant compte de la vérification.

##### 1.1.1.2.1. Comptage et listage des exigences de format ne faisant aucun renvoi vers une exigence

La gestion multi-documents doit permettre de compter et de lister les exigences de format ne faisant aucun renvoi vers des exigences.

NOTE – Il s'agit d'erreurs.

##### 1.1.1.2.2. Comptage et listage des exigences ne faisant aucun renvoi vers une exigence de format

La gestion multi-documents doit permettre de compter et de lister les exigences n'ayant aucun renvoi vers une exigence de format ou vers les documents du client (ou exigences sources externes).

NOTE – Il s'agit d'avertissements.

##### 1.1.1.2.3. Raccourci vers une exigence faisant l'objet d'une erreur

La gestion multi-documents doit permettre d'atteindre une exigence qui fait l'objet d'une erreur.

NOTE – Une erreur correspond à une exigence de format qui ne possède pas de renvoi vers une exigence.

##### 1.1.1.2.4. Raccourci vers une exigence faisant l'objet d'un avertissement

La gestion multi-documents doit permettre d'atteindre une exigence qui fait l'objet d'un avertissement.

NOTE – Un avertissement correspond à une exigence qui ne possède pas de renvoi vers une exigence de format.

#### **1.1.1.3. Opérations de déplacement d'une exigence**

- a) Couper avec la souris. Couper une exigence en utilisant le déplacement avec la souris ;
- b) Ajout dans le menu contextuel au même niveau. Cette possibilité doit être ajoutée dans le menu contextuel relatif à un noeud de l'arbre hiérarchique des exigences ;
- c) Couper avec le raccourci clavier. Couper une exigence en utilisant le raccourci du clavier 'Ctrl+X' ;
- d) Couper avec le menu contextuel. Couper une exigence en utilisant le menu contextuel permettant de couper ;
- e) Copier avec le raccourci clavier. Copier une exigence en utilisant le raccourci du clavier 'Ctrl+C' ;
- f) Copier avec le menu contextuel. Copier une exigence en utilisant le menu contextuel permettant de copier ;
- g) Coller. Coller une exigence et ses enfants sous une exigence ;
- h) Coller avec synchronisation. Coller une exigence et ses enfants sous une exigence en les synchronisant ;
- i) Coller au même niveau. Coller une exigence et ses enfants au même niveau que l'exigence sélectionnée pour le collage ;
- j) Coller au même niveau avec synchronisation. Coller une exigence avec synchronisation et ses enfants au même niveau que l'exigence sélectionnée pour le collage. Les enfants de l'exigence sont synchronisés également ;
- k) Coller avec liaison et synchronisation au même niveau. Coller une exigence avec synchronisation et liaison et ses enfants au même niveau que l'exigence sélectionnée pour le collage. Les enfants de l'exigence sont synchronisés et liés également.

NOTE – Il se peut que certaines des exigences listées ci-dessous soient présentes dans GenSpec.

#### **1.1.1.4. Gestion de la synchronisation**

NOTE – La responsabilité de la cohérence entre des exigences synchronisées en revient à l'administrateur ou à l'utilisateur, étant donné qu'aucune vérification n'est effectuée.

##### **1.1.1.4.1. Atteinte d'une exigence à partir de l'arbre global**

La gestion multi-documents doit permettre d'atteindre une exigence synchronisée à partir de l'arbre global. L'atteinte se fait à l'intérieur de l'arbre global.

NOTE – Atteindre une exigence signifie qu'elle soit automatiquement sélectionnée et ses propriétés affichées. Ces exigences sont atteintes par le menu contextuel présent dans 'Provient de' et 'Utilisés par'.

##### **1.1.1.4.2. Atteinte d'une exigence à partir d'un regroupement**

La gestion multi-documents doit permettre d'atteindre une exigence synchronisée qui n'apparaît pas dans le même regroupement. Les différentes possibilités sont les suivantes :

- a) L'exigence cible est interne au regroupement. L'atteinte se fait à l'intérieur du regroupement ;



- b) L'exigence cible est externe au regroupement et aucune autre fenêtre n'est affichée. Une nouvelle fenêtre est affichée automatiquement et l'atteinte se fait sur l'arbre global de cette fenêtre ; NOTE – L'ouverture d'une fenêtre supplémentaire permet de visualiser l'exigence cible tout en gardant une vue sur l'exigence source.
- c) L'exigence cible est externe au regroupement et une autre fenêtre est affichée. Une fenêtre doit afficher l'onglet de l'arbre global et l'atteinte se fait sur l'arbre global de cette fenêtre. Le choix de la fenêtre pour montrer l'exigence cible est tel qu'il s'agit d'une fenêtre différente de celle d'où provient la demande d'atteinte. NOTE – Cela permet de visualiser l'exigence cible tout en gardant une vue sur l'exigence source.

NOTE – Atteindre une exigence signifie qu'elle soit automatiquement sélectionnée et ses propriétés affichées.

#### **1.1.1.4.3. Synchronisation de tous les types**

La gestion multi-documents doit permettre de synchroniser tout type d'exigence.

NOTE – La responsabilité de la cohérence entre des exigences synchronisées en revient à l'administrateur ou à l'utilisateur, étant donné qu'aucune vérification n'est effectuée.

#### **1.1.1.5. Gestion de la génération**

##### **1.1.1.5.1. Génération d'un document regroupant l'ensemble des exigences**

La gestion multi-documents doit permettre de générer des documents regroupant l'ensemble des exigences du projet. Il s'agit de la génération de l'arbre global.

NOTE – Les documents qu'il est possible de générer sont ceux que propose l'outil GenSpec.

##### **1.1.1.5.2. Génération d'un document regroupant l'ensemble des exigences synchronisées**

La gestion multi-documents doit permettre de générer un nouveau tableau dans Microsoft Word regroupant les exigences synchronisées. Les différentes colonnes de ce tableau sont les suivantes : ID de l'exigence – Occurrence – 'No paraph' – Titre de l'exigence.

NOTE – Occurrence est le nom du regroupement qui contient l'exigence synchronisée en question.

'No paraph' est le numéro de paragraphe de l'exigence dans le regroupement correspondant.

L'accès à la génération de ce tableau est intégré dans la liste de choix de document à générer présente dans GenSpec.

##### **1.1.1.5.3. Génération d'un document regroupant les exigences d'un regroupement**

La gestion multi-documents doit permettre de générer des documents regroupant l'ensemble des exigences d'un regroupement.

NOTE – Les documents qu'il est possible de générer sont ceux que propose l'outil GenSpec.

##### **1.1.1.5.4. Réutilisation des options de génération existantes**

La gestion multi-documents doit permettre de réutiliser les options de génération présentes dans GenSpec.

#### **1.1.1.5.5. Omission des sections Autres fonctions dans l'arbre global**

La gestion multi-documents doit omettre toutes les sections Autres fonctions de l'arbre global lors de la génération de ce dernier.

NOTE – Cette omission est souhaitable car la génération de la section Autres fonctions n'apporte aucune information pertinente, l'information étant présente dans l'arbre global.

#### **1.1.1.5.6. Un onglet par regroupement**

La gestion multi-documents doit permettre que la fenêtre 'Progression de la génération du document Word' existante dans GenSpec puisse être utilisée pour chaque regroupement.

NOTE – Une fois la fenêtre 'Progression de la génération du document Word' affichée, il est possible de générer tous les regroupements en parcourant les onglets uns à uns.

#### **1.1.1.6. Modification de la fonctionnalité Exigences générées et applicables**

NOTE – Un groupe d'exigences est un 'groupe d'exigences générées' au sens de GenSpec.

##### **1.1.1.6.1. Création d'un fichier par groupe [O]**

La gestion multi-documents doit permettre de générer des documents séparés pour chaque groupe d'exigences. Le nom donné pour le document Microsoft Word contiendra entre autre le nom du groupe qui y est associé.

##### **1.1.1.6.2. Gestion les liaisons [O]**

La gestion multi-documents doit permettre d'importer les intrants, extrants et renvois d'une exigence lors de l'association de cette dernière à un groupe :

- a) Intrants [O]. Importer les intrants de l'exigence ;
- b) Extrants [O]. Importer les extrants de l'exigence.

##### **1.1.1.6.3. Avertissement d'un changement automatique dans un groupe**

La gestion multi-documents doit avertir qu'un ou plusieurs groupes vont subir des modifications suite à la dernière opération effectuée.

#### **1.1.1.7. Gestion des fenêtres de regroupements**

NOTE – Dans les exigences ci-dessous, nous appellerons une fenêtre, une fenêtre de regroupements d'arbres hiérarchiques.

##### **1.1.1.7.1. Affichage d'autant de fenêtres que l'on souhaite**

La gestion multi-documents doit permettre d'afficher autant de fenêtres de regroupement que l'on souhaite.

##### **1.1.1.7.2. Affichage des fenêtres**

La gestion multi-documents doit permettre d'afficher les différents regroupements d'un projet dans les fenêtres. Les fenêtres sont ordonnées selon les conditions suivantes :

- a) Aucun regroupement créé. L'arbre hiérarchique est créé dans la fenêtre sans appartenir à un onglet ;

- b) Au moins un regroupement créé. Chaque arbre hiérarchique est représenté dans un onglet différent ;
- c) Une seule fenêtre affichée. La fenêtre est nommée 'Arbre hiérarchique' ou 'Arbres hiérarchiques' en fonction du nombre d'onglets présents dans la fenêtre ;
- d) Plusieurs fenêtres affichées. Le nom des fenêtres doit être différent. Leur nom est différent de par l'utilisation d'un suffixe. NOTE – Par exemple, si deux fenêtres sont affichées, elles s'appellent respectivement 'Arbres hiérarchiques 1' et 'Arbres hiérarchiques 2' (dans le cas où plusieurs onglets sont créés)

#### **1.1.1.8. Gestion des exigences entre les regroupements**

NOTE – Un déplacement interne est le déplacement d'une exigence à l'intérieur de son regroupement.

Un déplacement externe est le déplacement d'une exigence à l'extérieur de son regroupement.

##### **1.1.1.8.1. Numéro hiérarchique des exigences d'un regroupement**

Les numéros hiérarchiques des exigences d'un regroupement doivent correspondre aux numéros hiérarchiques de ces mêmes exigences dans l'arbre global.

##### **1.1.1.8.2. Création de la section Autres fonctions**

La gestion multi-documents doit permettre de convertir les liaisons externes en liaisons internes à l'intérieur d'un regroupement :

- a) Création automatique de la section Autres fonctions. Afin de convertir une liaison externe en liaison interne, il faut créer automatiquement une section Autres fonctions qui permet d'insérer l'exigence faisant l'objet d'une liaison externe, de sorte à rendre cette exigence interne. L'organisation de la section Autres fonctions est la suivante : Niveau Autres fonctions. Niveau Intrants, Extrants. Niveau des exigences intrants et extrants. La provenance de l'intrant ou de l'extrant est notée entre parenthèses comme suffixe au titre de l'exigence ; NOTE – La provenance est le titre de la racine dans l'arbre global d'où provient l'exigence faisant l'objet d'une liaison externe.
- b) Interdiction de la création de la section Autres fonctions si elle est déjà présente. Si la section Autres fonction est présente dans un regroupement, il est interdit d'en créer une autre. Les ajouts manuels d'exigences relatifs à des autres fonctions doivent être insérés dans cette section existante.

NOTE – Une liaison interne est une liaison entre des exigences qui se trouvent dans le même regroupement. Une liaison externe est une liaison entre des exigences qui se trouvent dans des regroupements distincts.

## **2. EXIGENCES SPÉCIFIQUES RELATIVES À LA PREMIÈRE ITÉRATION**

Le produit doit fournir les moyens de gérer un tableau d'attribution des exigences à des composants ainsi que gérer la centralisation des informations au sein d'un projet.

Il en découle les exigences suivantes :

- 1) Exigences fonctionnelles

## **2.1. Exigences fonctionnelles**

Le produit doit être capable de réaliser un tableau d'attribution des exigences à des composants et de centraliser les informations d'un projet au sein d'une seule base de données.

Il en découle les exigences suivantes :

- 1) Tableau d'attribution des exigences à des composants
- 2) Gestion de la cohérence avec l'environnement extérieur

### **2.1.1. Tableau d'attribution des exigences à des composants**

Le produit doit permettre différentes gestions relatives au tableau d'attribution des exigences à des composants.

Il en découle les exigences suivantes :

- 1) Gestion des composants
- 2) Gestion des fonctions
- 3) Options et génération du tableau
- 4) Gestion de l'interface graphique
- 5) Gestion des accès avec l'interface graphique de GenSpec
- 6) Conversion des attributions existantes
- 7) Gestion de la validation

NOTE – Les composants sont classés sous un format hiérarchique. Nous appelons donc dans cette section 'arbre', l'arbre hiérarchique des composants.

#### **2.1.1.1. Gestion des composants**

Le produit doit gérer les composants qui comprennent les exigences suivantes :

##### **2.1.1.1.1. Création d'un composant**

Le produit doit permettre de créer un composant dans un niveau de l'arbre :

- a) Nom unique. Il faut donner un nom unique pour le composant dans son niveau ; NOTE – Des composants dans des niveaux différents peuvent posséder le même nom.
- b) Contenu du composant. Le composant ne contient aucune attribution lors de sa création. Le composant est ajouté dans l'arbre dans le niveau sélectionné pour sa création.

NOTE – Le niveau correspond à la profondeur du composant dans l'arbre.

##### **2.1.1.1.2. Modification du nom d'un composant**

Le produit doit permettre de renommer un composant, c'est à dire changer le nom du composant, sans que ce dernier ne figure dans la liste des composants du même niveau.

##### **2.1.1.1.3. Suppression d'un composant**

Le produit doit permettre de supprimer un composant et toutes les attributions qui y sont associées.

##### **2.1.1.1.4. Attribution des exigences**

Le produit doit permettre d'attribuer des exigences à des composants ou des fonctions. L'attribution d'une exigence se réalise en déplaçant l'exigence :

- a) Déplacement vers l'espace réservé aux attributions. Représente l'espace des attributions associé au composant sélectionné ; NOTE – L'espace réservé aux attributions est la liste contenant les attributions d'un composant ou d'une fonction.
- b) Cohérence avec les composants parents. Toute attribution d'exigences dans un composant implique une attribution automatique de ces exigences dans tous les composants parents du composant.

NOTE – Déplacer une exigence, et donc en créer une attribution, consiste à la faire glisser d'un arbre hiérarchique d'exigences vers la fenêtre d'attribution des exigences à des composants.

#### **2.1.1.1.5. Suppression d'une attribution**

Le produit doit permettre de supprimer une ou plusieurs attributions d'un composant :

- a) Suppression de l'attribution seule. Supprimer l'attribution sélectionnée du composant ;
- b) Suppression de l'attribution avec ses enfants et parents. Supprimer l'attribution sélectionnée du composant, ainsi que ses parents et enfants.

#### **2.1.1.1.6. Organisation des composants en niveaux**

Le produit doit permettre d'établir une hiérarchie de composants :

- a) Arbre hiérarchique de composants. Les composants sont classés par l'utilisation d'un arbre hiérarchique, ce dernier autorise que plusieurs composants en soient les racines ;
- b) Niveaux des composants. On peut disposer d'autant de niveaux que l'on souhaite, le premier niveau ayant une profondeur égale à 1.

#### **2.1.1.2. Gestion des fonctions**

Le produit doit gérer les fonctions qui comprennent les exigences suivantes :

##### **2.1.1.2.1. Même exigences pour la gestion des fonctions que pour la gestion des composants**

Le produit doit répondre aux exigences suivantes qui sont présentes dans la gestion des composants : Attribution des exigences, modification du nom d'un composant, suppression d'un composant, affichage du niveau de composants, suppression d'une attribution, envoi vers un composant fils ou une fonction, la sélection d'attributions et l'importation des exigences liées lors de l'attribution.

##### **2.1.1.2.2. Création d'une fonction**

Le produit doit permettre de créer une fonction dans l'espace réservé aux fonctions :

- a) Nom unique. Il faut donner un nom unique pour la fonction ;
- b) Contenu de la fonction. La fonction ne contient aucune attribution lors de sa création. Le nom de la fonction est présent dans la liste des fonctions ;
- c) Niveau des fonctions. Une fonction ne possède qu'un seul niveau, 'le niveau des fonctions'.

#### **2.1.1.3. Options et génération du tableau**

Le produit doit permettre de générer un tableau qui représente l'ensemble des attributions effectuées.

##### **2.1.1.3.1. Format de sortie du document**

Le produit doit permettre de choisir le format de sortie du document :

- a) Microsoft Excel. Générer le document dans le format Microsoft Excel.

#### 2.1.1.3.2. Organisation automatique des colonnes du tableau

Le produit doit permettre, avant la génération, de choisir parmi deux formats de colonnes prédéfinies :

- a) Numéro hiérarchique – Titre – ID – Fonction – Composant (niveau 1) – ... – Composant (niveau n). L'ordre des colonnes dans le tableau. Un tri alphanumérique croissant est effectué sur la première colonne (le numéro hiérarchique) ;
- b) ID – Numéro hiérarchique – Titre – Fonction – Composant (niveau 1) – ... – Composant (niveau n). L'ordre des colonnes dans le tableau. Un tri alphabétique croissant est effectué sur la première colonne (l'ID).

#### 2.1.1.3.3. Regroupement fonctionnel

Le produit doit permettre qu'un regroupement fonctionnel soit autorisé ou non :

- a) Un regroupement fonctionnel est souhaité. Dans la fenêtre des options, le choix du niveau pour les fonctions doit être grisé ; NOTE – En effet, les fonctions définies dans le niveau des fonctions serviront à la colonne 'Fonction' dans le tableau généré, et par conséquent, il n'est pas nécessaire de choisir un niveau dans l'arbre hiérarchique des exigences.
- b) Un regroupement fonctionnel n'est pas souhaité. Choisir le niveau de l'arbre hiérarchique qui servira pour la fonction lié à l'exigence. La fonction de l'exigence est alors le titre de l'exigence du parent de niveau sélectionné ; NOTE – Les fonctions définies dans le niveau des fonctions ne sont donc pas représentées lors de la génération.
- c) Génération si un regroupement fonctionnel est souhaité. La colonne 'fonction' du tableau à générer doit correspondre aux fonctions présentes dans le niveau des fonctions ;
- d) Génération si un regroupement fonctionnel n'est pas souhaité. La colonne 'fonction' du tableau à générer doit correspondre à l'exigence parent du niveau sélectionné dans la fenêtre des options. Il correspond au niveau du parent de l'exigence – source de l'attribution – dans l'arbre hiérarchique des exigences. Le titre de l'exigence du parent est alors utilisé comme fonction. NOTE – Le niveau des fonctions n'est donc pas pris en compte pour la génération.

NOTE – Un regroupement fonctionnel détermine si les fonctions définies dans le niveau des fonctions seront utilisées ou non lors de la génération du tableau. En effet, il n'est pas toujours nécessaire d'utiliser le niveau des fonctions pour évoquer la fonction d'une attribution, le titre d'une exigence parent peut suffire à en définir la fonction.

#### 2.1.1.4. Gestion de l'interface graphique

Le produit doit fournir les moyens visuels pour effectuer des opérations, naviguer dans l'interface graphique et faciliter les tâches d'attribution des exigences à des composants et des fonctions.

##### 2.1.1.4.1. Menu contextuel pour les attributions

Le produit doit fournir un menu contextuel permettant de réaliser les opérations listées ci-dessous :

- a) Valider cette attribution. Valide une attribution dont l'exigence a subi une modification ;
- b) Suppression de l'attribution seule. Supprime l'attribution sélectionnée ;
- c) Suppression de l'attribution avec ses enfants et parents. Supprimer l'attribution sélectionnée du composant, ainsi que ses parents et enfants.

NOTE – Un menu contextuel apparaît lors d'un clic sur le bouton droit de la souris et donne les options relatives à l'élément sélectionné.

#### **2.1.1.4.2. Menu contextuel pour les composants**

Le produit doit fournir un menu contextuel permettant de réaliser les opérations listées ci-dessous :

- a) Ajout sous un composant. Ajoute un composant sous le composant sélectionné ; NOTE – Cette exigence n'est applicable que pour les composants.
- b) Ajout d'un composant au même niveau. Ajoute un composant au même niveau que le composant sélectionné ;
- c) Modification du nom. Modifie le nom du composant ;
- d) Suppression. Supprime le composant.

NOTE – Un menu contextuel apparaît lors d'un clic sur le bouton droit de la souris et donne les options relatives au composant sélectionné.

#### **2.1.1.4.3. Menu contextuel pour les fonctions**

Le produit doit fournir un menu contextuel permettant de réaliser les opérations listées ci-dessous :

- a) Ajout d'une fonction. Ajoute une fonction à la liste des fonctions ;
- b) Modification du nom. Modifie le nom de la fonction ;
- c) Suppression. Supprime la fonction.

NOTE – Un menu contextuel apparaît lors d'un clic sur le bouton droit de la souris et donne les options relatives à la fonction sélectionnée.

#### **2.1.1.4.4. Choix d'affichage**

Le produit doit permettre de choisir parmi un des deux types d'affichages proposés, et ce dans la fenêtre d'attribution des exigences à des composants. Ces deux types sont les suivants :

- a) Les composants et les attributions. Affiche l'arbre des composants et les attributions au composant sélectionné ;
- b) Les fonctions et les attributions. Affiche la liste des fonctions et les attributions à la fonction sélectionnée.

#### **2.1.1.5. Gestion des accès avec l'interface graphique de GenSpec**

Le produit doit être accessible à partir de l'interface graphique de GenSpec.

##### **2.1.1.5.1. Accès à la fenêtre des attributions**

Le produit doit permettre d'accéder à la fenêtre permettant de réaliser des attributions à partir d'un nouveau menu de GenSpec qui nommé 'Outils' :

- a) Menu Outils. Ce nouveau menu contient une seule entrée permettant d'accéder à la fenêtre en question ; NOTE – GenSpec possèdera donc un nouveau menu 'Outils', ainsi qu'une fenêtre existante 'Outils'.
- b) Position du menu Outils. Ce menu doit être situé dans l'interface graphique de GenSpec de la manière suivante : Fichier, Edition, Génération, Session, Outils, Fenêtres et Aide.

##### **2.1.1.5.2. Accès à la fenêtre 'Progression de la génération du document Excel'**

Le produit doit fournir une nouvelle fenêtre permettant de mettre en évidence l'avancement de la génération du tableau d'attribution des exigences à des composants :



- a) Nouvelle fenêtre. Cette nouvelle fenêtre doit être semblable à celle utilisée pour la génération de document Microsoft Word existante, sauf qu'elle s'applique pour générer dans le format Excel ;
- b) Contenu de la liste 'Documents à générer'. Cette fenêtre propose dans la liste 'Document à générer', une seule entrée qui est le tableau d'attribution des exigences à des composants ;
- c) Bouton Options de génération. Cette fenêtre doit disposer d'un bouton permettant d'accéder aux options de génération du tableau ;
- d) Position du bouton des Options de génération. Ce bouton doit être situé à côté du bouton 'Fermer' existant dans la fenêtre similaire, dans le bas de la fenêtre. NOTE – La fenêtre similaire est 'Progression de la génération du document Word'.

#### **2.1.1.5.3. Accès à la fenêtre des options de génération du tableau**

Le produit doit permettre d'accéder à la fenêtre des options de génération à partir du bouton 'Options de génération'.

#### **2.1.1.5.4. Réorganisation des icônes**

Le produit doit réorganiser les icônes existantes de sorte à supporter les nouvelles fonctionnalités liées au développement d'une gestion multi-documents :

- a) Bouton Excel. Disposer d'un nouveau bouton dans la fenêtre 'Outils' permettant d'accéder à la fenêtre 'Progression de la génération du document Excel' ;
- b) Icône Excel. Le bouton Excel doit disposer de l'icône représentant Microsoft Excel.

NOTE – Cette réorganisation porte sur la fenêtre 'Outils' existante dans GenSpec.

#### **2.1.1.6. Conversion des attributions existantes**

Le produit doit permettre d'importer les attributions réalisées antérieurement et de les intégrer dans la base de données (version 6.4.25) et ce, par l'utilisation d'une nouvelle application, indépendante de GenSpec.

##### **2.1.1.6.1. Intrants de l'application**

Le produit doit réaliser la conversion grâce aux fichiers suivants :

- a) La table contenant les attributions existantes. Permettre de choisir la base de données contenant les attributions existantes ; NOTE – Le nom de la table contenant les attributions existantes est 'AssociationApplComp'.
- b) Les tables listant les fonctions, les applications et les composants existants. Permettre de choisir la base de données contenant les fonctions, les applications et les composants existants.

##### **2.1.1.6.2. Méthodologie d'importation**

Le produit doit respecter, dans l'ordre, les étapes suivantes :

- a) Convertir la base de données. En utilisant GenSpec, mettre à jour la base de données contenant les attributions existantes ;
- b) Choisir la base de données contenant les fonctions, applications et composants. Dans la nouvelle application, choisir la base de données contenant les informations relatives aux fonctions, composants et applications ;
- c) Choisir la base de données contenant les attributions existantes. Dans la nouvelle application, choisir la base de données contenant les attributions existantes ;

- d) Intégrer les attributions dans la gestion multi-documents. Intégrer les attributions existantes dans les tables relatives à l'attribution des exigences à des composants/fonctions ;
- e) Supprimer les tables relatives aux attributions existantes. Supprimer la table AssociationApplComp, ainsi que les tables ListeFonctions, ListeComposants et ListeApplications présentes dans la base de données contenant les attributions existantes.

#### **2.1.1.6.3. Règles de conversion**

Le produit doit respecter les règles de conversion suivantes :

- a) Fonction. Une fonction présente dans la liste des fonctions existantes correspond à une fonction au sens de la gestion multi-documents ;
- b) Application. Une application présente dans la liste des applications existantes correspond à un composant de niveau 1 au sens de la gestion multi-documents ;
- c) Composant. Un composant présent dans la liste des composants existants correspond à un composant de niveau 2 au sens de la gestion multi-documents.

#### **2.1.1.6.4. Génération du tableau de conception**

Le produit doit permettre d'intégrer les aspects suivants dans le document Excel généré en fonction des fonctionnalités supportées par GenSpec :

- a) Sans la centralisation des informations au sein de GenSpec. Le document généré doit contenir une colonne supplémentaire nommée 'SES' et dont le contenu, identique pour chaque ligne, est l'information présente dans la colonne 'PROD' de la table 'Params' et ce, dans la base de données de la spécification relative au tableau de conception ; NOTE – Cette information donne le nom de la spécification. Par exemple : 'SBC', 'TC'.
- b) Avec la centralisation des informations au sein de GenSpec. Le document généré ne doit pas contenir de colonne nommée 'SES'. NOTE – Cette information n'est pas nécessaire lorsque l'ensemble des exigences de toutes les spécifications est présent dans un seul projet GenSpec.

#### **2.1.1.7. Gestion de la validation**

Le produit doit fournir une validation pour les composants, fonctions et les attributions.

##### **2.1.1.7.1. Modification d'une exigence**

Le produit doit mettre en évidence les attributions dont les exigences sources ont subi une modification :

- a) Libellé des attributions. Les attributions ayant subi une modification doivent posséder le suffixe : "[\*]".

##### **2.1.1.7.2. Suppression d'une exigence**

Le produit doit en cas de suppression d'une exigence, supprimer automatiquement toutes les attributions – relatives à cette exigence – présentes dans les fonctions et les composants.

##### **2.1.1.7.3. Validation des attributions**

Le produit doit permettre de valider une attribution par l'utilisation d'un menu contextuel relatif à cette attribution :

- a) Libellé du menu contextuel. Le libellé du menu contextuel pour la validation d'une attribution est le suivant : 'Valider cette attribution'.

### **2.1.2. Gestion de la cohérence avec l'environnement extérieur**

Le produit doit pouvoir intégrer l'ensemble des fonctionnalités liées à la gestion multi-documents tout en maintenant à jour l'aide et la compatibilité avec les versions antérieures de la base de données.

Il en découle les exigences suivantes :

- 1) Aide
- 2) Mise à jour de la base de données
- 3) Usager concepteur
- 4) Cohésion avec les fonctionnalités existantes

#### **2.1.2.1. Aide**

Le produit doit fournir une aide qui permet de comprendre son utilisation. L'aide de la gestion multi-documents fait partie de l'aide de GenSpec.

#### **2.1.2.2. Mise à jour de la base de données**

Le produit doit gérer les versions précédentes de la base de données de GenSpec, suite aux modifications engendrées par les fonctionnalités relatives à la gestion multi-documents.

#### **2.1.2.3. Usager concepteur**

Le produit doit supporter un nouveau type d'utilisateur nommé "concepteur" au sein de GenSpec.

#### **2.1.2.4. Cohésion avec les fonctionnalités existantes**

Le produit doit se baser sur les principes existants dans 'Groupe d'exigences générées et applicables', et ce afin de garder une cohésion entre les nouvelles et les fonctionnalités existantes de GenSpec :

- a) Menus contextuels pour les fonctions et les composants. Le menu contextuel pour les composants et les fonctions doit être semblable à celui utilisé dans 'Groupe' ;
- b) Menus contextuels pour les attributions. Le menu contextuel pour les attributions doit être semblable à celui utilisé dans 'Exigences contenues dans le groupe'.

### **3. EXIGENCES SPÉCIFIQUES RELATIVES À LA DEUXIÈME ITÉRATION**

La gestion multi-documents doit fournir les moyens de gérer un tableau d'attribution des exigences à des composants ainsi que gérer la centralisation des informations au sein d'un projet.

Il en découle les exigences suivantes :

- 1) Exigences fonctionnelles

### **3.1. Exigences fonctionnelles**

La gestion multi-documents doit être capable de réaliser un tableau d'attribution des exigences à des composants et de centraliser les informations d'un projet au sein d'une seule base de données.

Il en découle les exigences suivantes :

- 1) Centralisation des informations

### **3.1.1. Centralisation des informations**

La gestion multi-documents doit permettre de centraliser les informations d'un projet dans une seule base de données.

Il en découle les exigences suivantes :

- 1) Gestion des regroupements
- 2) Opérations de déplacement d'une exigence
- 3) Gestion de la génération
- 4) Cohérence avec les fonctionnalités existantes

#### **3.1.1.1. Gestion des regroupements**

NOTE – Un regroupement d'exigences consiste en un arbre hiérarchique représentant une ou plusieurs exigences.

##### **3.1.1.1.1. Propriétés**

La gestion multi-documents doit assurer qu'à tout moment, l'arbre global contient au moins l'ensemble des informations présentes dans tous les regroupements :

- a) Racines de l'arbre global. L'arbre global peut posséder plusieurs racines.

#### **3.1.1.2. Opérations de déplacement d'une exigence**

##### **3.1.1.2.1. Coller au même niveau**

La gestion multi-documents doit pouvoir coller une exigence et ses enfants au même niveau que l'exigence sélectionnée pour le collage :

- a) Ajout dans le menu contextuel. Cette possibilité doit être ajoutée dans le menu contextuel relatif à un noeud de l'arbre hiérarchique des exigences.

##### **3.1.1.2.2. Ajouter au même niveau que cette exigence**

La gestion multi-documents doit pouvoir ajouter une exigence au même niveau que l'exigence sélectionnée pour l'ajout :

- a) Ajout dans le menu contextuel. Cette possibilité doit être ajoutée dans le menu contextuel relatif à un noeud de l'arbre hiérarchique des exigences.

#### **3.1.1.3. Gestion de la génération**

##### **3.1.1.3.1. Génération d'un document regroupant l'ensemble des exigences**

La gestion multi-documents doit permettre de générer des documents regroupant l'ensemble des exigences contenues dans l'arbre global.

NOTE – Les documents qu'il est possible de générer sont ceux que propose l'outil GenSpec.

#### **3.1.1.4. Cohérence avec les fonctionnalités existantes**

##### **3.1.1.4.1. Fonctionnalités présentes dans le menu Outils**

La gestion multi-documents doit s'assurer que les fonctionnalités présentes dans le menu 'Outils' soient fonctionnelles et ce, malgré la présence de plusieurs racines dans l'arbre global.

#### **4. EXIGENCES SPÉCIFIQUES RÉALISÉES**

La gestion multi-documents doit fournir les moyens de gérer un tableau d'attribution des exigences à des composants ainsi que gérer la centralisation des informations au sein d'un projet.

Il en découle les exigences suivantes :

- 1) Exigences fonctionnelles

Réf. besoin client : [2] 03 p.01.

#### **4.1. Exigences fonctionnelles**

La gestion multi-documents doit être capable de réaliser un tableau d'attribution des exigences à des composants et de centraliser les informations d'un projet au sein d'une seule base de données.

Il en découle les exigences suivantes :

- 1) Tableau d'attribution des exigences à des composants
- 2) Centralisation des informations
- 3) Gestion de la cohérence avec l'environnement extérieur

#### **4.1.1. Centralisation des informations**

La gestion multi-documents doit permettre de centraliser les informations d'un projet dans une seule base de données.

Il en découle les exigences suivantes :

- 1) Gestion des regroupements
- 2) Opérations de déplacement d'une exigence
- 3) Gestion de la génération
- 4) Cohérence avec les fonctionnalités existantes

##### **4.1.1.1. Gestion des regroupements**

NOTE – Un regroupement d'exigences consiste en un arbre hiérarchique représentant une ou plusieurs exigences.

###### **4.1.1.1.1. Propriétés**

La gestion multi-documents doit assurer qu'à tout moment, l'arbre global contient au moins l'ensemble des informations présentes dans tous les regroupements :

- a) Racines de l'arbre global. L'arbre global peut posséder plusieurs racines.

##### **4.1.1.2. Opérations de déplacement d'une exigence**

###### **4.1.1.2.1. Coller au même niveau**

La gestion multi-documents doit pouvoir coller une exigence et ses enfants au même niveau que l'exigence sélectionnée pour le collage :

- a) Ajout dans le menu contextuel. Cette possibilité doit être ajoutée dans le menu contextuel relatif à un noeud de l'arbre hiérarchique des exigences.

###### **4.1.1.2.2. Ajouter au même niveau que cette exigence**

La gestion multi-documents doit pouvoir ajouter une exigence au même niveau que l'exigence sélectionnée pour l'ajout :

- a) Ajout dans le menu contextuel. Cette possibilité doit être ajoutée dans le menu contextuel relatif à un noeud de l'arbre hiérarchique des exigences.

##### **4.1.1.3. Gestion de la génération**

###### **4.1.1.3.1. Génération d'un document regroupant l'ensemble des exigences**

La gestion multi-documents doit permettre de générer des documents regroupant l'ensemble des exigences contenues dans l'arbre global.

NOTE – Les documents qu'il est possible de générer sont ceux que propose l'outil GenSpec.



#### **4.1.1.4. Cohérence avec les fonctionnalités existantes**

##### **4.1.1.4.1. Fonctionnalités présentes dans le menu Outils**

La gestion multi-documents doit s'assurer que les fonctionnalités présentes dans le menu 'Outils' soient fonctionnelles et ce, malgré la présence de plusieurs racines dans l'arbre global.

#### **5. EXIGENCES SPÉCIFIQUES NON RÉALISÉES**

La gestion multi-documents doit fournir les moyens de gérer un tableau d'attribution des exigences à des composants ainsi que gérer la centralisation des informations au sein d'un projet.

Il en découle les exigences suivantes :

- 1) Exigences fonctionnelles

Réf. besoin client : [2] 03 p.01.

### **5.1. Exigences fonctionnelles**

La gestion multi-documents doit être capable de réaliser un tableau d'attribution des exigences à des composants et de centraliser les informations d'un projet au sein d'une seule base de données.

Il en découle les exigences suivantes :

- 1) Tableau d'attribution des exigences à des composants
- 2) Centralisation des informations

### **5.1.1. Centralisation des informations**

La gestion multi-documents doit permettre de centraliser les informations d'un projet dans une seule base de données.

Il en découle les exigences suivantes :

- 1) Gestion de la vérification
- 2) Gestion de la synchronisation
- 3) Gestion de la génération
- 4) Gestion des fenêtres de regroupements
- 5) Gestion des exigences entre les regroupements

#### **5.1.1.1. Gestion de la vérification**

NOTE – Les moyens peuvent être une interface graphique rendant compte de la vérification.

##### **5.1.1.1.1. Comptage et listage des exigences de format ne faisant aucun renvoi vers une exigence**

La gestion multi-documents doit permettre de compter et de lister les exigences de format ne faisant aucun renvoi vers des exigences.

NOTE – Il s'agit d'erreurs.

##### **5.1.1.1.2. Comptage et listage des exigences ne faisant aucun renvoi vers une exigence de format**

La gestion multi-documents doit permettre de compter et de lister les exigences n'ayant aucun renvoi vers une exigence de format ou vers les documents du client (ou exigences sources externes).

NOTE – Il s'agit d'avertissements.

##### **5.1.1.1.3. Raccourci vers une exigence faisant l'objet d'une erreur**

La gestion multi-documents doit permettre d'atteindre une exigence qui fait l'objet d'une erreur.

NOTE – Une erreur correspond à une exigence de format qui ne possède pas de renvoi vers une exigence.

##### **5.1.1.1.4. Raccourci vers une exigence faisant l'objet d'un avertissement**

La gestion multi-documents doit permettre d'atteindre une exigence qui fait l'objet d'un avertissement.

NOTE – Un avertissement correspond à une exigence qui ne possède pas de renvoi vers une exigence de format.

#### **5.1.1.2. Gestion de la synchronisation**

NOTE – La responsabilité de la cohérence entre des exigences synchronisées en revient à l'administrateur ou à l'utilisateur, étant donné qu'aucune vérification n'est effectuée.

#### **5.1.1.2.1. Atteinte d'une exigence à partir de l'arbre global**

La gestion multi-documents doit permettre d'atteindre une exigence synchronisée à partir de l'arbre global. L'atteinte se fait à l'intérieur de l'arbre global.

NOTE – Atteindre une exigence signifie qu'elle soit automatiquement sélectionnée et ses propriétés affichées. Ces exigences sont atteintes par le menu contextuel présent dans 'Provient de' et 'Utilisés par'.

#### **5.1.1.2.2. Atteinte d'une exigence à partir d'un regroupement**

La gestion multi-documents doit permettre d'atteindre une exigence synchronisée qui n'apparaît pas dans le même regroupement. Les différentes possibilités sont les suivantes :

- a) L'exigence cible est interne au regroupement. L'atteinte se fait à l'intérieur du regroupement ;
- b) L'exigence cible est externe au regroupement et aucune autre fenêtre n'est affichée. Une nouvelle fenêtre est affichée automatiquement et l'atteinte se fait sur l'arbre global de cette fenêtre ; NOTE – L'ouverture d'une fenêtre supplémentaire permet de visualiser l'exigence cible tout en gardant une vue sur l'exigence source.
- c) L'exigence cible est externe au regroupement et une autre fenêtre est affichée. Une fenêtre doit afficher l'onglet de l'arbre global et l'atteinte se fait sur l'arbre global de cette fenêtre. Le choix de la fenêtre pour montrer l'exigence cible est tel qu'il s'agit d'une fenêtre différente de celle d'où provient la demande d'atteinte. NOTE – Cela permet de visualiser l'exigence cible tout en gardant une vue sur l'exigence source.

NOTE – Atteindre une exigence signifie qu'elle soit automatiquement sélectionnée et ses propriétés affichées.

#### **5.1.1.2.3. Synchronisation de tous les types**

La gestion multi-documents doit permettre de synchroniser tout type d'exigence.

NOTE – La responsabilité de la cohérence entre des exigences synchronisées en revient à l'administrateur ou à l'utilisateur, étant donné qu'aucune vérification n'est effectuée.

#### **5.1.1.3. Gestion de la génération**

##### **5.1.1.3.1. Génération d'un document regroupant l'ensemble des exigences synchronisées**

La gestion multi-documents doit permettre de générer un nouveau tableau dans Microsoft Word regroupant les exigences synchronisées. Les différentes colonnes de ce tableau sont les suivantes : ID de l'exigence – Occurrence – 'No paraph' – Titre de l'exigence.

NOTE – Occurrence est le nom du regroupement qui contient l'exigence synchronisée en question. 'No paraph' est le numéro de paragraphe de l'exigence dans le regroupement correspondant. L'accès à la génération de ce tableau est intégré dans la liste de choix de document à générer présente dans GenSpec.

##### **5.1.1.3.2. Omission des sections Autres fonctions dans l'arbre global**

La gestion multi-documents doit omettre toutes les sections Autres fonctions de l'arbre global lors de la génération de ce dernier.

NOTE – Cette omission est souhaitable car la génération de la section Autres fonctions n'apporte aucune information pertinente, l'information étant présente dans l'arbre global.

#### **5.1.1.4. Gestion des fenêtres de regroupements**

NOTE – Dans les exigences ci-dessous, nous appellerons une fenêtre, une fenêtre de regroupements d'arbres hiérarchiques.

##### **5.1.1.4.1. Affichage d'autant de fenêtres que l'on souhaite**

La gestion multi-documents doit permettre d'afficher autant de fenêtres de regroupement que l'on souhaite.

##### **5.1.1.4.2. Affichage des fenêtres**

La gestion multi-documents doit permettre d'afficher les différents regroupements d'un projet dans les fenêtres. Les fenêtres sont ordonnées selon les conditions suivantes :

- a) Aucun regroupement créé. L'arbre hiérarchique est créé dans la fenêtre sans appartenir à un onglet ;
- b) Au moins un regroupement créé. Chaque arbre hiérarchique est représenté dans un onglet différent ;
- c) Une seule fenêtre affichée. La fenêtre est nommée 'Arbre hiérarchique' ou 'Arbres hiérarchiques' en fonction du nombre d'onglets présents dans la fenêtre ;
- d) Plusieurs fenêtres affichées. Le nom des fenêtres doit être différent. Leur nom est différent de par l'utilisation d'un suffixe. NOTE – Par exemple, si deux fenêtres sont affichées, elles s'appellent respectivement 'Arbres hiérarchiques 1' et 'Arbres hiérarchiques 2' (dans le cas où plusieurs onglets sont créés)

#### **5.1.1.5. Gestion des exigences entre les regroupements**

NOTE – Un déplacement interne est le déplacement d'une exigence à l'intérieur de son regroupement. Un déplacement externe est le déplacement d'une exigence à l'extérieur de son regroupement.

##### **5.1.1.5.1. Numéro hiérarchique des exigences d'un regroupement**

Les numéros hiérarchiques des exigences d'un regroupement doivent correspondre aux numéros hiérarchiques de ces mêmes exigences dans l'arbre global.

##### **5.1.1.5.2. Création de la section Autres fonctions**

La gestion multi-documents doit permettre de convertir les liaisons externes en liaisons internes à l'intérieur d'un regroupement :

- a) Création automatique de la section Autres fonctions. Afin de convertir une liaison externe en liaison interne, il faut créer automatiquement une section Autres fonctions qui permet d'insérer l'exigence faisant l'objet d'une liaison externe, de sorte à rendre cette exigence interne. L'organisation de la section Autres fonctions est la suivante : Niveau Autres fonctions. Niveau Intrants, Extrants. Niveau des exigences intrants et extrants. La provenance de l'intrant ou de l'extrant est notée entre parenthèses comme suffixe au titre de l'exigence ; NOTE – La provenance est le titre de la racine dans l'arbre global d'où provient l'exigence faisant l'objet d'une liaison externe.
- b) Interdiction de la création de la section Autres fonctions si elle est déjà présente. Si la section Autres fonction est présente dans un regroupement, il est interdit d'en créer une autre. Les ajouts manuels d'exigences relatifs à des autres fonctions doivent être insérés dans cette section existante.

NOTE – Une liaison interne est une liaison entre des exigences qui se trouvent dans le même regroupement. Une liaison externe est une liaison entre des exigences qui se trouvent dans des regroupements distincts.



## **Annexe G**

# **Conception : Glossaire**

La dernière modification du contenu de ce document date du 10 janvier 2007.



*Document de conception*

**GenSpec**

**Glossaire**

**CON10004**

**OCTOBRE 2006**

## **1. INTRODUCTION**

### **1.1. Objet**

Ce document comporte un ensemble de diagrammes et de schémas correspondant à la conception physique de la fonctionnalité de *glossaire* à insérer au sein de *GenSpec*.

Ce document s'adresse à toutes personnes concernées par le développement de la fonctionnalité de *glossaire* pour *GenSpec*.

### **1.2. Portée**

Au sein de ce document, on peut retrouver différents diagrammes et schémas permettant de fournir de l'information selon différents points de vue des éléments à implémenter au sein de *GenSpec*. Notons que les schémas suivants comportent également des éléments associés à l'autre projet *multi-document* - qui sera développé en parallèle au *glossaire* pour *GenSpec* - et ce, afin d'éviter tout problèmes de cohésion ou duplication inutile de différents aspects liés aux projets.

### **1.3. Définitions, acronymes et abréviations**

Aucuns objets.

### **1.4. Références**

[1] GenSpec Glossaire – Étude – ETU10004, Octobre 2006.

[2] GenSpec Glossaire – Spécifications des exigences – SES10004, 2006.

## 1.5. Vue d'ensemble

Ce document se compose des 7 chapitres :

1. INTRODUCTION : Ce chapitre fournit une vue d'ensemble de toute la spécification. Il en décrit :
  - l'objet ;
  - la portée ;
  - les définitions, acronymes et abréviations ;
  - les références ;
  - une vue d'ensemble.
2. DESCRIPTION GÉNÉRALE : Ce chapitre fournit une description générale des différents chapitres présents au sein du document.
3. DIAGRAMME DE COMPOSANTS : Ce chapitre fournit un diagramme de composants. Il en décrit :
  - les composants importants du pour le développement du glossaire.
4. DIAGRAMMES DE SEQUENCES : Ce chapitre fournit un ensemble de diagrammes de composants. Il en décrit :
  - différents cas d'interactions entre les différents composants.
5. DIAGRAMMES DE CLASSES : Ce chapitre fournit un diagramme de classes. Il en décrit :
  - les différentes classes et méthodes à développer.
6. MACHINE À ÉTATS : Ce chapitre fournit un diagramme d'états vis-à-vis du glossaire. Il en décrit :
  - les différents états - ainsi que les transitions pour passer entre ces états - de la définition d'un terme au sein du glossaire.
7. SHÉMA ENTITÉ-ASSOCIATION : Ce chapitre fournit une vue de la base de données. Il en décrit :
  - deux schémas réduits à la fonctionnalité de *glossaire* (conceptuels et relationnels);
  - deux schémas globaux (conceptuels et relationnels).

## **2. DESCRIPTION GÉNÉRALE**

Ce chapitre fournit une description générale des différents éléments contenus au sein de ce document.

### **2.1. Diagramme de composants**

Le diagramme de composants reprend l'ensemble des composants distincts au sein du système et ce, pour les deux projets (*Glossaire* et *multi-documents*) à développer.

### **2.2. Diagrammes de séquence**

Ce chapitre reprend l'ensemble des diagrammes de séquence associé à la nouvelle fonctionnalité de *glossaire* pour *GenSpec*. Ces diagrammes fournissent différentes vues concernant les interactions entre les composants selon certains cas.

### **2.3. Diagramme de classe**

Le diagramme de classes, au sein de chapitre, reprend l'ensemble des méthodes à établir lors du développement de la nouvelle fonctionnalité de *glossaire* ainsi que la gestion *multi-documents* au sein de *GenSpec*.

### **2.4. Machine à états**

Cette partie comprend un diagramme d'états reprenant les différents états dans lequel une définition d'un terme au sein d'un glossaire peut se retrouver. On peut y retrouver également les causes, les origines de ces changements d'états.

### **2.5. Schéma Entité-association**

Ce chapitre comporte deux types de schémas distincts. Tout d'abord, on retrouve deux schémas (conceptuel et relationnel) représentant la base de données propre au glossaire et ce, afin de fournir une vue générale des changements limités au glossaire à réaliser. Ensuite, deux autres schémas (conceptuel et relationnel) sont fournis avec l'ensemble des autres classes de *GenSpec* afin d'avoir une vue complète du système.

### 3. DIAGRAMME DE COMPOSANTS

Le diagramme de composants reprend l'ensemble des composants vis-à-vis des projets à développer. Ce diagramme est subdivisé en trois parties (*packages*) distincts représentant la distribution des composants selon les deux projets (multi-documents et glossaire) ainsi qu'une partie reprenant les éléments en commun.

### 3.1. Diagramme

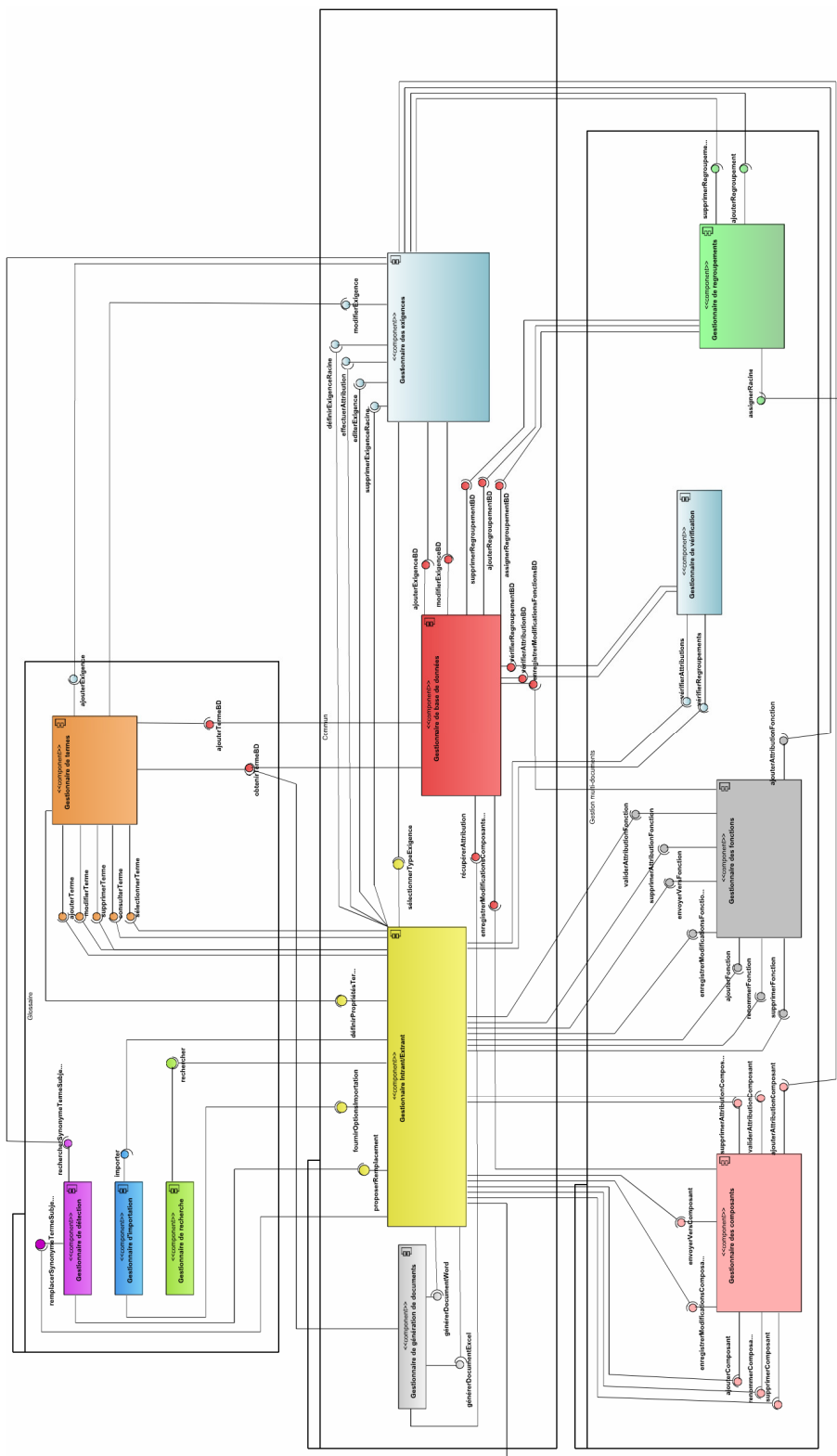


Figure 1 - Diagramme de composants

### 4. DIAGRAMMES DE SEQUENCE

Les diagrammes de séquence présents au sein de cette section permettent de définir les interactions entre les différents composants présents au sein du système selon certaines situations.

#### 4.1. Ajouter un terme au glossaire

Ce diagramme de séquence reprend l'ensemble des opérations à effectuer lors de l'ajout d'un terme au sein du glossaire. Notons que l'argument *informations* au sein de *ajouterTerme* et *ajouterTermBD* correspond à l'ensemble des informations qui sont associés à un terme.

*Pré-condition* : Le terme et les informations associées aux termes doivent être définies par l'utilisateur.

*Post-condition* : Le terme ainsi que l'ensemble de ses informations associées ont été enregistrées au sein de la base de données si le terme que l'on souhaite définir n'est pas un synonyme compris au sein d'une liste de synonymes d'un autre terme défini dans le glossaire.

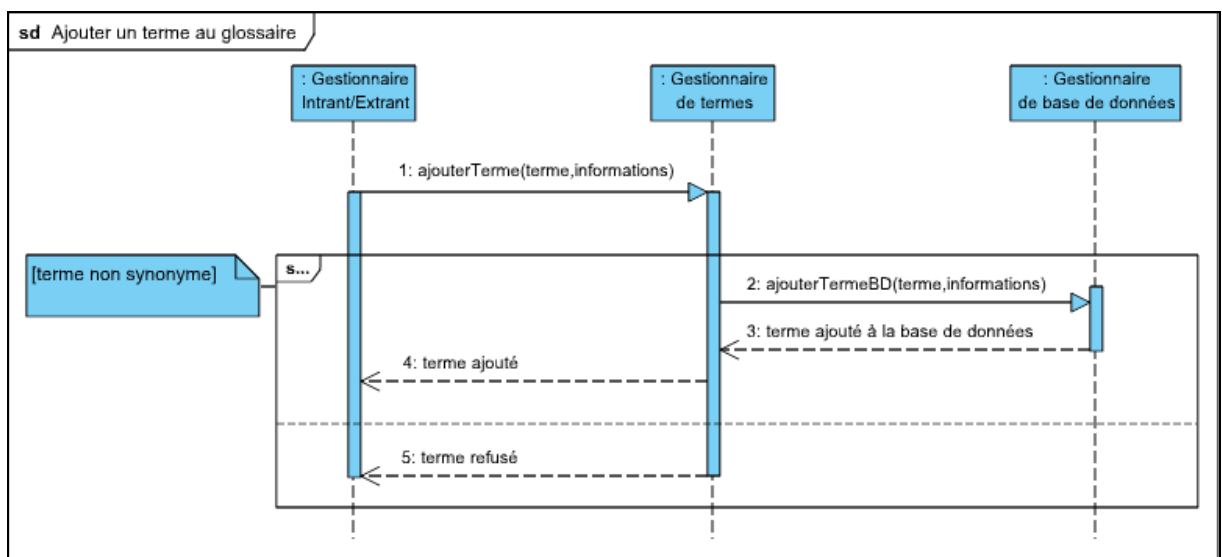


Figure 2 - Ajouter un terme au glossaire

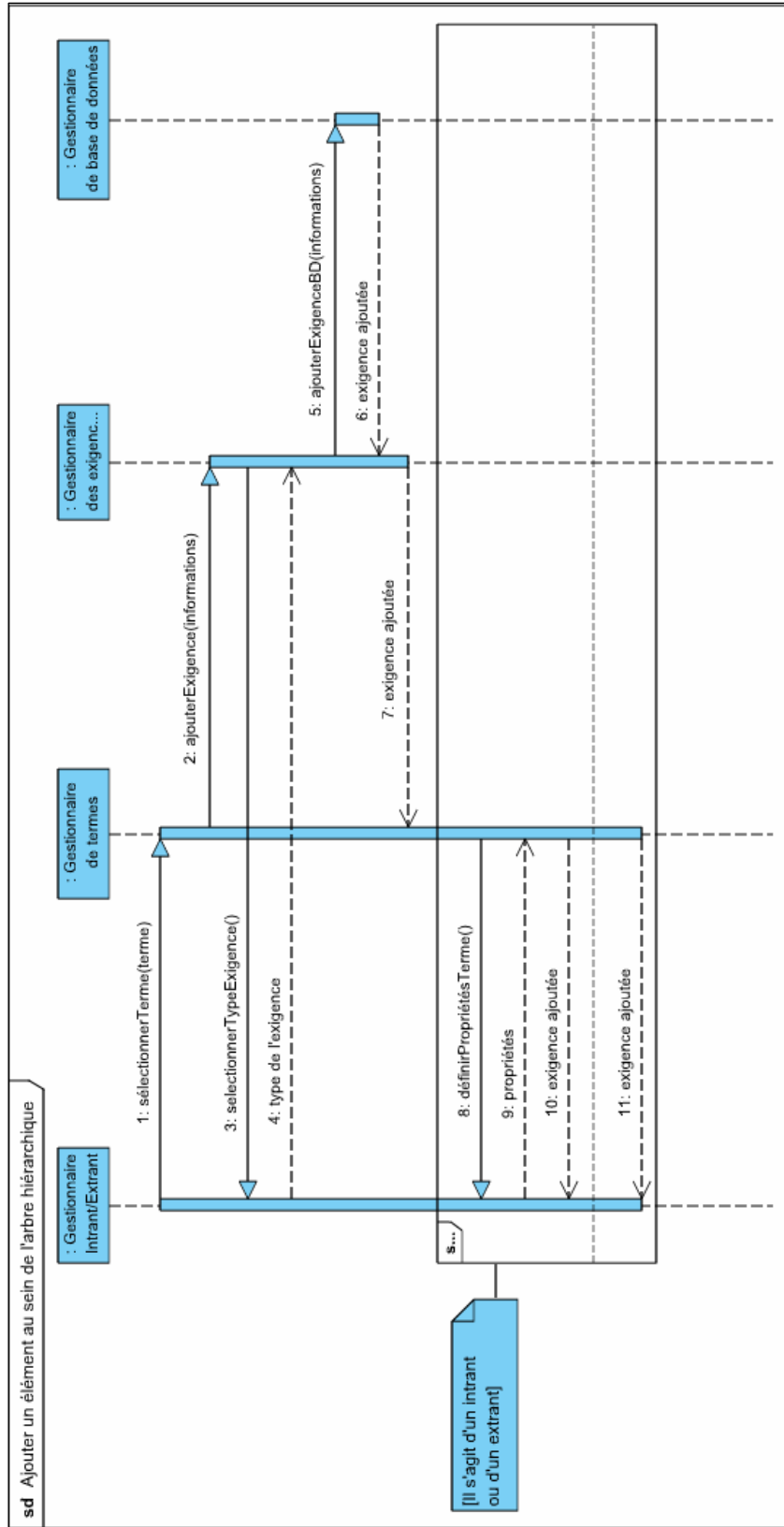
#### **4.2. Ajouter un élément au sein de l'arbre hiérarchique**

L'insertion d'un nouvel élément au sein de l'arbre hiérarchique à partir du glossaire va créer une nouvelle exigence au sein de cet arbre. Le terme du glossaire, outre le fait qu'il soit défini au sein du glossaire, va devenir également une exigence. De plus, dans le cas de la définition d'un intrant ou un extrant, la propriété intrant et/ou extrant pourra être associé au terme dans le glossaire.

*Pré-condition* : Le terme à insérer, comme exigence, comme nouvelle exigence (intrans/extrant) au sein de l'arbre hiérarchique est défini au sein du glossaire.

*Post-condition* : Le terme à été inséré au sein de l'arbre hiérarchique.





### 4.3. Consulter les informations d'un terme

Consulter l'ensemble des informations d'un terme revient à obtenir les informations comprises au sein de la base de données concernant ce terme.

*Pré-condition* : Le terme à consulter est défini au sein du glossaire.

*Post-condition* : Les informations sur le terme ont été obtenues.

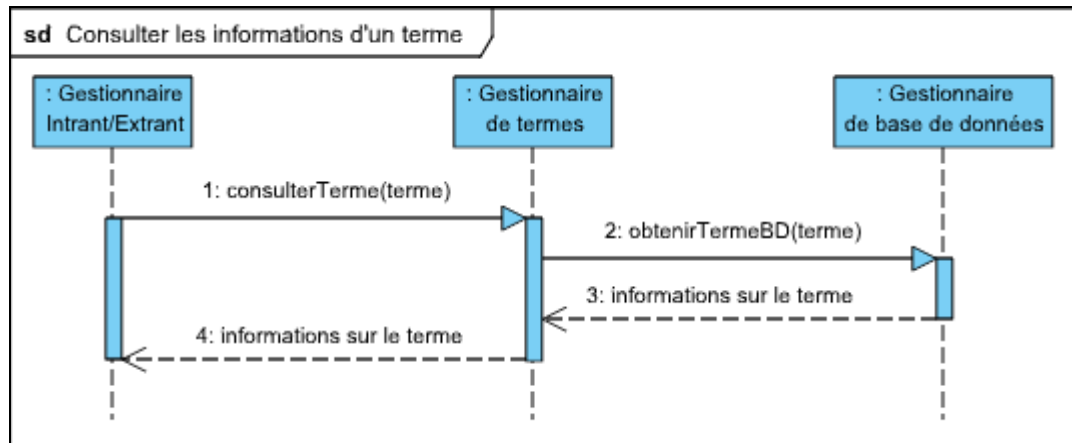


Figure 4 - Consulter les informations d'un terme

#### 4.4. Détection de synonymes ou termes subjectifs

Recherche un synonyme ou un terme subjectif au sein des exigences revient à vérifier, lors de la rédaction des exigences, si l'utilisateur emploie ou non un synonyme ou un terme subjectif. L'utilisateur se voit alors proposer le terme correspondant au synonyme ou la suppression du terme subjectif.

*Pré-condition* : Une exigence est en cours de rédaction et un synonyme **ou** un terme subjectif est employé.

*Post-condition* : Le synonyme a été remplacé par son terme associé si le remplacement est accepté

**ou** le terme subjectif a été supprimé si le remplacement est accepté.

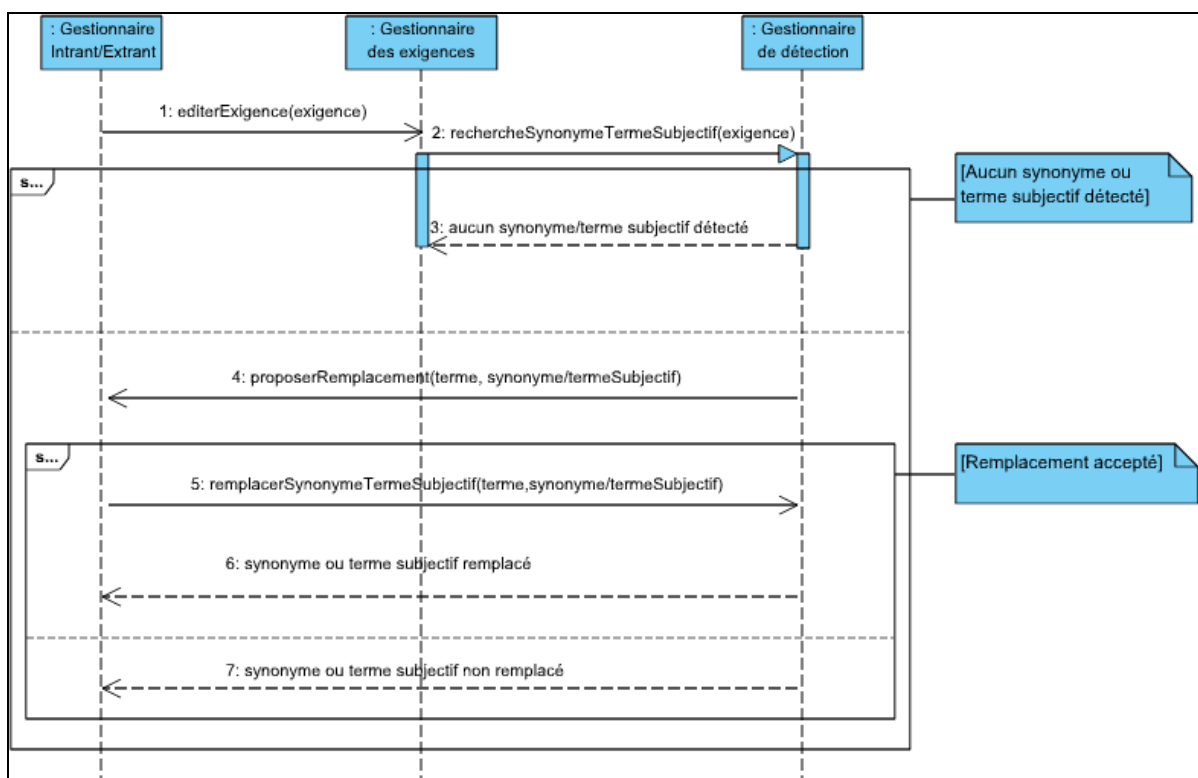


Figure 5 - Détection de synonymes ou termes subjectifs

#### 4.5. Générer le document associé au glossaire

Générer le document associé au glossaire revient à produire un document *Microsoft Word* contenant la définition, le but pour le système et le but pour l'acteur pour chaque terme défini au sein du glossaire.

*Pré-condition* : Aucune.

*Post-condition* : Le document reprenant l'ensemble des informations sur le terme a été généré.

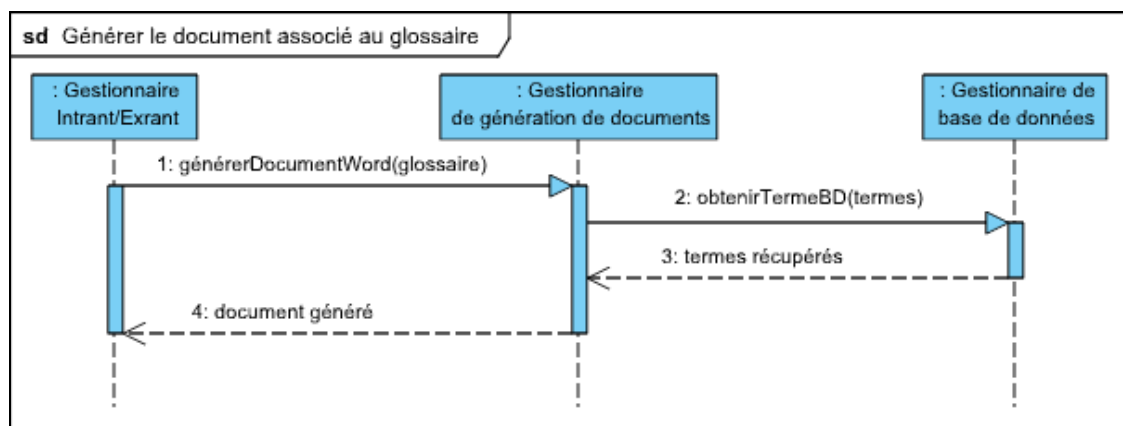


Figure 6 - Générer le document associé au glossaire

#### 4.6. Importer un glossaire

Importer un glossaire revient à insérer au sein d'un projet, un ensemble de termes définis au sein d'un projet précédent.

*Pré-condition* : Un glossaire venant d'un autre projet doit exister - et il peut être vide - .

*Post-condition* : Le glossaire venant d'un projet précédent a été insérer au sein du projet actuel.

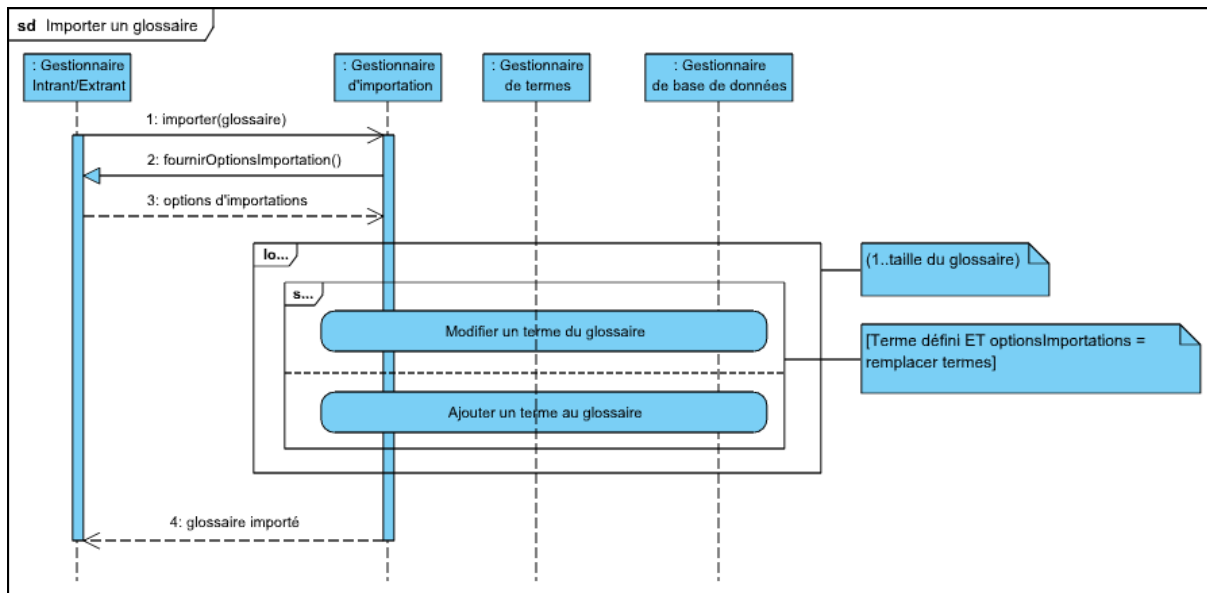


Figure 7 - Importer un glossaire

#### 4.7. Modifier un terme du glossaire

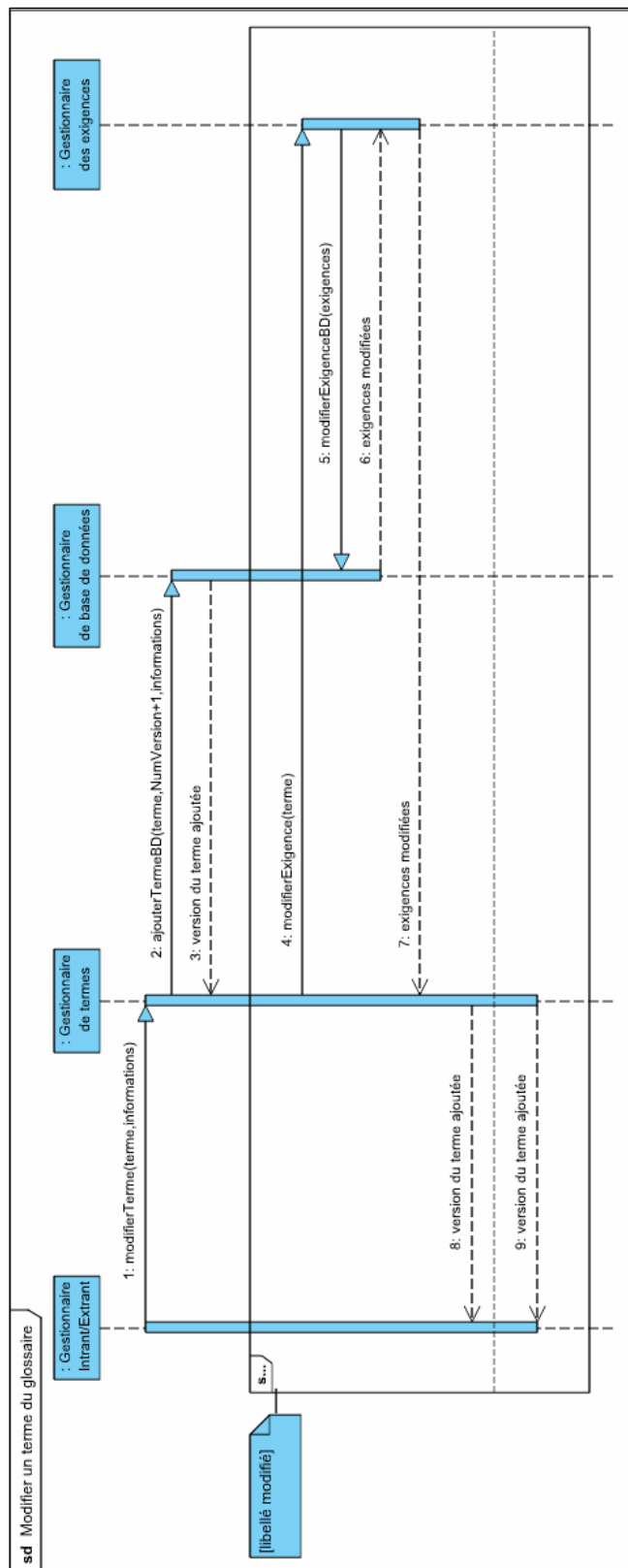
Modifier un terme présent au sein du glossaire revient à effectuer un quelconque changement sur une quelconque information associée à un terme au sein du glossaire. Cependant, il est important de noter que la modification n'entraîne pas un changement des informations associées au terme en tant que tel, elle ajoute une nouvelle version des informations du terme.

*Pré-condition* : Le terme à modifier doit être déjà défini au sein du glossaire

**et** des modifications ont été réalisées sur ses informations.

*Post-condition* : Les modifications effectuées sur les informations d'un terme ont été enregistrées au

sein de la base de données.



#### 4.8. Rechercher un terme

Rechercher un terme, cela revient à fournir un ensemble de caractéristiques ou informations afin de retrouver un terme défini au sein du glossaire.

*Pré-condition* : Aucune.

*Post-condition* : L'utilisateur dispose d'une liste de termes (qui peut être vide) correspondant aux options de recherches définies.

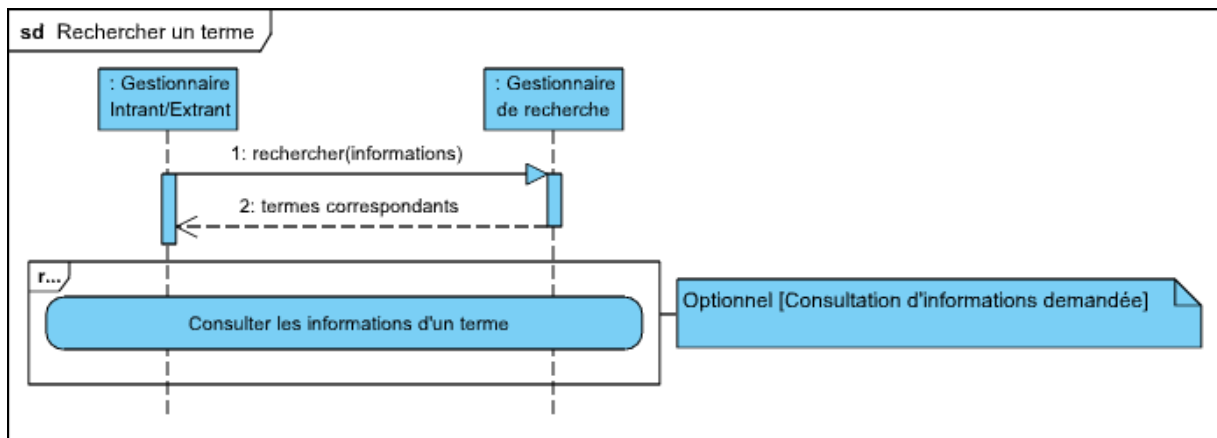


Figure 9 - Recherche un terme

#### 4.9. Supprimer un terme présent au sein du glossaire

Supprimer un terme présent au sein du glossaire revient à modifier le statut du terme au sein de la base de données. En effet, un terme n'est jamais réellement supprimé du glossaire. Celui-ci devient inaccessible pour les utilisateurs mais il est toujours présent au sein de la base de données.

*Pré-condition* : Le terme à supprimer est défini au sein du glossaire.

*Post-condition* : Le terme à supprimer n'est plus visible au sein du glossaire de *GenSpec*

**et** le terme est marqué supprimé au sein de la base de données.

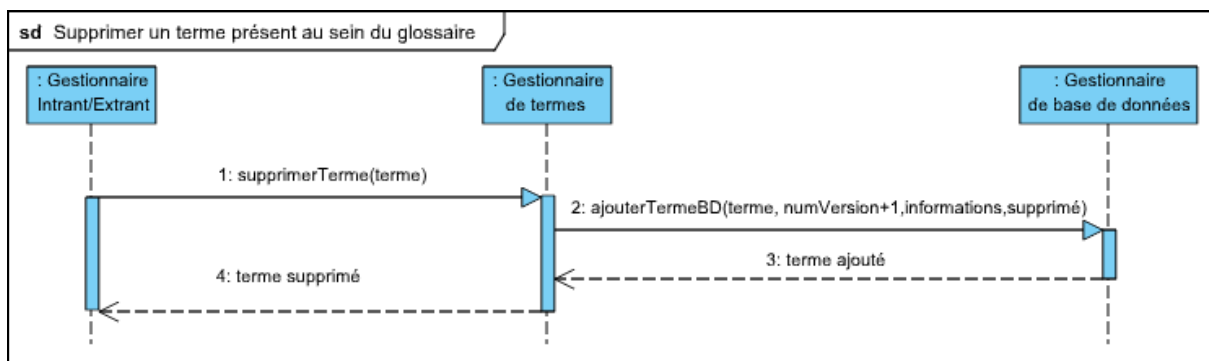


Figure 10 - Supprimer un terme présent au sein du glossaire

### 5. DIAGRAMME DE CLASSES

Le diagramme de classes comporte les différentes classes et les méthodes à réaliser. On retrouve ici, tout comme pour le diagramme de composants, des éléments concernant le projet *multi-documents* ainsi que le projet *glossaire*. Une distinction a été établie entre les éléments qui sont associés au *glossaire*, au *multi-document* et ceux qui sont liés aux deux projets.

### 5.1. Diagramme

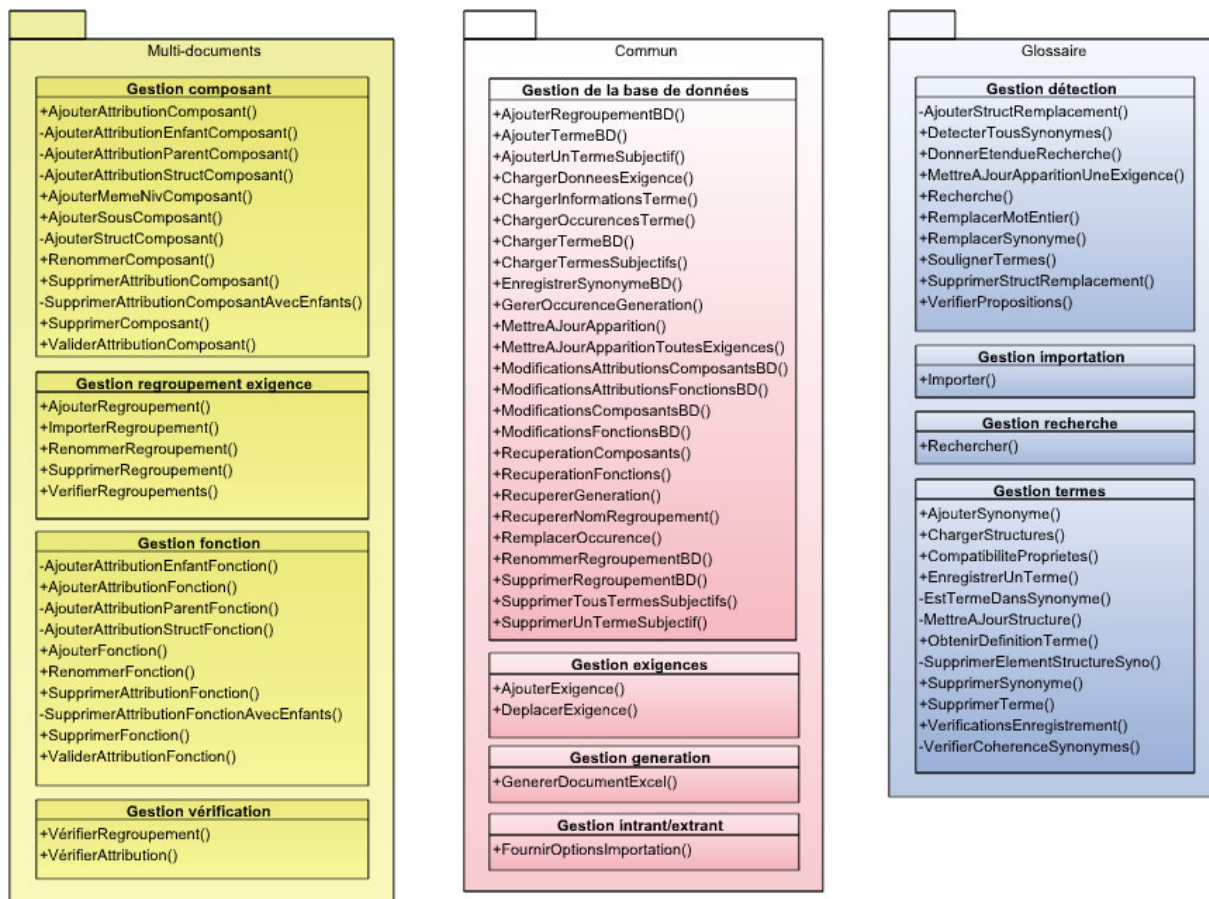


Figure 11 - Diagramme de classes



## 6. DIAGRAMME D'ÉTATS

Ce chapitre reprend l'unique diagramme d'états associé à la fonctionnalité de *glossaire*.

### 6.1. États de l'ensemble des informations d'un terme au sein du glossaire

Ce diagramme d'états représente les différents états possibles d'une définition au sein du glossaire ainsi que les différentes transitions possibles entre ces états.

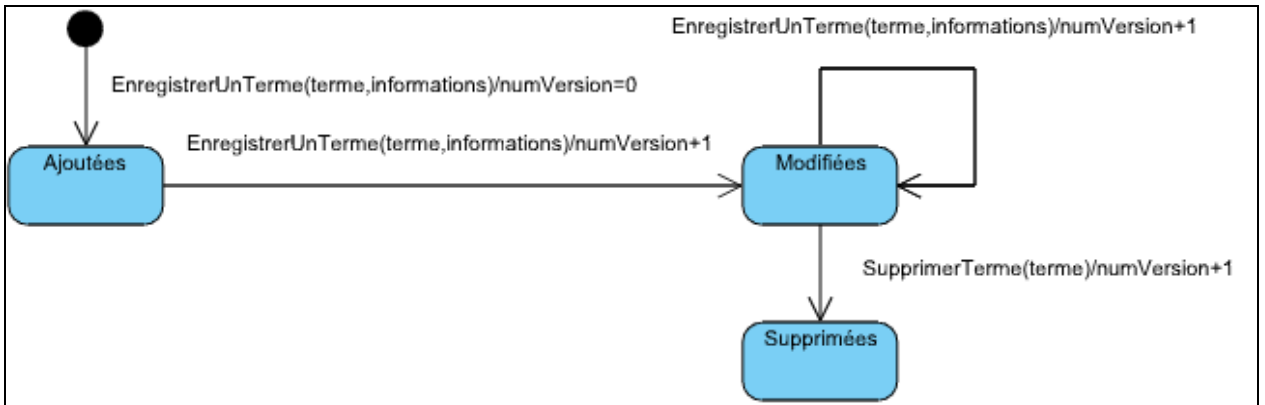


Figure 12 - États de la définition d'un terme

## 7. SCHÉMA DE BASE DE DONNÉES

Au sein de ce chapitre, on retrouve 2 types de schémas représentant la future base de données : réduit et global.

Les schémas réduits (conceptuel et relationnel) à l'aspect *glossaire* fournissent une vue d'ensemble limitée uniquement à la fonctionnalité de glossaire. Les schémas globaux, quant à eux, fournissent un point de vue général sur l'ensemble des informations que compose un projet et ce, y compris le projet *multi-document* qui sera en cours de développement de façon parallèle avec le *glossaire*.

### 7.1. Schémas réduits au *glossaire*

Les schémas ci-dessous représentent la structure de la base de données associée au *glossaire* de *GenSpec*. Notons que le schéma s'avère simplifié. En effet, les tables EXIGENCE et UTILISATEUR sont déjà présents au sein de la base de données actuelle (au sein de *GenSpec*) et ont été représentée ici de manière très simplifiée afin de ne pas alourdir les schémas inutilement.

#### 7.1.1. Schéma conceptuel

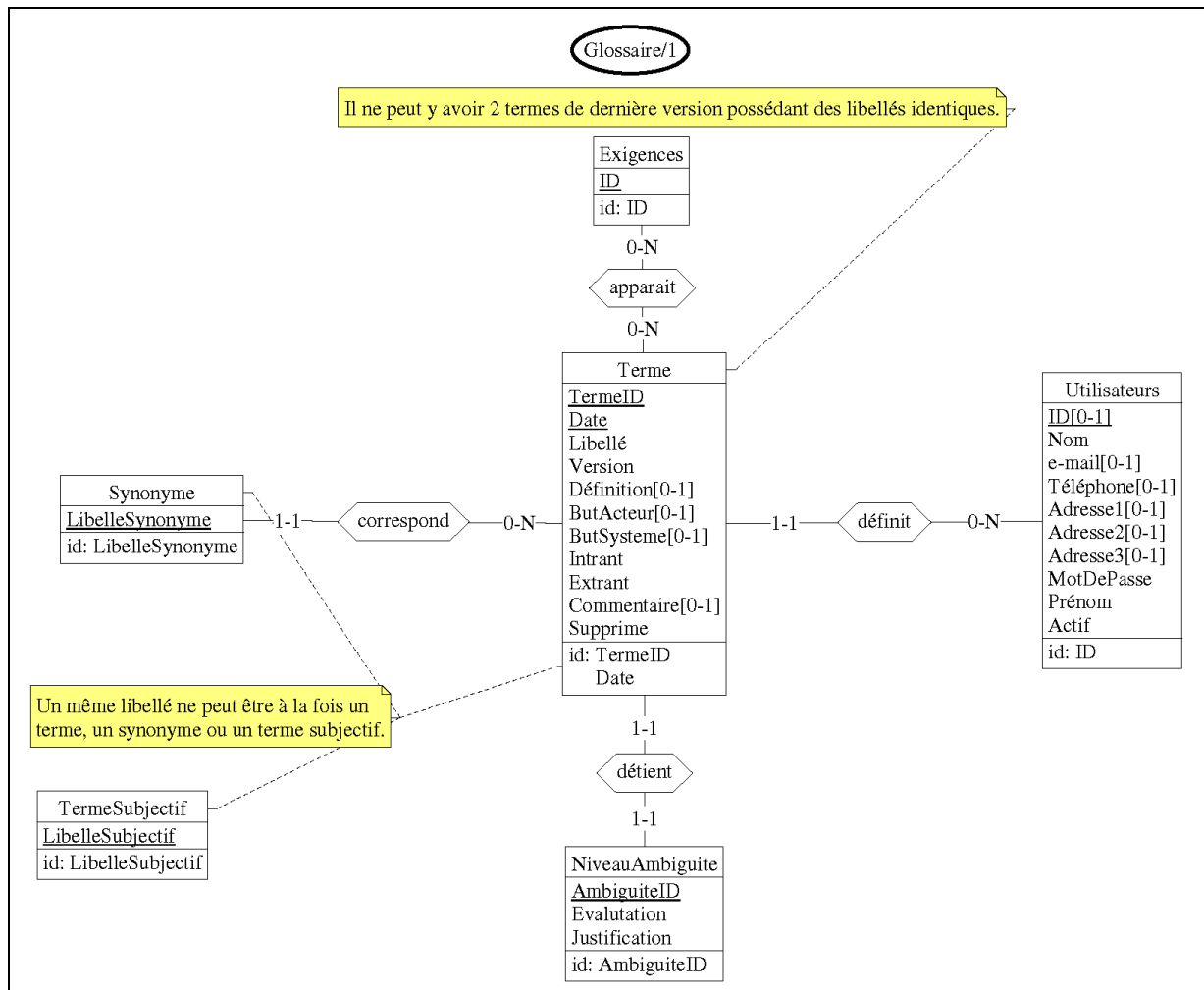


Figure 13 - Schéma conceptuel réduit au *glossaire*

7.1.2. Schéma relationnel

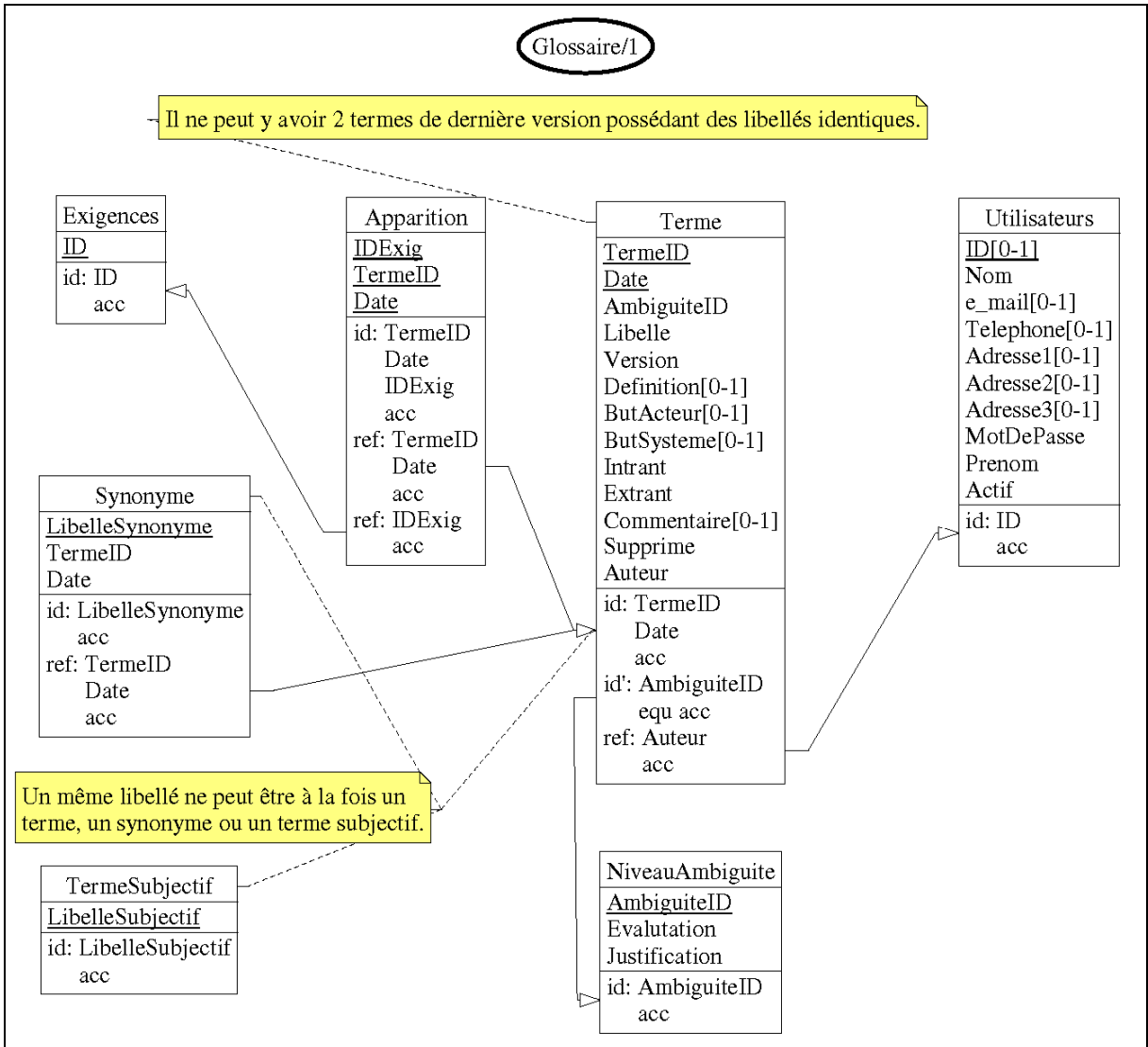


Figure 14 - Schéma relationnel réduit au *glossaire*

### 7.1.3. Sémantique

La section suivante définit la sémantique des différents champs associés au sein de cette base de données. Les tables EXIGENCE et UTILISATEUR ne seront pas spécifiées ici car celles-ci sont définies au sein de la base de données (excepté le champ «Actif» de la table UTILISATEUR qui en est un nouvel attribut).

#### 7.1.3.1. Tables

##### 7.1.3.1.1. Table TERME

Un terme est un élément qui peut être utilisé au sein d'une exigence, qui possède une version et un niveau d'ambiguïté. Il peut également posséder une liste de synonymes.

- **TermeID** : TermeID correspond au numéro identificateur d'un terme.
- **Date** : Date correspond à la date de création, de modification ou de suppression de l'ensemble des informations associées à un terme.
- **Libellé** : Le libellé correspond à la syntaxe d'un terme.
- **Version** : La version d'un terme correspond à son numéro de version suite aux différentes modifications appliquées au minimum à une seule des informations d'un terme.
- **Définition** : La définition du terme correspond à l'explication du concept, du sens du mot selon l'utilisateur qui l'a défini. La définition correspond à la sémantique du terme.
- **ButActeur** : Le but pour l'acteur correspond au but du terme pour tout acteur lié au projet en cours de spécification. L'acteur peut être un humain ou tout autre système.
- **ButSystème** : Le but pour le système décrit quelle est l'utilité du terme employé pour le système associé au projet en cours de spécification.
- **Intrant** : Intrant défini sur le terme possède la propriété intrant (true) ou non (false).
- **Extrant** : Extrant défini sur le terme possède la propriété extrant (true) ou non (false).
- **Commentaire** : Commentaire comprend l'ensemble des commentaires que l'utilisateur qui a fourni la dernière version de la définition juge important de noter concernant le terme.
- **Supprimé** : Supprimé définit si un terme est supprimé ou non du glossaire (un terme n'étant jamais réellement supprimé de la base de données). Si sa valeur est vrai (true) alors il est supprimé et si elle est à faux (false) alors le terme n'est pas supprimé.

##### 7.1.3.1.2. Table TERME SUBJECTIF

Un terme subjectif est un terme pour lequel il est préférable qu'il ne soit pas utilisé au sein d'une exigence. A lui seul, le terme peut entraîner de l'ambiguïté de part son caractère subjectif. Les termes subjectifs sont fournis par l'administrateur.

- **LibelleSubjectif** : Le libellé subjectif correspond à la syntaxe du terme subjectif.

##### 7.1.3.1.3. Table SYNONYME

Un synonyme correspond à un terme de sens correspondant à un terme défini au sein du glossaire. Un synonyme ne peut être associé à plus de un terme.

- **LibelleSynonyme** : L'énoncé d'un synonyme correspond à la syntaxe du synonyme correspondant. La sémantique du synonyme est définie de manière indirecte par la sémantique du terme auquel il correspond.

#### 7.1.3.1.4. Table NIVEAU AMBIGUÏTE

Le niveau d'ambiguïté correspond à une estimation, de la part d'un utilisateur, du niveau d'ambiguïté de la définition d'un terme.

- **AmbiguïteID** : Le numéro correspond à la version de la définition de l'ambiguïté d'un terme. Le numéro est identifiant permettant de différencier les versions du niveau ambiguïté
- **Evaluation** : L'évaluation correspond à l'estimation de part l'utilisateur du niveau ambiguïté de la définition d'un terme.
- **Justification** : La justification correspond à l'explication fournie par l'utilisateur d'un tel niveau ambiguïté associé à la définition d'un terme.

#### 7.1.3.1.5. Table UTILISATEUR

Un utilisateur correspond à un ensemble d'informations concernant tout utilisateur de GenSpec.

- **Actif** : Actif détermine si l'utilisateur est actif ou non dans le projet *GenSpec*. Un utilisateur est inactif s'il est l'auteur d'un terme importé au sein du glossaire dans le projet *GenSpec* en cours. Il ne peut pas se connecter au projet *GenSpec*. Un utilisateur est dit actif dans tous les autres cas.

### 7.1.3.2. Liens

#### 7.1.3.2.1. Apparaît

Un terme peut apparaître dans plusieurs exigences (ou pas). Une exigence peut également posséder plusieurs termes au sein de son énoncé ou de son libellé.

#### 7.1.3.2.2. Correspond

Un synonyme correspond à un et un seul terme. Un terme peut posséder plusieurs synonymes mais un synonyme ne peut être lié à plus de 1 terme.

#### 7.1.3.2.3. Définit

Une version contient un et un seul utilisateur. Cela correspond à l'utilisateur qui a fourni la version de la définition du terme.

#### 7.1.3.2.4. Détient

Un terme détient un niveau ambiguïté Un niveau d'ambiguïté correspondant uniquement à un et un seul terme.

## **7.2. Schémas globaux**

Cette partie comporte les deux schémas (conceptuel et relationnel) représentant la base de données à établir. Au sein de ces schémas globaux, on retrouve l'ensemble des tables et relations de la base de données de *GenSpec*. Notons qu'aucune modification n'a été apportée aux tables et relations existantes. Les projets en développement (glossaire et multi-documents) se veulent séparer de toute restructuration importante de la base de données ainsi que le support de ces modifications majeures au sein de l'entière de *GenSpec*. Les nouvelles tables et relations insérées par les nouveaux projets ne réalisent aucuns changements dans la structure des tables et relations déjà présentes (excepté le champ "Actif" au sein de la table UTILISATEUR).

### **7.2.1. Schéma conceptuel**

On retrouve ici le schéma conceptuel de base de données globale de *GenSpec*. Ce schéma comprend les aspects liés à la gestion multi-documents (en bleu) et au glossaire (en vert).



### 7.2.2. Schéma relationnel

Cette partie comporte la version relationnelle du schéma de la base de données globale de *GenSpec*. Tout comme pour le schéma conceptuel, cette base de données comporte les aspects liés à la gestion multi-document (en bleu) et au glossaire (en vert).

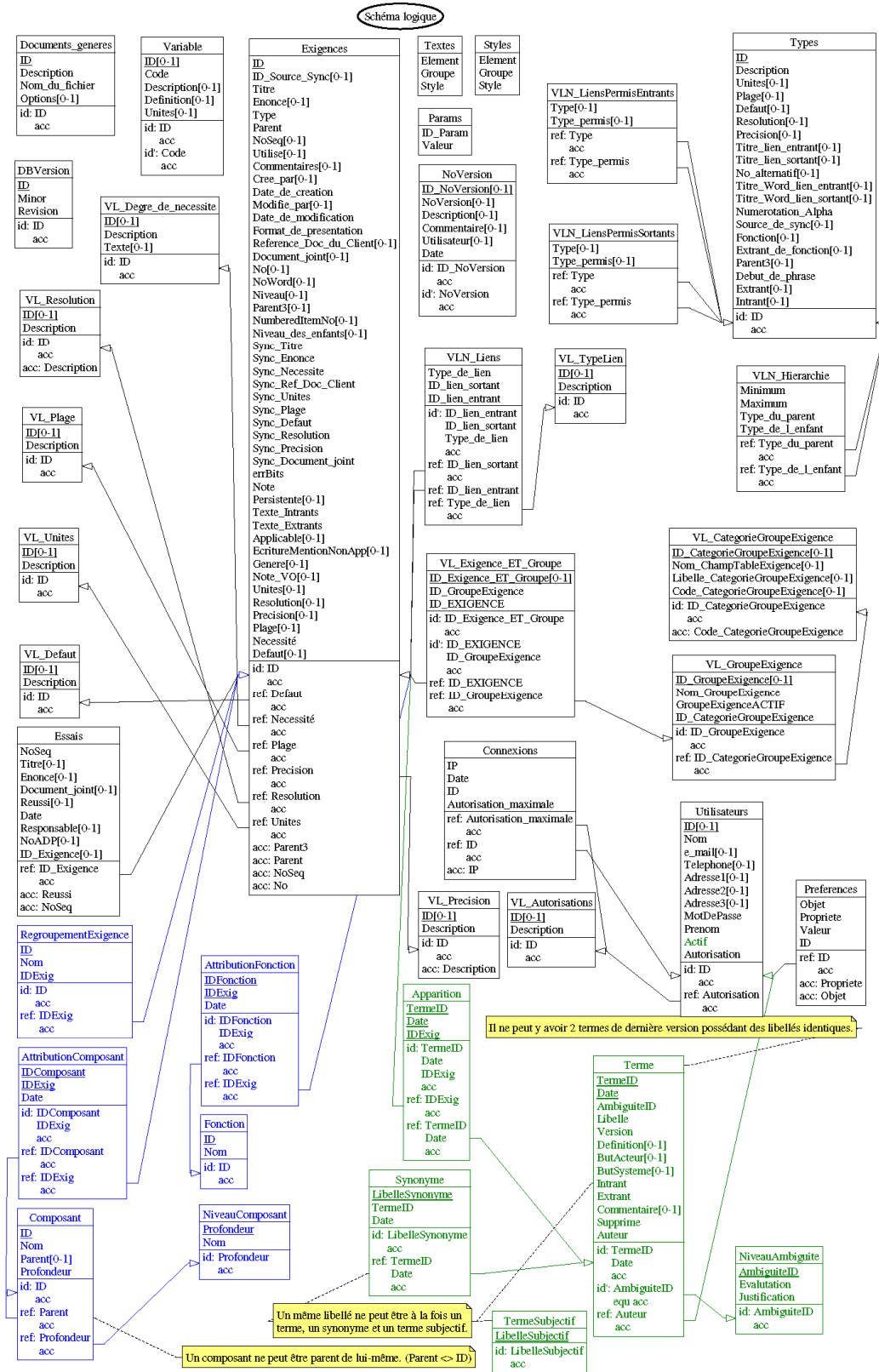




Figure 16 - Schéma relationnel global

### 7.2.3. Sémantique

La sémantique de l'ensemble des champs associés à la globalité de la base de données peut être retrouvée au sein de la base de données ou au sein de ces schémas (projet *DB-main*) directement.

La sémantique de ces différents champs n'a pas été reportée ici dans un esprit de concision.

## 8. CONCLUSION

Le document ci-présent offre donc une conception pour établir un *glossaire* au sein de *GenSpec*.

Ces différents schémas permettent de fournir différents points de vue concernant le développement à effectuer.

Ce document constitue à l'aboutissement de la définition d'un besoin de la part des utilisateurs de *GenSpec*. La prochaine étape est alors l'implémentation et de la mise en œuvre de l'ensemble des informations récoltées au sein des différents documents pour établir un *glossaire* correspondant à l'attente des utilisateurs de *GenSpec*.



## **Annexe H**

# **Conception : GMDE et AECF**

La dernière modification du contenu de ce document date du 10 janvier 2007.

*Document de conception*

# **Gestion multi-documents**

## **GenSpec**

**CON10003**

**NOVEMBRE 2006**

## 1. INTRODUCTION

### 1.1. Objet

Ce document s'adresse à toute personne concernée par l'ajout de la fonctionnalité de gestion multi-documents pour GenSpec.

Il décrit les différents schémas relatifs à la conception de la gestion multi-documents pour GenSpec.

### 1.2. Portée

Il expose les différents diagrammes jugés utiles à la conception des fonctionnalités relatives à la gestion multi-documents. Notons que ces diagrammes intègrent des informations relatives au glossaire pour GenSpec. En effet, ce projet étant développé en parallèle et utilisant des ressources communes, il paraissait opportun de les regrouper dans des diagrammes communs.

### 1.3. Définitions, acronymes et abréviations

Composant	: Un composant est un élément de base d'un ensemble plus complexe. Un système peut être décomposé en composants. Des exigences peuvent être attribuées à des composants, ce qu'on appelle des attributions. L'ensemble des exigences d'un composant étant satisfaites implique que le composant est lui-même satisfait.
Fonction	: Les exigences attribuées à des fonctions indiquent la fonction - pour le concepteur - que jouent ces exigences dans le système. Des exigences peuvent être attribuées à des fonctions, ce qu'on appelle des attributions.
Niveau	: Un niveau correspond à un ensemble de composants ayant la même profondeur. Notons que les composants peuvent être décomposés en plusieurs niveaux, alors que les fonctions ne possèdent qu'un seul niveau, "le niveau des fonctions".
Attribution	: Une attribution est la représentation d'une exigence dans un composant ou une fonction. On parle d'exigences dans les arbres hiérarchiques des exigences, et d'attributions dans les composants ou fonctions.
Regroupement	: Ensemble d'exigences regroupées dans un arbre hiérarchique d'exigences. Un regroupement ne possède qu'une seule exigence racine. Son but est de pouvoir travailler sur une partie uniquement des exigences.
Arbre global	: Arbre hiérarchique d'exigences pouvant posséder plusieurs racines et regroupant l'ensemble des exigences d'un projet.
Fenêtre de regroupement	: Fenêtre incluant l'arbre global, ainsi que les éventuels regroupements sous forme d'onglets.

### 1.4. Références

- [1] Nicolas Pirmez , "Gestion multi-documents pour GenSpec – Étude – ETU10003", Octobre 2006.
- [2] Olivier Pire, "Glossaire pour GenSpec – Étude – ETU10004", Octobre 2006.
- [3] Jean-Marc Hick, SQL-Script\_Generator pour DB-Main. (Importateur de schémas DB-Main vers Access).

## 1.5. Vue d'ensemble

Ce document se compose des 6 chapitres suivants :

1. INTRODUCTION : Ce chapitre fournit une vue d'ensemble de toute la spécification. Il en décrit :
  - l'objet ;
  - la portée ;
  - les définitions, acronymes et abréviations ;
  - les références ;
  - une vue d'ensemble.
2. DESCRIPTION GÉNÉRALE : Ce chapitre décrit brièvement les différents chapitres présents au sein de ce document de conception.
3. DIAGRAMME DE COMPOSANTS : Ce chapitre décrit le diagramme de composants ainsi que l'ensemble des composants présents dans ce dernier.
4. DIAGRAMME DE SÉQUENCE : Ce chapitre décrit les différents diagrammes de séquence jugés utiles. Chacun de ces diagrammes est accompagné par une brève description ainsi qu'une pré-condition et une post-condition.
5. DIAGRAMME DE CLASSES : Ce chapitre décrit le diagramme de classes. Ce dernier présente les classes et méthodes à développer.
6. SCHÉMAS ENTITÉS-ASSOCIATIONS : Ce chapitre fournit une description des différents schémas relatifs à la base de données. Il contient :
  - les schémas réduits présentant les fonctionnalités liées à la gestion multi-documents seuls ;
  - les schémas décrivant la base de données existante dans GenSpec.

## **2. DESCRIPTION GÉNÉRALE**

Cette section décrit les différents diagrammes présents dans ce document de conception pour la gestion multi-documents.

### **2.1. Diagramme de composants**

Il présente les différents composants du système et ce, pour la gestion multi-documents [1] et pour le glossaire [2]. Il représente la définition, la structure interne et les dépendances des types de composants.

### **2.2. Diagramme de séquence**

Il présente différents comportements, jugés utiles à modéliser, et ce, par des interactions entre les différents composants.

### **2.3. Diagramme de classes**

Il présente les différentes méthodes à réaliser lors du développement des nouvelles fonctionnalités. Il contient également les classes et méthodes relatives au glossaire pour GenSpec.

### **2.4. Schémas de base de données**

Il présente les différents types de schémas relatifs à la base de données. On y retrouve les schémas conceptuels et relationnels de la gestion multi-documents. D'une part, ce derniers sont représenter de manière indépendante vis-à-vis de la base de données existante, et d'autre part, ils sont intégrés dans les schémas de base de données existants et ce, afin de donner un aperçu des futurs schémas.



### **3. DIAGRAMME DE COMPOSANTS**

Ce diagramme présente les différents composants en relation avec les fonctionnalités de glossaire et de gestion multi-documents à développer pour GenSpec.

Notons que ce dernier est divisé en trois packages. Premièrement, celui destiné au glossaire, deuxièmement, le package relatif à la gestion multi-documents, et troisièmement le package commun qui réunit les composants et interfaces communes à ces deux projets.



### **3.2. Description des composants**

Dans cette section, nous décrivons brièvement le rôle de chacun des composants présents au sein du digramme de composants présenté ci-dessus.

#### **3.2.1. Gestionnaire de la base de données**

Il gère les accès avec la base de données de GenSpec. Il est responsable de l'enregistrement permanent des informations ainsi que de la consultation des données enregistrées.

#### **3.2.2. Gestionnaire des exigences**

Il gère les exigences présentent dans l'arbre hiérarchique des exigences. Il fournit les fonctionnalités existantes relatives à l'ajout et au déplacement d'exigences. Enfin, il permet l'importation des exigences dans les regroupements d'exigences.

#### **3.2.3. Gestionnaire de génération de documents**

Il gère tous ce qui est relatif à la génération de documents Microsoft Word ou Excel. De plus, il fournit la possibilité de paramétrer les options de génération.

#### **3.2.4. Gestionnaire de la vérification**

Il gère les vérifications liées aux regroupements et aux composants/fonctions.

#### **3.2.5. Gestionnaire des composants**

Il gère la hiérarchie des composants. Il permet de créer, renommer et supprimer des composants.

#### **3.2.6. Gestionnaire des fonctions**

Il gère la liste des fonctions. Il permet de créer, renommer et supprimer des fonctions.

#### **3.2.7. Gestionnaire des regroupements**

Il gère les regroupements d'exigences, notamment en donnant la possibilité d'ajouter, de renommer et de supprimer des regroupements.

#### **3.2.8. Gestionnaire intrant/extrant**

Il gère les interactions avec les utilisateurs. Il invoque les opérations sur les autres composants de par l'interface homme-machine.

#### **3.2.9. Gestionnaire de recherche**

Il gère toutes les recherches relatives aux termes définis dans le glossaire de GenSpec.

#### **3.2.10. Gestionnaire d'importation**

Il gère les importations de glossaires existants, ainsi que toute la gestion relative aux incohérences pouvant découler de ces importations (par exemple les conflits de termes existants).

#### **3.2.11. Gestionnaire des termes**

Il gère les opérations de création, de modification, de consultation et de suppression de terme.

### **3.2.12. Gestionnaire de détection**

Il gère les opérations de détection de synonymes et de termes subjectifs au sein des exigences.

#### 4. DIAGRAMMES DE SÉQUENCE

Cette section expose les différents diagrammes de séquence. Ces derniers sont accompagnés par une brève description, d'une pré-condition et d'une post-condition relative au séquençement.

Notons également que pour chacune des pré-conditions, les conditions suivantes sont satisfaites. L'application est disponible, et par conséquent il est possible de réaliser des attributions d'exigences et de créer des fonctions ou des composants. De plus, la base de données fonctionne correctement.

##### 4.1. Ajouter une attribution à un composant

Ajouter une attribution à un composant, montre que l'exigence en question fait l'objet d'une apparition dans un composant. L'ajout en base de données est réalisé une fois que l'enregistrement a eu lieu.

- *Pré-condition* : Une exigence est disponible dans un arbre hiérarchique des exigences, car c'est à partir de cette dernière qu'il est possible de réaliser l'attribution.
- *Post-condition* : L'exigence est attribuée dans le composant cible.

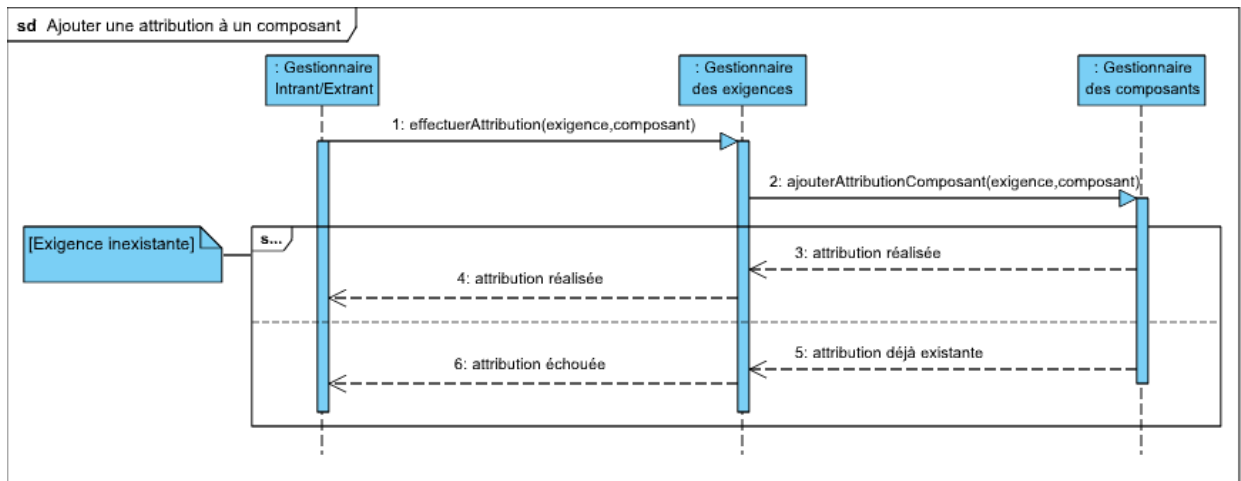


Figure 2 - Attribuer une exigence à un composant

#### 4.2. Ajouter une attribution à une fonction

Ajouter une attribution à une fonction, montre que l'exigence en question fait l'objet d'une apparition dans une fonction. L'ajout en base de données est réalisé une fois que l'enregistrement a eu lieu.

- *Pré-condition* : Une exigence est disponible dans un arbre hiérarchique des exigences, car c'est à partir de cette dernière qu'il est possible de réaliser l'attribution.
- *Post-condition* : L'exigence est attribuée dans la fonction cible.

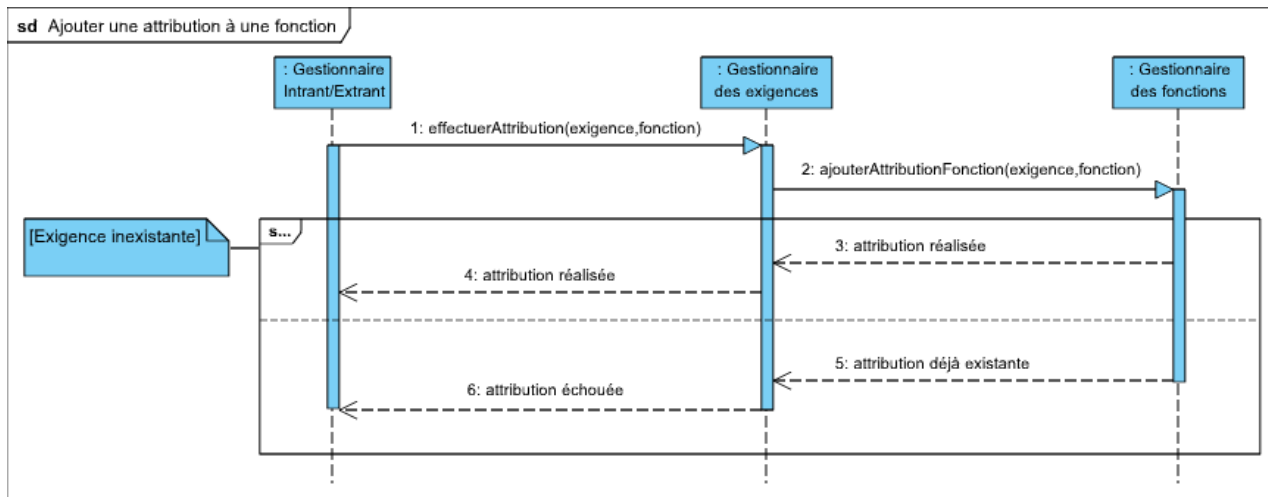


Figure 3 - Attribuer une exigence à une fonction

### 4.3. Ajouter un composant

Ajouter un composant dans la hiérarchie de composants. Un composant peut être ajouté sous un autre (ce dernier devient alors son parent), ou peut être ajouté au même niveau qu'un autre composant (ce dernier devient alors son frère) au sein de la hiérarchie. L'ajout en base de données est réalisé une fois que l'enregistrement a eu lieu.

- Pré-condition : Aucune.
- Post-condition : Le nouveau composant est créé si le nom est unique et est disposé à recevoir des attributions d'exigences.

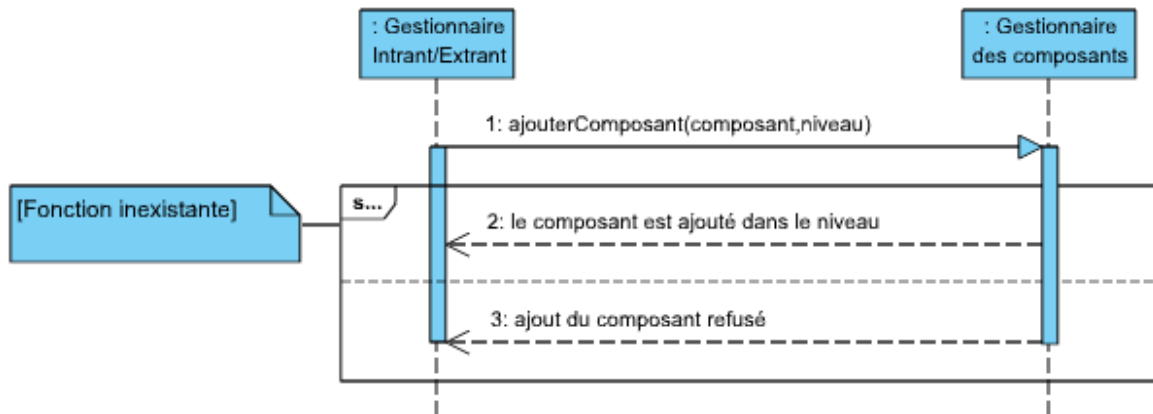


Figure 4 - Ajouter un composant

#### 4.4. Ajouter une fonction

Ajouter une fonction permet d'ajouter une fonction dans le niveau des fonctions. L'ajout en base de données est réalisé une fois que l'enregistrement a eu lieu.

- Pré-condition : Aucune.
- Post-condition : La nouvelle fonction est créée si le nom est unique et est disposé à recevoir des attributions d'exigences.

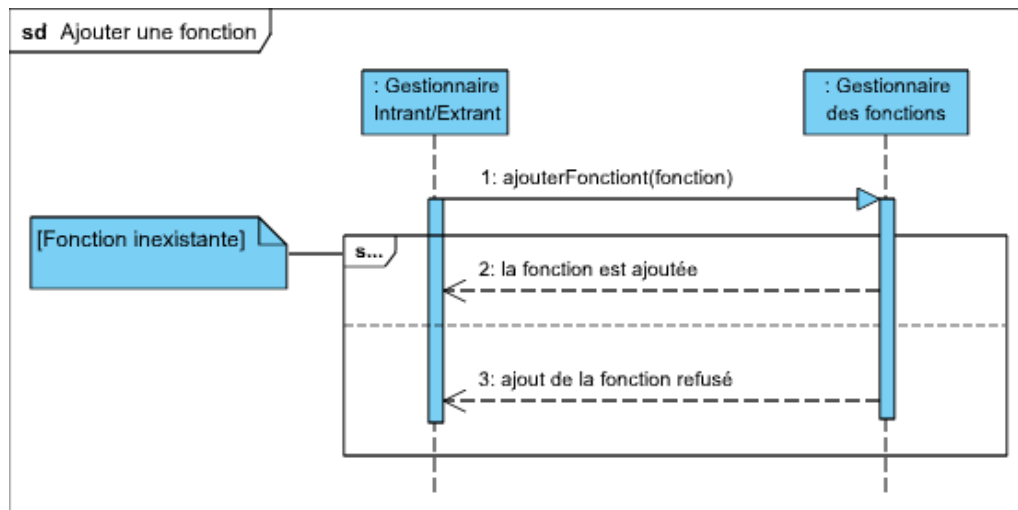


Figure 5 - Ajouter une fonction



#### 4.5. Création d'un arbre hiérarchique pour un regroupement d'exigences

Un nouvel arbre hiérarchique est créé de sorte à accueillir un regroupement d'exigences. L'intérêt de ce dernier est de présenter un ensemble restreint d'exigences dans un arbre hiérarchique d'exigences.

- *Pré-condition* : Il existe au moins un regroupement existant.
- *Post-condition* : Un arbre hiérarchique est disponible pour y présenter les exigences relatives au regroupement d'exigences.

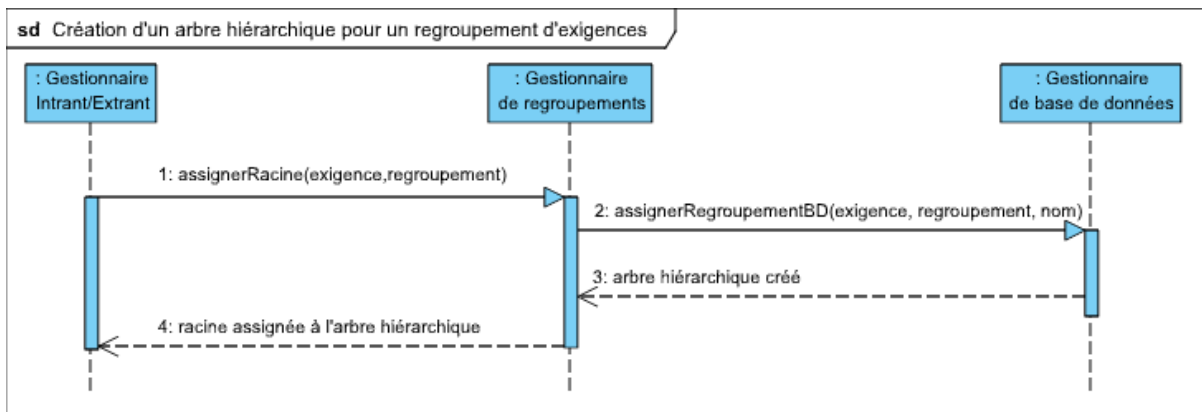


Figure 6 - Création d'un arbre hiérarchique pour un regroupement d'exigences

#### 4.6. Créer un regroupement d'exigences

Lors de la définition d'une exigence au premier niveau de décomposition (racine), un regroupement se crée automatiquement. Cette exigence sera donc l'ancêtre de toute exigence appartenant au regroupement qu'elle constitue. La définition d'une exigence comme racine est réalisée par la création ou le déplacement de cette dernière au premier niveau de décomposition.

- Pré-condition : Une exigence a été définie comme étant une racine.
- Post-condition : Le regroupement est créé.

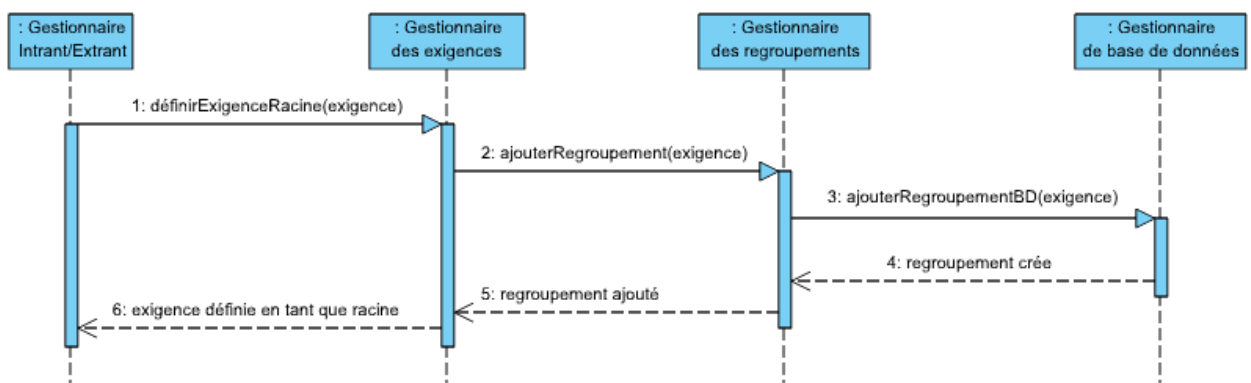


Figure 7 - Créer un regroupement d'exigences

#### 4.7. Enregistrer les composants

Enregistre les modifications relatives aux composants survenues depuis le dernier enregistrement.

- *Pré-condition* : Aucune.
- *Post-condition* : Les modifications apportées aux composants sont enregistrées.

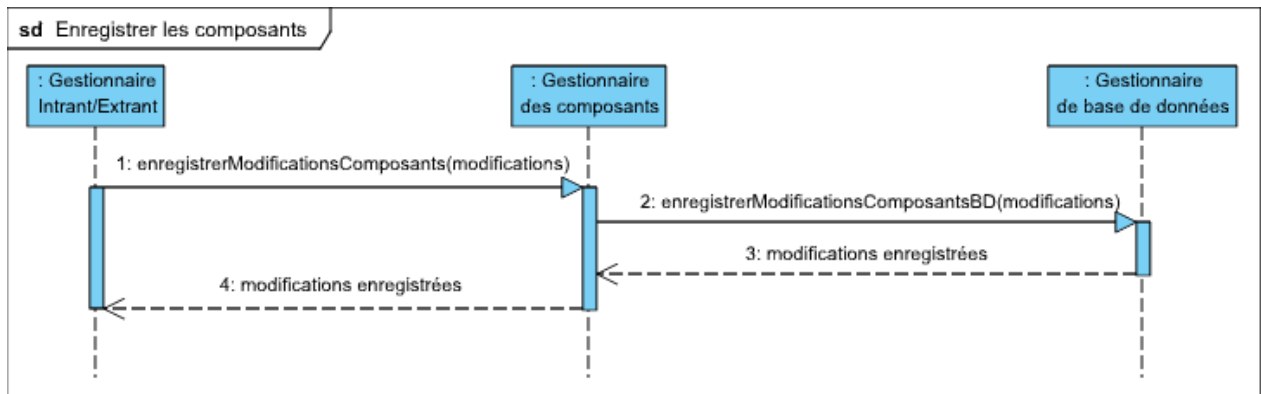


Figure 8 - Enregistrer les composants

#### 4.8. Enregistrer les fonctions

Enregistre les modifications relatives aux fonctions survenues depuis le dernier enregistrement.

- *Pré-condition* : Aucune.
- *Post-condition* : Les modifications apportées aux fonctions sont enregistrées.

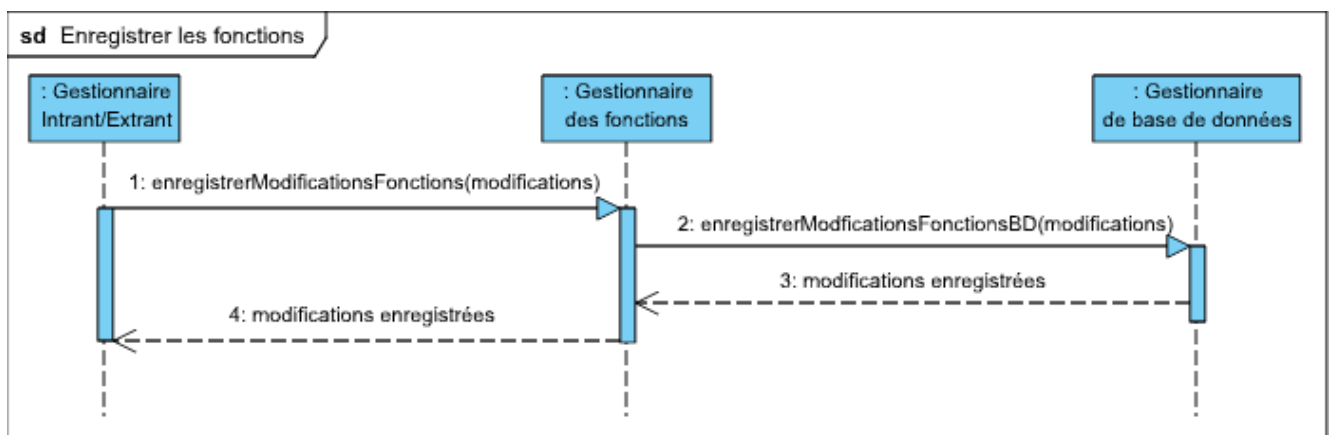


Figure 9 - Enregistrer les fonctions

#### 4.9. Envoyer une attribution d'exigence vers un composant ou une fonction

Envoyer une attribution vers un composant ou une fonction permet de copier une ou plusieurs attributions existantes vers une fonction ou un composant existant.

- Pré-condition : Au moins une attribution existe.
- Post-condition : L'attribution est copiée dans le composant ou la fonction cible si elle n'y était pas présente.

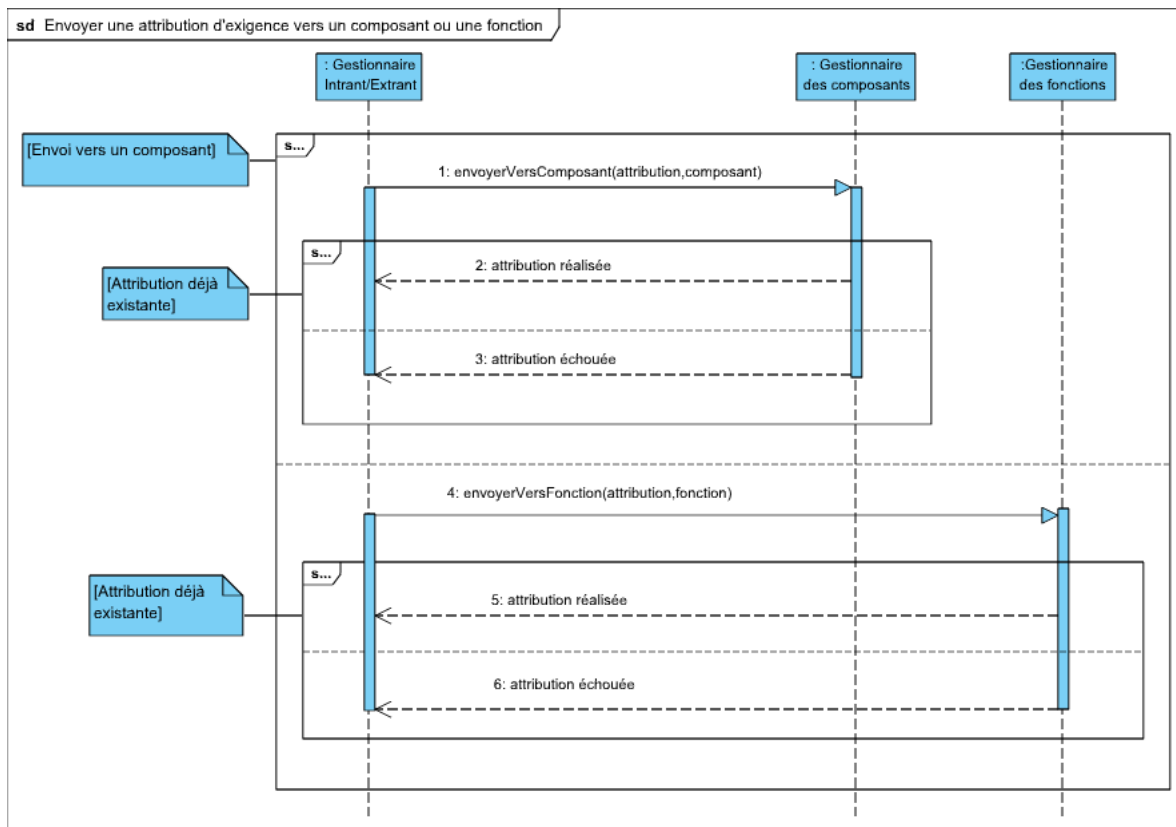


Figure 10 - Envoyer une attribution d'exigence vers un composant ou une fonction

#### 4.10. Générer le tableau d'attribution des exigences à des composants et fonctions

Générer le tableau d'attribution des exigences à des composants permet de générer ce tableau dans un nouveau document Microsoft Excel.

- Pré-condition : Aucune.
- Post-condition : Le tableau d'attribution des exigences à des composants est généré dans un nouveau document Microsoft Excel.

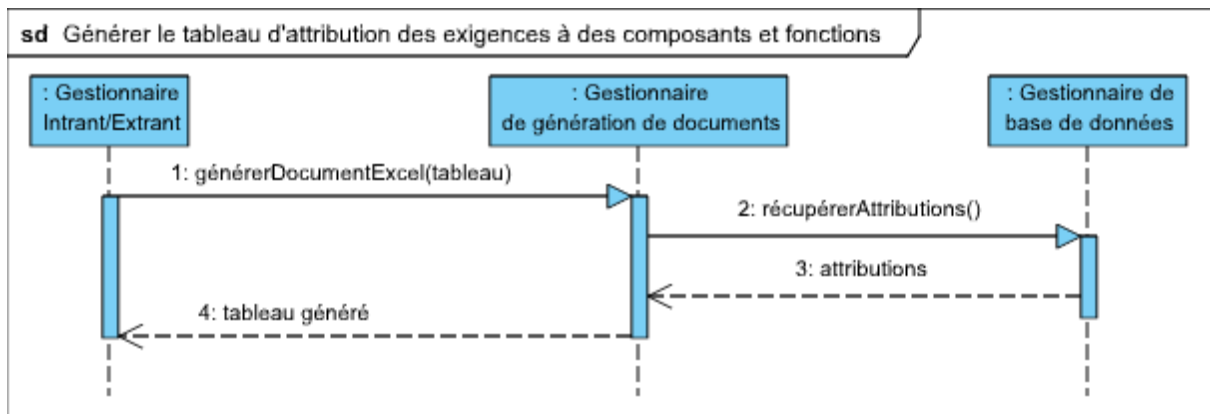


Figure 11 - Générer le tableau d'attribution des exigences à des composants et fonctions

#### 4.11. Renommer un composant

Renommer un composant permet la modification du nom du composant. La modification en base de données est réalisée une fois que l'enregistrement a eu lieu.

- Pré-condition : Le composant à renommer existe.
- Post-condition : Le nom du composant est modifié si le nom de ce dernier n'existe pour aucun de ses composants frères.

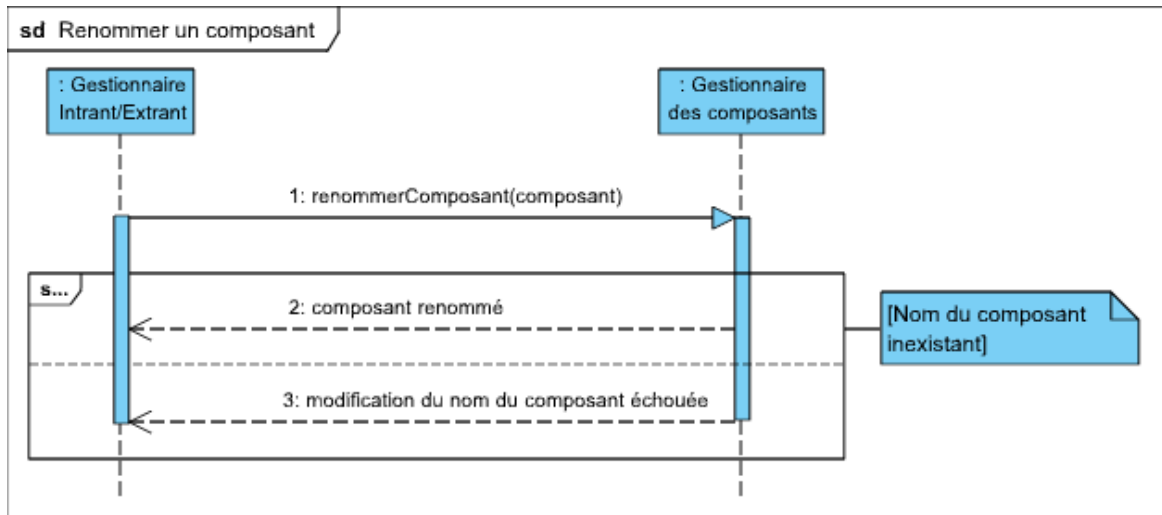


Figure 12 - Renommer un composant

#### 4.12. Renommer une fonction

Renommer une fonction permet la modification du nom de la fonction. La modification en base de données est réalisée une fois que l'enregistrement a eu lieu.

- Pré-condition : La fonction à renommer existe.
- Post-condition : Le nom de la fonction est modifié si le nom de cette dernière n'existe pour aucun des fonctions soeurs.

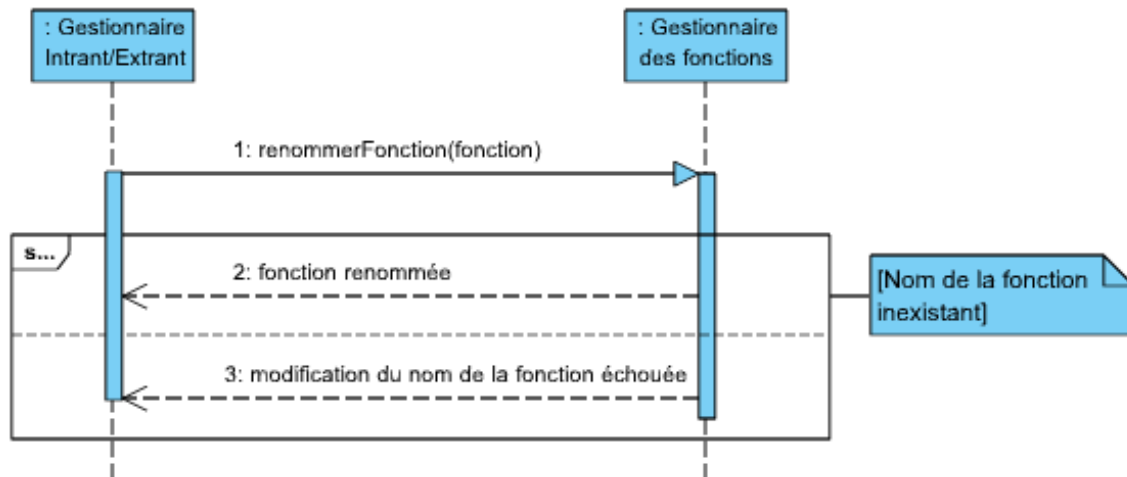


Figure 13 - Renommer une fonction

#### 4.13. Supprimer un composant

Supprime un composant et ses attributions de la hiérarchie des composants. Ni le composant, ni ses attributions ne sont supprimés de la base de données tant que l'enregistrement n'a pas eu lieu.

- Pré-condition : Le composant à supprimer existe.
- Post-condition : Le composant et ses attributions ne sont plus accessibles à l'utilisateur.

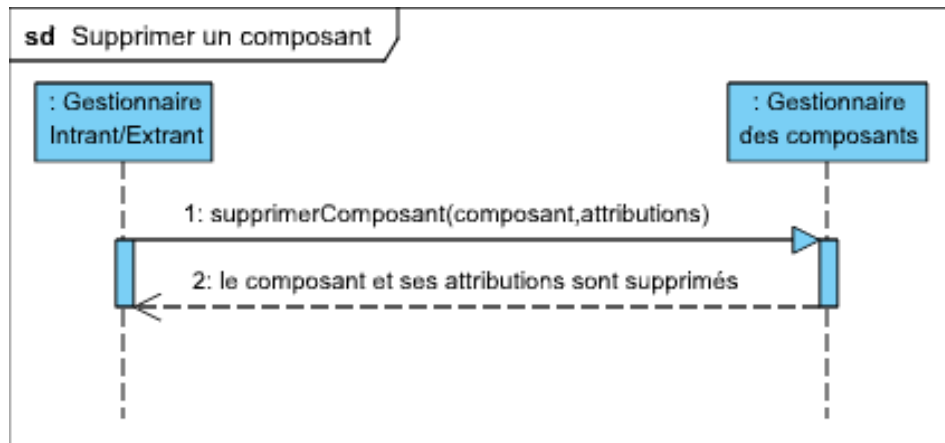


Figure 14 – Supprimer un composant

#### 4.14. Supprimer un regroupement d'exigences

La suppression d'une exigence racine a pour effet de supprimer toutes les exigences qui raffinent cette exigence (les exigences filles). Le regroupement ne possède donc plus aucune exigence et est donc supprimé.

- Pré-condition : Il existe au moins une exigence.
- Post-condition : Le regroupement est supprimé.

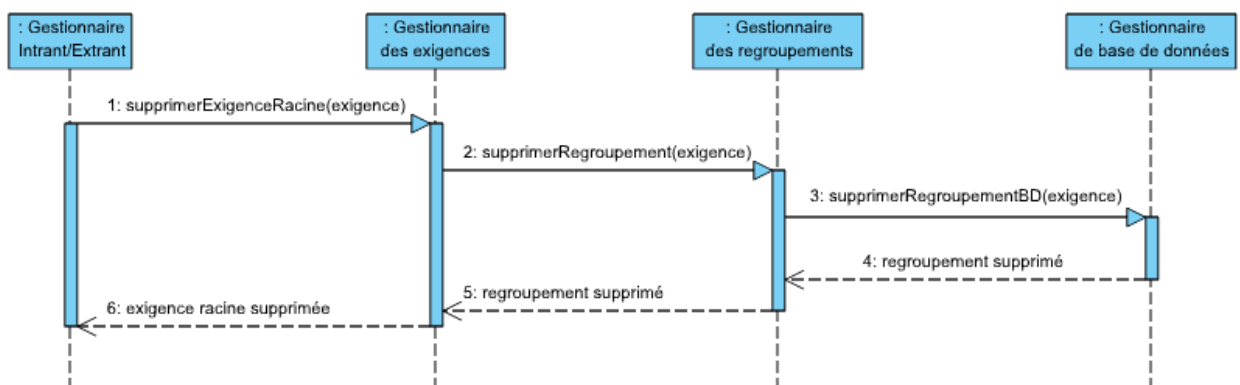


Figure 15 – Supprimer un regroupement d'exigences

#### 4.15. Supprimer une attribution d'exigence d'un composant

Supprimer une attribution d'un composant consiste à retirer une attribution appartenant au composant. Notons que la portée de la suppression s'applique à l'attribution supprimée, et pas sur l'exigence source de l'attribution. La suppression de l'attribution en base de données est réalisée une fois que l'enregistrement a eu lieu.

- Pré-condition : Une attribution existe.
- Post-condition : L'attribution destinée à être supprimée n'est plus attribuée au composant à partir duquel la suppression a lieu.

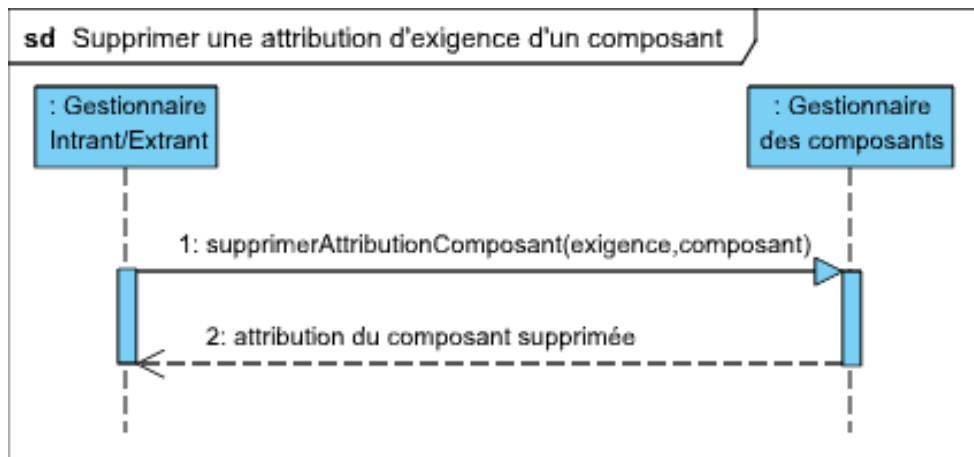


Figure 16 - Supprimer une attribution d'exigence d'un composant



#### 4.16. Supprimer une attribution d'exigence d'une fonction

Supprimer une attribution d'une fonction consiste à retirer une attribution appartenant à la fonction. Notons que la portée de la suppression s'applique à l'attribution supprimée, et pas sur l'exigence source de l'attribution. La suppression de l'attribution en base de données est réalisée une fois que l'enregistrement a eu lieu.

- Pré-condition : Une attribution existe.
- Post-condition : L'attribution destinée à être supprimée n'est plus attribuée à la fonction à partir de laquelle la suppression a lieu.

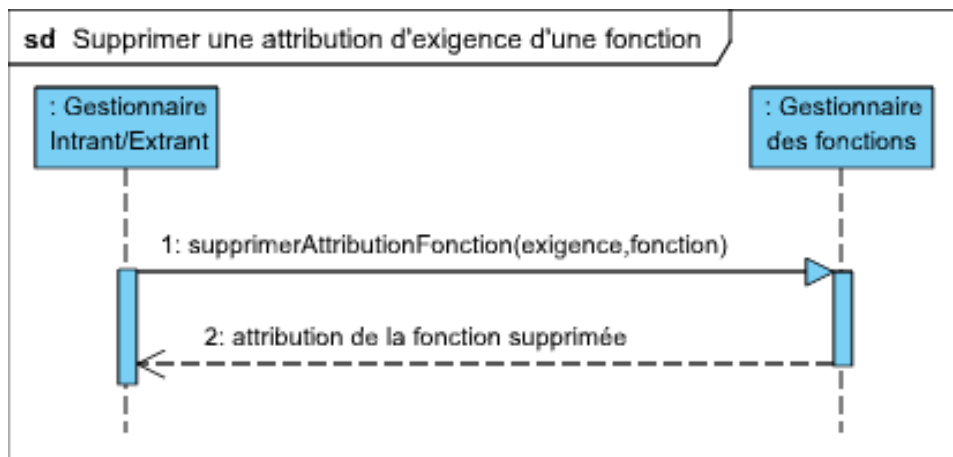


Figure 17 - Supprimer une attribution d'exigence d'une fonction

#### 4.17. Supprimer une fonction

Supprime une fonction et ses attributions de la liste des fonctions. Ni la fonction, ni ses attributions ne sont supprimés de la base de données tant que l'enregistrement n'a pas eu lieu.

- Pré-condition : La fonction à supprimer existe.
- Post-condition : La fonction et ses attributions ne sont plus accessibles à l'utilisateur.

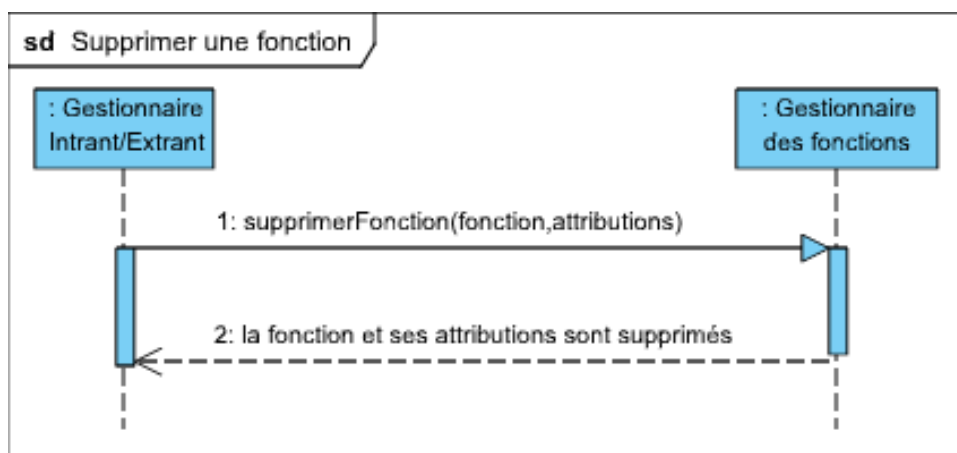


Figure 18 – Supprimer une fonction

#### 4.18. Valider une attribution d'un composant

Valide une attribution dont l'exigence source a subi une modification. Cette validation a pour but de mettre en évidence que l'exigence à laquelle elle fait référence a été modifiée. La validation sera enregistrée dans la base de données une fois que l'enregistrement aura eu lieu.

- Pré-condition : Au moins une attribution d'un composant est à valider.
- Post-condition : L'attribution est validée.

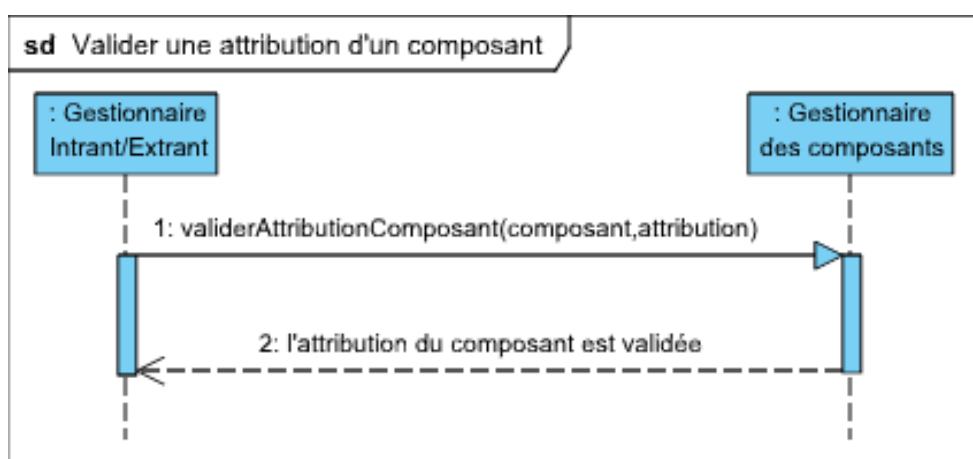


Figure 19 – Valider une attribution d'un composant

#### 4.19. Valider une attribution d'une fonction

Valide une attribution dont l'exigence source a subi une modification. Cette validation a pour but de mettre en évidence que l'exigence à laquelle elle fait référence a été modifiée. La validation sera enregistrée dans la base de données une fois que l'enregistrement aura eu lieu.

- Pré-condition : Au moins une attribution d'une fonction est à valider.
- Post-condition : L'attribution est validée.

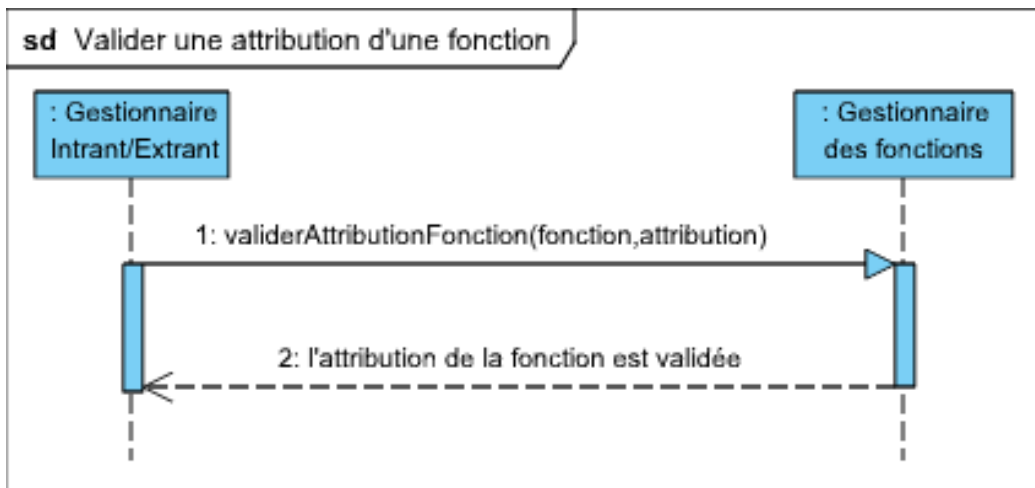


Figure 20 – Valider une attribution d'une fonction

#### 4.20. Vérifier le tableau d'attribution des exigences à des composants/fonctions

Vérifier le tableau d'attribution des exigences à des composants permet de mettre en évidence les composants et fonctions vides. Ces dernières pouvant éventuellement être vide – sans aucune attribution - dû à un oubli du concepteur.

- Pré-condition : Aucune.
- Post-condition : Les composants et fonctions vides sont mis en évidence.

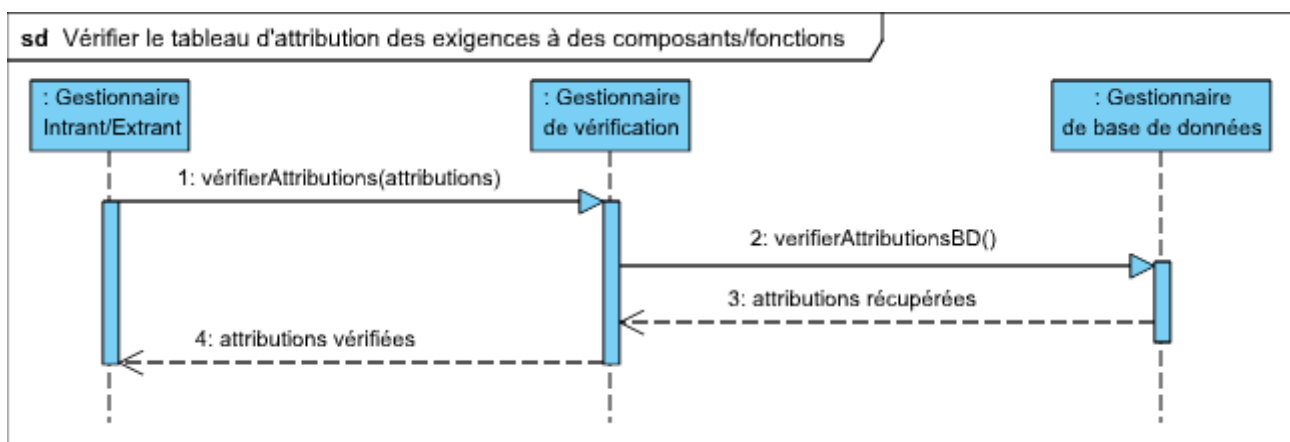


Figure 21 - Vérifier le tableau d'attribution des exigences à des composants/fonctions

#### 4.21. Vérifier un regroupement d'exigences

Vérifier un regroupement d'exigences consiste à mettre en évidence des incohérences relatives aux exigences. Prenons par exemple le cas d'une exigence d'interface qui n'est jamais utilisée par une exigence système. Cette dernière ne serait alors d'aucune utilité et pourrait donc être supprimée.

- Pré-condition : Au moins un regroupement d'exigences existe.
- Post-condition : Les regroupements sujets à vérification sont vérifiés, et par conséquent l'utilisateur est averti du résultat de la vérification.

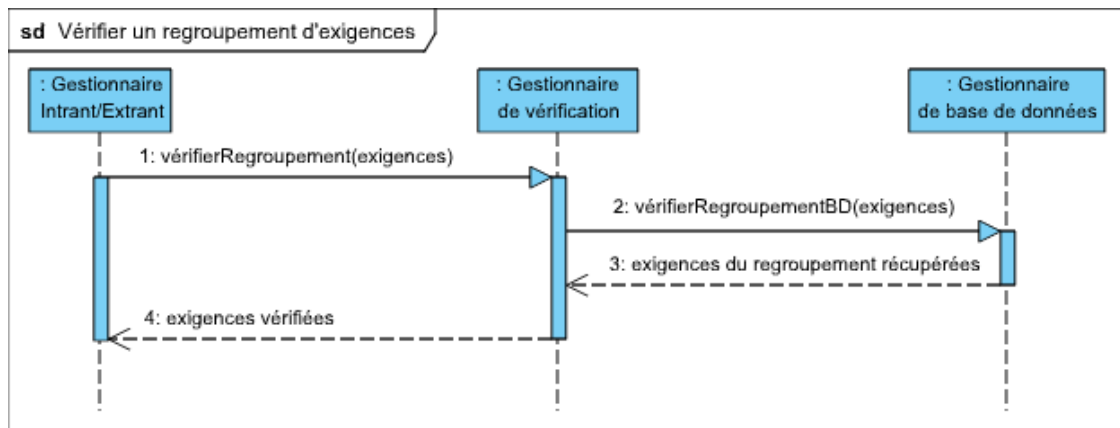


Figure 22 - Vérifier un regroupement d'exigences

## 5. DIAGRAMME DE CLASSES

Le diagramme de classes ci-dessus inclut les fonctionnalités liées au glossaire de GenSpec, ainsi qu'à la gestion multi-documents. C'est pour cette raison qu'il contient plusieurs packages distincts en terme de fonctionnalités. Le package commun représente les classes qui sont utilisées d'une part par le glossaire, et d'autre part, par la gestion multi-documents.

Les différentes classes représentées découlent du diagramme de composants présenté dans la section précédente. Dès lors, pour en savoir plus sur la sémantique des différentes classes, se référer à la section qui présente les différents composants.

### 5.1. Diagramme

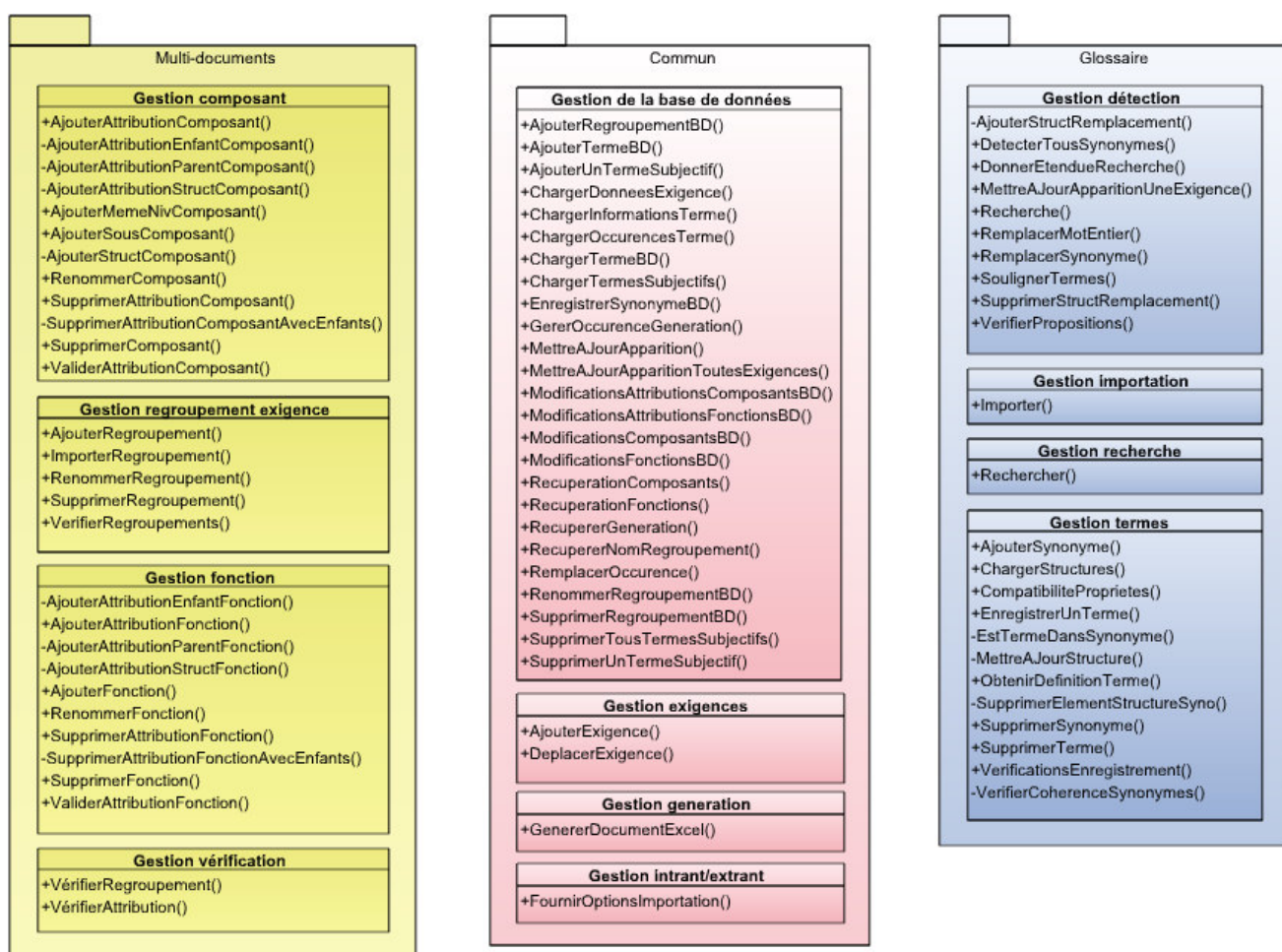


Figure 23 - Diagramme de classes

## **6. SCHÉMAS DE BASE DE DONNÉES**

Cette section décrit les différents schémas relatifs à la base de données de GenSpec. Par souci de clarté, deux types de schémas ont été réalisés. Premièrement, les schémas réduits décrivant uniquement les fonctionnalités relatives à la gestion multi-documents. Deuxièmement, les schémas globaux exposant, dans un seul schéma, l'intégration des nouvelles entités et relations avec celles existantes.

Notons enfin que des fonctionnalités relatives au glossaire pour GenSpec apparaissent également dans les schémas, en effet, le développement de ces dernières se réalise en parallèle avec la gestion multi-documents.

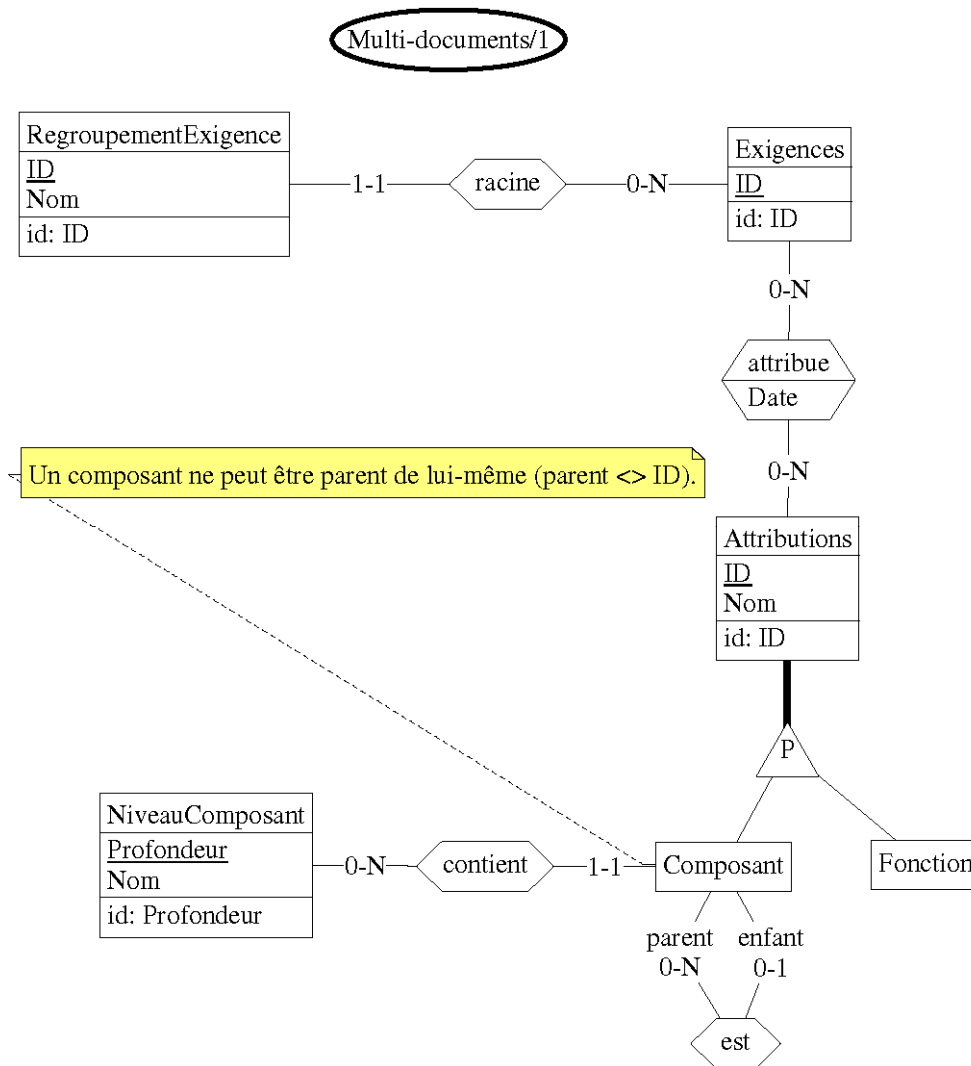
### **6.1. Schémas réduits**

Cette section décrit les schémas conceptuels entités-associations des deux parties de la gestion multi-documents, à savoir :

- l'attribution des exigences à des composants ;
- la centralisation des informations.

**6.1.1. Schéma conceptuel**

Le schéma conceptuel ci-dessous permet de répondre aux besoins de l'attribution des exigences à des composants, ainsi qu'à la centralisation des informations. Les tables qui en découlent sont à intégrer dans la base de données existante de GenSpec.



**Figure 24 - Schéma conceptuel pour la gestion multi-documents**

### 6.1.2. Schéma relationnel

Notons que la transformation du schéma conceptuel en schéma relationnel pouvait se réaliser de plusieurs manières. Le choix de transformer la relation "Is-a" s'est porté sur une transformation descendante, c'est pourquoi nous ne retrouvons plus l'entité "Regroupement" dans le schéma relationnel. Cependant, les attributs de cette dernière ont été importés dans les tables "Composant" et "Fonction". Enfin, signalons qu'une contrainte a découlé de cette transformation.

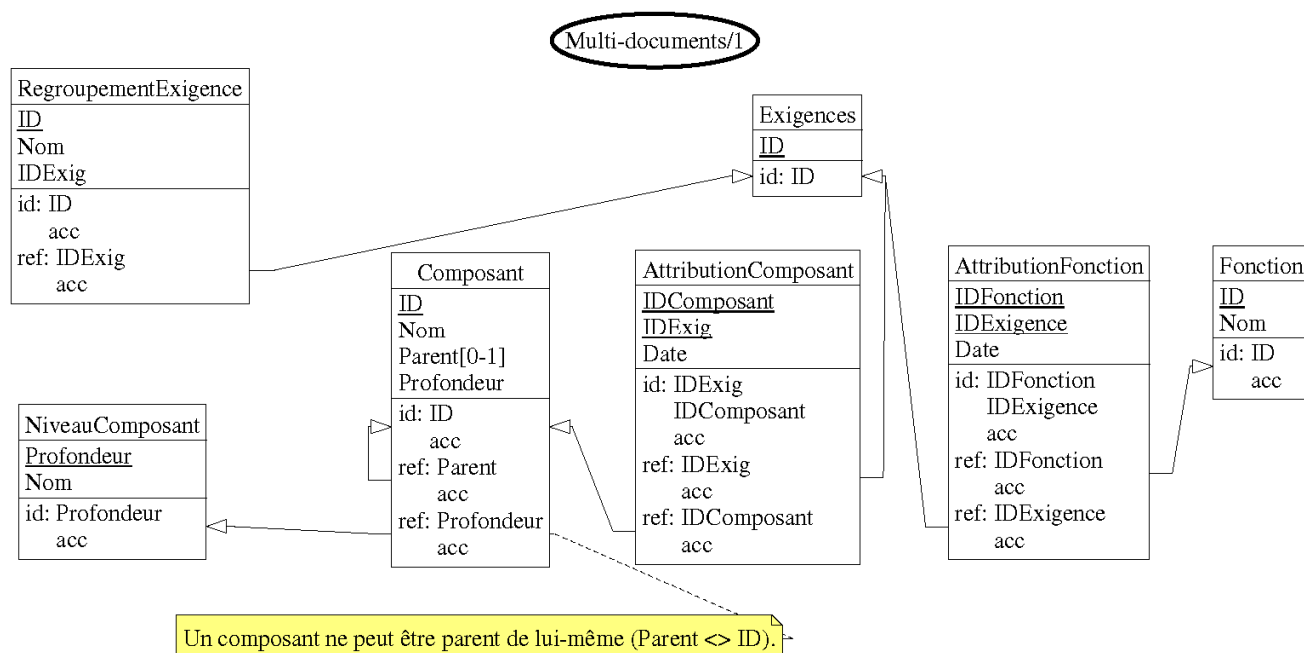


Figure 25 - Schéma relationnel pour la gestion multi-documents

### 6.1.3. Description sémantique

Il s'agit dans cette section de donner une description informelle des entités, relations et attributs présents dans le schéma ci-dessus.

#### 6.1.3.1. Les entités et leurs attributs

- **Composant** : Décrit chaque composant du système.
  - **ID** : L'identifiant du composant.
  - **Nom** : Le nom donné au composant
  - **Parent** : L'identifiant du composant parent
  - **Profondeur** : La profondeur à laquelle se trouve le composant dans la hiérarchie des composants.
- **Fonction** : Décrit chaque fonction du système.
  - **ID** : L'identifiant de la fonction.
  - **Nom** : Le nom donné à la fonction.
- **NiveauComposant** : Décrit pour chaque niveau de composant, son nom et sa profondeur dans l'arbre hiérarchique des composants.



- **Profondeur** : Profondeur du niveau dans l'arbre hiérarchique des composants.
- **Nom** : Le nom du niveau. Ce nom est donné afin de pouvoir parler d'un niveau sans pour autant évoquer sa profondeur. Par exemple, le niveau des applications, des composants systèmes, des composants unitaires, etc.
- **AttributionComposant** : Décrit chaque attribution d'exigence liée à un composant.
  - **IDComposant** : L'identifiant du composant.
  - **IDExig** : L'identifiant de l'exigence, source de l'attribution.
  - **Date** : La date de l'attribution.
- **AttributionFonction** : Décrit chaque attribution d'exigence liée à une fonction.
  - **IDFonction** : L'identifiant de la fonction.
  - **IDExig** : L'identifiant de l'exigence, source de l'attribution.
  - **Date** : La date de l'attribution.
- **RegroupementExigence** : Décrit pour chaque niveau de composant, son nom et sa profondeur dans l'arbre hiérarchique des composants.
  - **ID** : L'identifiant du regroupement.
  - **Nom** : Le nom du regroupement.
  - **IDExig** : L'identifiant de l'exigence.
- **Exigences** : Décrit l'entité existante "Exigences" dans la base de données de GenSpec (seul l'ID est représenté, les autres attributs n'apportant aucune information pertinente dans ce contexte).
  - **ID** : L'identifiant de l'exigence.

#### 6.1.3.2. Les relations entre les entités

- **contient** : Représente le contenu d'un niveau de composants.
- **est** : Représente la hiérarchie entre les composants.
- **attribue** : Représente le lien existant entre l'attribution et son exigence source.
  - **Date** : La date de l'attribution
- **racine** : Représente le rôle de racine que joue une exigence dans un regroupement.

## 6.2. Schéma globaux

Cette section présente le schéma conceptuel ainsi que le schéma relationnel de la base de données de GenSpec. Notons que les informations relatives au glossaire et à la gestion multi-documents y ont été intégrées.

Concrètement, GenSpec utilise une base de données Microsoft Access. Aucun schéma de base de données n'étant disponible dans la documentation de GenSpec, il a été nécessaire de le recréer. Pour ce faire, un outil permettant de convertir une base de données Access en DB-Main a été utilisé [3]. Les schémas ci-dessous, convertis dans le formalisme de DB-Main, représentent donc les schémas de base de données de GenSpec.

Afin de différencier l'existant des fonctionnalités liées au glossaire et à la gestion multi-documents, des couleurs ont été utilisées. Ces couleurs sont les suivantes :

- le bleu montre les aspects de la gestion multi-documents ;
- le vert montre les aspects du glossaire ;
- le noir montre les informations existantes, c'est-à-dire précédant tout développement lié au glossaire ou à la gestion multi-documents.



6.2.2. Schéma relationnel

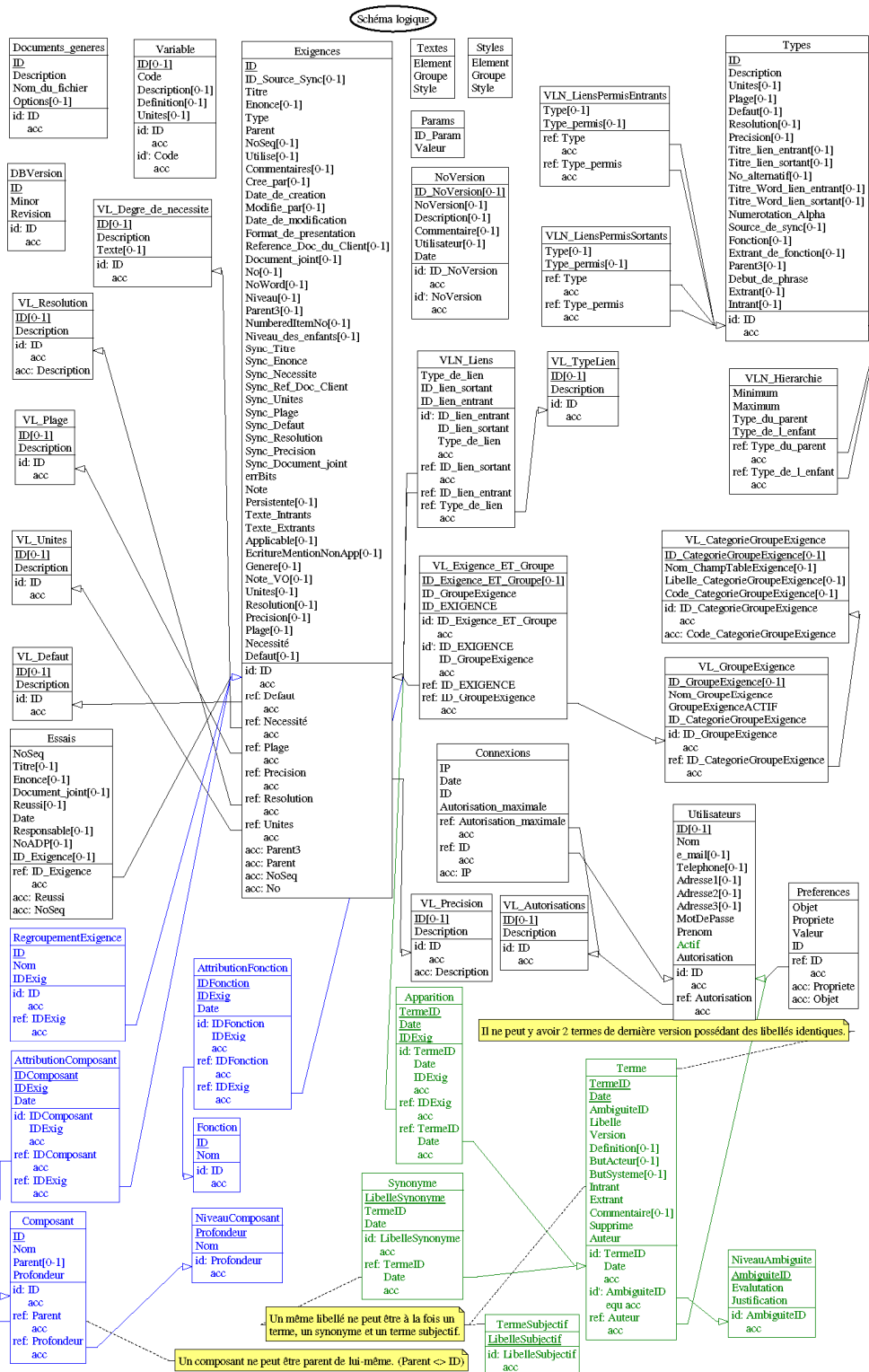


Figure 27 - Schéma relationnel

## 7. CONCLUSION

Ce document a exposé les différents diagrammes réalisés liés à la gestion multi-documents pour GenSpec.

Ces diagrammes ont permis de représenter les nouvelles fonctionnalités qui vont être intégrées à GenSpec. De plus, il a permis de représenter les différentes interactions avec les fonctionnalités existantes dans GenSpec.

Enfin, il permet de passer à la phase suivante du projet, à savoir le codage des fonctionnalités liées à la gestion multi-documents.

## Annexe I

# Sources linguistiques d'ambiguïté

La dernière modification du contenu de ce document date du 26 octobre 2006.

*Document complémentaire*

# **Sources linguistiques de l'ambiguïté**

## **GenSpec**

**NOVEMBRE 2006**

Ce résumé a été établi à partir de [Berry and Kamsties, 2003]. Il fournit une définition des différents types d'exigences ainsi qu'un ensemble d'informations et de règles générales sur la gestion de l'ambiguïté au sein des exigences.

## 1. FORMES D'AMBIGUÏTÉS

### 1.1. Ambiguïté Lexicale

L'ambiguïté lexicale survient lorsqu'un mot possède plusieurs sens. Cette ambiguïté peut être subdivisée en 2 groupes : homonymie et polysémie.

#### 1.1.1. Homonymie

L'homonymie survient quand 2 mots différents possèdent la même représentation écrite et phonétique, alors qu'ils sont de sens distincts et d'étymologies différentes – i.e. différentes histoire de développement - .

*Exemple* : « Canon » (d'une armée) et « Canon » (de la messe) sont homonymes. [2]

#### 1.1.2. Polysémie

La polysémie survient quand un mot possède plusieurs sens apparentés mais seulement une étymologie.

##### A.1.1.1.1 Polysémie systématique

La polysémie systématique survient quand la raison de la polysémie est confuse entre les classes, c'est-à-dire, entre l'unité et le type, et entre le processus et le produit.

*Exemples* :

- Confusion entre unité et type : "J'aime ce manteau" peut référer à un manteau particulier ou à un type de manteau.
- Confusion entre produit et processus : "Construction" peut référer au processus de construction tout comme au résultat final de ce processus.

### 1.2. Ambiguïté Syntaxique

L'ambiguïté syntaxique, appelé aussi "ambiguïté structurelle", survient quand une séquence de mots donnés peut correspondre à plus d'une structure grammaticale, chacune possédant un sens différent.

#### 1.2.1. Ambiguïté analytique

L'ambiguïté analytique survient quand le rôle des constituants à l'intérieur d'une expression ou d'une phrase est ambigu.

*Exemple* : « Le professeur d'art tibétain » peut être lu comme « Le professeur (d'art tibétain) » ou « Le professeur d'art qui est tibétain »".



### 1.2.2. Ambiguïté d'attachement

L'ambiguïté d'attachement survient quand un constituant particulier d'une phrase, tel qu'une phrase prépositionnelle ou une clause relative, peut être attaché à 2 parties d'une phrase.

*Exemple* : « *Les policiers on tué les émeutiers avec des armes* » possède deux significations différentes. En effet, on peut l'interpréter comme « *les policiers ont tué les émeutiers qui possédaient des armes* » ou « *les policiers ont utilisé des armes pour tuer les émeutiers* ».

### 1.2.3. Ambiguïté de coordination

L'ambiguïté de coordination survient quand :

- Plus d'une conjonction, *et* ou *ou*, est utilisé dans une phrase
- Une conjonction est utilisée avec un modificateur

*Exemple* : « *Jeune homme et femme* » peut référer à « *[jeune homme] et femme* » ou « *jeune [homme et femme]* »

### 1.2.4. Ambiguïté elliptique

Une ellipse est un trou dans une phrase causé par omission d'un constituant syntaxique ou lexical nécessaire.

L'ambiguïté elliptique survient quand il n'est pas certain qu'une phrase contienne ou non une ellipse.

*Exemple* : "Peter connaît un homme plus riche que Trump" a 2 sens. Soit Peter connaît un homme qui est plus riche que Trump ou Peter connaît un homme plus riche que toute personne que Trump connaît.

## 1.3. Ambiguïté sémantique

L'ambiguïté sémantique survient quand il n'existe plus d'une manière de lire une phrase dans son contexte bien qu'elle ne contienne aucune ambiguïté structurelle ou lexicale.

L'ambiguïté sémantique peut être causée par :

### 1.3.1. Ambiguïté de coordination

*Voir 1.2.3. Ambiguïté de coordination.*

### 1.3.2. Ambiguïté référentielle

L'ambiguïté référentielle peut survenir à l'intérieur d'une phrase ou entre les phrases et son contexte de discours. *Voir référence 1.4.1 Ambiguïté référentielle.*

### 1.3.3. Ambiguïté de possibilité

L'ambiguïté de possibilité survient quand les opérateurs *chaque, tous, quelques, etc.* peuvent entrer dans différentes possibilités de relations avec les autres constituants de la phrase.

*Exemple* : « *Personne n'a vu un cochon avec des ailes* » possède 2 possibilités. Premièrement, on peut le comprendre comme il n'existe pas de cochons avec des ailes ou deuxièmement, il existe un cochon mythique avec des ailes et personne ne l'a jamais vu.

### 1.4. Ambiguïté pragmatique

Pragmatique correspond à l'étude des relations entre le langage et le contexte.

L'ambiguïté pragmatique survient quand une phrase possède plusieurs sens dans le contexte dans lequel elle est prononcée.

#### 1.4.1. Ambiguïté référentielle

Une anaphore est un élément d'une phrase qui dépend pour sa référence de la référence d'un autre élément ou d'une autre phrase. Cet autre élément est appelé *antécédent* et doit apparaître plus tôt dans la même phrase ou dans la phrase précédente.

L'ambiguïté référentielle survient quand une anaphore peut prendre sa référence de plus d'un élément, chacun pouvant jouer le rôle de l'antécédent.

*Exemple* : « *Les camions devraient traiter les chemins avant qu'ils ne gèlent.* » L'antécédent de l'anaphore « *ils* » peut référer aussi bien aux camions qu'aux chemins.

#### 1.4.2. Ambiguïté déictique

L'ambiguïté déictique survient quand des pronoms, des adverbes de place, tel que *maintenant* et *ici*, et d'autres éléments grammaticaux, tel que le temps, ont plus d'une référence pointée dans le contexte. Le contexte comprend une personne dans une conversation, un endroit particulier, une instance particulière de temps, ou une expression dans une phrase précédente ou à venir.

### 1.5. Imprécision et généralités

L'imprécision a une application particulière vis-à-vis des exigences d'un programme. Une exigence est imprécise si l'on ne sait mesurer si une exigence est remplie ou non.

*Remarque* : Une exigence non-fonctionnelle est souvent vague parce qu'il n'y a pas de manière précise de la décrire et de la mesurer.

La différence entre l'imprécision et la généralité est l'existence de cas limites. Une expression imprécise résiste aux enquêtes grâce aux cas limites, tandis qu'une expression générale peut être faite de manière plus précise.

*Exemple* :

- "Cousin" en anglais est général. Il correspond aussi bien au masculin qu'au féminin mais il peut être décrit de manière plus précise si besoin est.

- "Cousin" n'est pas vague. En effet, il n'y a aucun doute sur le fait de dire si une personne est un cousin ou non.

## 2. TECHNIQUES POUR S'OCCUPER DE L'AMBIGUÏTÉ

Cette section présente différentes techniques pour diminuer l'ambiguïté. Ces techniques peuvent être divisées en trois groupes en rapport avec les activités d'ingénierie des exigences sur lesquels elles s'appliquent.

Pour l'*élicitation des exigences*, on retrouve deux stratégies pour minimaliser les ambiguïtés :

- Premièrement un contexte doit être établi car le langage est toujours interprété dans un contexte.
- Deuxièmement, la paraphrase effectuée par l'analyste de ce qu'il a compris des clients et utilisateurs dans ses propres mots afin que les clients et utilisateurs remarquent leurs propres ambiguïtés.

Pour la *documentation des exigences*, il y a trois stratégies qui peuvent être différenciées pour éviter l'ambiguïté dans les exigences écrites :

- Premièrement, la précision du langage naturel peut être renforcée.
- Deuxièmement, plus d'information contextuelle peut être fournie pour permettre au lecteur de résoudre les ambiguïtés lui-même.
- Troisièmement, les conventions sur comment les phrases ambiguës devraient être interprétées peuvent être établies entre l'auteur et le lecteur.

Pour la *validation des exigences*, il y a quatre stratégies pour détecter les ambiguïtés :

- Premièrement, la formalisation des exigences informelles.
- Deuxièmement, la recherche de modèles particuliers d'ambiguïtés.
- Troisièmement, comparer les interprétations d'un document par différents participants. Si leurs interprétations diffèrent, c'est qu'il y a des ambiguïtés dans le document général.
- Quatrièmement, la communication d'une interprétation à l'auteur des exigences après quoi il peut facilement voir quels sont les incompréhensions.

### 2.1. Documentation : Amélioration de la précision de langage naturel

Glossaires, guides de style, modèles de phrases et les langages contrôlés améliorent la précision et diminuent l'ambiguïté du langage naturel.

Un *glossaire* définit les termes et les phrases importantes utilisées dans un document d'exigences. D'où, cela permet d'éviter l'ambiguïté lexicale. Cela demande beaucoup d'effort pour créer et valider un glossaire mais l'effort initial est payant puisque ce glossaire peut être réutilisé dans des futurs projets dans le même domaine d'application.

Un *guide de style* constitue un ensemble de règles concernant les bonnes pratiques vis-à-vis de l'écriture des exigences. Elles peuvent être utilisées également pour vérifier l'existence de problèmes possibles vis-à-vis des exigences.

Un *langage contrôlé* est un sous-ensemble d'un langage naturel pour l'utilisation dans un environnement spécifique. L'objectif d'un langage contrôlé est d'améliorer la lisibilité et la compréhensibilité de tout type de documentation technique en réduisant l'ambiguïté inhérente du langage naturel à travers une grammaire restreinte et un vocabulaire fixé.

## 2.2. Documentation : Fournir plus d'informations contextuelles

Il existe différentes sources pour fournir de l'information contextuelle (et pour laisser moins de place à l'interprétation) :

- **Commentaire** : Un commentaire peut être utilisé pour expliquer le cadre d'une exigence.
- **Raisonnement** : Un raisonnement décrit pourquoi on a besoin d'une exigence.
- **Critère de capacité** : Un critère de capacité décrit une condition qu'un programme doit remplir pour satisfaire une exigence.
- **Cas de test** : Un cas de test, qui est une forme plus élaborée de critère de capacité, décrit un intrant possible et l'extrait qui est attendu explicitement.
- **Exigence inverse** : Une exigence inverse décrit fonctionnellement ce que le programme ne doit pas réaliser. *Remarque* : Les exigences inverses sont souvent utilisées de manière abusive pour exprimer des exigences non-fonctionnelles. Cependant, à l'origine, une exigence inverse exclut des interprétations possibles d'une ou plusieurs exigences. L'utilisation d'exigences inverses permet alors de diminuer l'ambiguïté des exigences fonctionnelles.
- **Informations de traçabilité** : De l'information de traçabilité sur les dépendances entre les exigences (la traçabilité exigences-exigences) permet de résoudre l'ambiguïté d'une exigence si le lien, permettant d'identifier une relation proche entre les exigences, fournit assez d'informations contextuelles.

## 2.3. Documentation : Établir des conventions d'interprétation

Il n'existe pas vraiment de techniques pour établir des conventions d'interprétations car cette stratégie est très pragmatique.

## 3. ÉVITER LES AMBIGUITÉS DANS LES SPÉCIFICATIONS ÉTABLIES EN LANGAGE NATUREL

On couvre, au sein de cette partie, un ensemble d'ambiguïtés courantes.

### 3.1. Mots ambigus, vagues et incertains

Les mots suivants communément utilisés sont ambigus et devraient être évité ou utiliser avec précautions au sein des exigences :

#### Mots ambigus :

- **Et** : Il est utilisé pour :
  - marquer la concurrence entre des évènements ou actions
  - marquer que plusieurs conditions doivent être rencontrées

- marquer un ordre temporel des évènements et actions
- établir une énumération n'impliquant pas du tout d'ordre.

On peut résoudre le problème de différentes manières selon les cas :

- Diviser la phrase originelle en plusieurs phrases
  - Écrire ce que cela veut vraiment dire
  - Utiliser "En même temps" pour la concurrence des évènements ou actions
  - Utiliser "Et après" pour la concurrence des évènements et actions
  - Utiliser le simple "Et" pour marquer que plusieurs conditions doivent être rencontrées de manière concurrente.
- Inclure : Cela peut correspondre à "consiste à" ou "contient un sous-ensemble". Le problème peut être résolu en notant ce qu'il signifie.
  - Après, Avant, Suivant, Précédent, etc. : Chacun de ces mots possède un sens bien établi mais aussi une référence variable. Le problème peut être résolu par une utilisation consciencieuse accompagné par la vérification de l'unicité du référent.
  - Minimum et maximum : Minimum et maximum ne permettent pas de spécifier si l'intervalle déterminé est ouvert ou fermé. Le problème peut être résolu en notant ce qu'il signifie.
  - Ou : Cela peut correspondre à "l'un ou l'autre mais pas les deux" - ou exclusif – ou "l'un ou l'autre ou les deux" – ou inclusif -. Le problème peut être résolu en notant ce qu'il signifie.

#### **Mots vagues :**

Les mots vagues devraient être évités par des quantifications précises. Des mots tel que : acceptable, précis, approprié, facile, efficace, essentielle, immédiat, minimum, maximum, périodiquement, suffisant, facile d'utilisation (*user friendly*), etc.

#### **Mots incertains :**

L'incertitude est exprimée en utilisant des mots tel que : non limité à, peut, probablement, possiblement, habituellement, etc. Ces mots devraient être évités en faveur d'une liste complète, d'une décision complète, d'une estimation précise de probabilité, etc. Si un fait est inconnu ou incertain au moment de la rédaction, le besoin de plus de recherche doit être noté explicitement. De cette façon, le lecteur connaît l'incertitude, le degré d'incertitude et l'intention de l'enlever.

### **3.2. Quantifications**

Les termes quantificateurs sont des termes de langage naturel qui sont équivalent au quantificateur universel et au quantifieur existentiel en mathématique.

### 3.2.1. Tous et Chacun

*Tous* et *chacun* sont des quantifieurs universel équivalents dans le sens où chacun est utilisé pour décrire des propriétés qui appartiennent à tous les membres d'un ensemble. Dans toute phrase contenant un quantificateur universel, il y a un risque pour que la phrase soit fautive simplement parce que très peu de spécifications universelles à propos du monde n'ont pas d'exceptions.

*Exemple* : « *Tout le monde a un numéro d'assurance unique* ». Si l'on abstrait le fait que la phrase ait du sens ou non, on considère qu'il n'y a personne qui n'a pas ou plus d'un numéro d'assurance. On peut considérer la phrase de 2 façons distinctes :

1. Chaque personne a son propre numéro d'assurance;
2. toutes les personnes partagent un même numéro d'assurance unique.

Voir l'Article de référence (point 5.2.1) [1] pour plus d'exemples.

### 3.2.2. Un, Tous, Chacun, Certains, Quelque, et Le comme quantifieur

Selon l'emploi et les combinaisons réalisées vis-à-vis de ces quantifieurs, de nombreuses interprétations de la même phrase peuvent être réalisées.

*Exemple* : Prenons les phrases :

- « *Un bureau a une porte reliant le bureau à un couloir.* »
- « *Le bureau a une porte reliant le bureau à un couloir.* »

De part ces 2 cas, on voit distinctement que dans la première phrase, l'emploi de « *un* » implique une certaine généralité tel que « *chaque* » vis-à-vis du terme « *bureau* ». Par contre, l'utilisation du mot « *le* » implique une référence unique à un bureau.

Voir l'Article de référence (point 5.2.2) [2] pour plus d'exemples.

### 3.2.3. Beaucoup et Peu

*Beaucoup* et *peu* souffrent du même problème que *Tous*. Chacun d'eux est pluriel mais ceux-ci ne donnent jamais d'indices précis sur le nombre d'élément qu'ils représentent.

Voir l'Article de référence (point 5.2.3) [2] pour plus d'exemples (exemples s'appliquant uniquement à l'anglais).

## 3.3. Seulement, Aussi et autres

*Seulement*, *Aussi* et d'autres mots similaires rencontrent le même problème : celui d'un placement correct au sein d'une phrase. Le sens de la phrase contenant un de ces mots dépend fortement du placement d'un de ces mots problématiques.

### 3.3.1. Seulement

Pour être correct, un *seulement* devrait précéder immédiatement le mot ou la phrase qu'il limite.

*Exemple* : Si l'on souhaite dire que la seule chose que le garçon apporte est son chien, on peut dire correctement : « *Le garçon apporte seulement son chien.* » et non : « *Le garçon seulement apporte son chien* » (Erreur plus commune en anglais).

Voir l'Article de référence (point 5.3.1) [2] pour plus d'exemples.

### 3.3.2. Aussi

*Aussi* souffre du même problème que *seulement* de part le fait qu'il est supposé se trouver immédiatement après le mot ou la phrase qu'il modifie, tandis que la plupart des gens le positionne de une ou deux manière standard en fonction de ce qui est *aussi*.

*Exemple* : « *Le garçon apporte aussi son chien.* » est compris comme, en plus de tout ce que le garçon amène, il amène son chien. Certains personnes écrivent, en voulant établir le même sens : « *Aussi le garçon apporte son chien* » (1) ou « *Le garçon aussi apporte son chien* » (2). (1) correspond au fait qu'en plus que toutes les autres personnes qui amènent leur chien, le garçon amène le sien. L'alternative (2), quant à elle, signifie qu'en plus que tout ce que le garçon fait (possiblement à son chien), il amène son chien.

Voir l'Article de référence (point 5.3.2) [2] pour plus d'exemples.

## 3.4. Ambiguïté structurelle

Chaque ambiguïté décrite au sein de cette section survient de l'incertitude émanant de la modification, par un mot ou une phrase donnée proche, du sens d'un mot.

### 3.4.1. Références pronominales

Certaines phrases peuvent s'avérer ambiguës de part la présence de pronoms. L'emploi de pronoms peut être dangereux lorsque la référence du pronom en question est problématique. En effet, si un pronom peut faire référence à deux mots précédemment cités, le sens de la phrase s'en avère modifié.

*Exemple* : « *Une cuisine a une porte reliant la cuisine à un couloir.* » ou « *Une cuisine a une porte la connectant à un couloir.* ». On voit, vis-à-vis de la première alternative (et également du contexte) que « *la* » correspond à lui cuisine. Cependant, « *la* » peut néanmoins faire référence à « *cuisine* » ou « *porte* ».

Voir l'Article de référence (point 5.4.2) [2] pour plus d'exemples.

### 3.4.2. Ceci (ce, cela) et idées complètes

Il s'agit là d'une erreur d'usage extrêmement populaire. La cause de cette ambiguïté est le *ceci* qui réfère à une idée complète plutôt qu'à un mot spécifique comme il le devrait. *Ceci* peut être employé comme un pronom ou il peut faire référence à un nom précédemment mentionné. Cependant, beaucoup de personnes utilisent *ceci* comme le premier mot d'une phrase référant non pas un mot précédemment mentionné mais à toute une idée capturée au sein des phrases précédentes.

*Exemple* : « *Une exigence doit être non ambiguë dans le sens que différents utilisateurs devraient donner la même interprétation à l'exigence. Cela a deux aspects.* »

Selon les règles grammaticales, « *cela* » réfère au sujet de la phrase précédente : « *une exigence* ». Cependant, une exigence ne possède pas les 2 aspects mentionnés. « *Cela* » réfère alors ici au concept entier définis dans la première phrase. Dès lors, une meilleure façon d'écrire

la phrase serait : « Ce concept d'interprétations multiples par différents lecteurs a deux aspects ».

Voir l'Article de référence (point 5.4.3) [2] pour plus d'exemples.

### 3.4.3. Autrement

Le but d'*autrement* est de spécifier ce qui est ou ce qui va arriver quand une condition précédemment mentionnée n'est pas rencontrée. S'il y a plus d'une condition décrivant la priorité d'*autrement*, il est très facile d'utiliser *autrement* d'une manière qui laisse le lecteur indécis quant à la condition sur laquelle l'*autrement* s'applique. Le lecteur peut douter quant à savoir si c'est la réponse alternative qui doit s'appliquer de part l'*autrement* ou si l'on n'est pas concerné par la situation. La portée d'*autrement* est ambiguë.

Exemple : Si C1, A1. Si C2, A2. Si C3, A3. Autrement, A4.

Voir l'Article de référence (point 5.4.4) [2] pour plus d'exemples.

### 3.4.4. Pas (Not)

Excepté pour une exception majeure, *pas* nie uniquement le mot qui le suit. L'exception (majeure) est le fait de nier le verbe d'une phrase qui engendre le fait de nier toute la phrase.

Exemple : « Il n'a pas amené son chien » correspond au cas où il n'a pas amené son chien.

Voir l'Article de référence (point 5.4.5) [2] pour plus d'exemples.

### 3.4.5. Pas (Not) et Parce que

Un cas particulier d'ambiguïté de possibilité (Voir 1.3.3.Ambiguïté de possibilité) survient quand un *parce que* est utilisé après une clause contenant une négation tel que *pas (not)*.

Voir l'Article de référence (point 5.4.6) [2] pour plus d'exemples (exemples s'appliquant uniquement à l'anglais).

### 3.4.6. Et et Ou dans la même phrase

En mathématique, il existe des règles de précédences qui gouvernent le sens d'une phrase avec plus d'un opérateur logique, *et* et *ou*. Le *et* établit un lien plus fort que le *ou* et est associatif à gauche. Si une association différente est désirée, des parenthèses sont utilisées pour établir le désir de manière claire. Le langage naturel n'a pas de règles de précédences ou d'associativité. Il n'est pas considéré non plus de bonne forme d'employer des parenthèses en langage naturel pour indiquer la précedence et l'associativité. Dans le cas d'une phrase avec seulement une sorte d'opérateur logique, il n'y a pas de problèmes car l'associativité du langage naturel commence à partir du début de la phrase.

Exemple : « Avec chaque entrée, vous avez un légume et une salade ou une soupe. » peut avoir deux sens : « Avec chaque entrée, vous avez (un légume et (une salade ou une soupe)) » ou « Avec chaque entrée, vous avez ((un légume et salade) ou une soupe).

Voir l'Article de référence (point 5.4.7) [2] pour plus d'exemples.



### 3.5. Parallélisme

Beaucoup de gens ne respectent pas le parallélisme quand il est requis. Par exemple, dans les phrases connectées par une conjonction, chacun de ces phrases devraient être au sein de la même partie du discours. En plus de ça, à l'intérieur de chacune des parties du discours chacune des phrases devraient avoir la même structure grammaticale.

#### 3.5.1. Parallélisme assumé

Si deux phrases présentes la même structure, ce parallélisme structurel implique un parallélisme sémantique. De ce fait le lecteur peut être contraint à croire qu'elles sont sémantiquement identiques. Cependant, deux phrases de structures identiques peuvent s'avérer différentes sémantiquement.

*Exemple* : « *Des chaussures doivent être portées. Les chiens doivent être transportés.* » vis-à-vis des escalators. Le parallélisme structurel implique la parallélisme sémantique.

Par contre, il n'y a pas de manière de voir si le parallélisme sémantique n'est pas prévu par le parallélisme structurel. Elles sont toutes les deux de la forme : « *X doit être Y* ». Dès lors, le lecteur pourrait induire qu'elles sont sémantiquement identiques. Cependant, sémantiquement elles sont différentes sur ce que le lecteur doit faire.

*Voir l'Article de référence (point 5.5.1) [2] pour plus d'exemples.*

#### 3.5.2. Que, Différent de et le parallélisme

Quand une comparaison est faite en utilisant *que*, il est essentiel que les deux côtés du *que* soit parallèle afin de permettre de déterminer de manière non ambiguë ce qui est comparé.

*Exemple* : « *Cleveland est plus proche de Philadelphie que (de) New York.* » peut être lue comme : « *La distance entre Cleveland et Philadelphie est plus petite que la distance entre Cleveland et New York.* ». Ce qui est vrai. Par contre, « *La distance entre Cleveland et Philadelphie n'est pas plus petite que la distance entre New York que Philadelphie.* »

Il est à noter que le cas précédent s'applique essentiellement à l'anglais.

*Voir l'Article de référence (point 5.5.2) [2] pour plus d'exemples.*

#### 3.5.3. Mots entre guillemets et Signification

Une autre source d'ambiguïté est la confusion entre un mot utilisé pour lui-même et le même mot utilisé pour fournir une signification. De manière générale, un mot est utilisé pour vouloir dire sa signification. Pour distinguer les deux utilisations d'un mot, l'utilisation d'un mot pour lui-même est normalement effectuée entre guillemets.

*Exemple* : Elle a dis "non"

Et pas : Elle a dis non.

*Voir l'Article de référence (point 5.6) [2] pour plus d'exemples.*

### **3.6. Expressions temporelles**

Les spécifications de temps, aussi bien les instants que les durées, sont particulièrement problématiques et implique des problèmes d'ouverture et de fermeture d'intervalles ainsi que des prépositions qui ne se traduisent pas bien entre les langages.

#### **Bibliographie**

- [1] Berry D.M. and Kamsties E., "From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity – A Handbook", November 2003.
- [2] <http://fr.wiktionary.org/wiki/homonyme> (Version last revised December 6, 2006) (Date of access 17/12/06)



## Annexe J

# Proposition de schéma conceptuel de la base de données *GenSpec*

### Description du schéma

Cette annexe propose un autre schéma conceptuel de la BD d'un projet *GenSpec*. Ce schéma a pour objectif de présenter les TE et relations jugées importantes. Aucune implémentation ne découle de ce schéma, celui-ci correspond à une nouvelle structure de la BD dans le cas d'une ré-ingénierie de *GenSpec*. En effet, il s'agit d'une restructuration totale de la BD existante. Son implémentation nécessiterait une ré-ingénierie totale du code de *GenSpec* afin qu'il puisse supporter les modifications apportées à la BD.

Deux éléments majeurs sont mis en évidence :

- la notion d'occurrence d'une exigence ;
- les relations entre les termes, les termes subjectifs et les synonymes.

### Occurrence d'une exigence

Un des problèmes actuel de *GenSpec* concerne la gestion des exigences synchronisées. Si une même exigence est utilisée à plusieurs endroits dans un projet *GenSpec*, elle est copiée dans la BD autant de fois qu'elle apparaît dans un projet, amenant une redondance importante de données. L'alternative proposée est d'introduire la notion d'occurrence d'une exigence. Toute utilisation d'une exigence dans un projet correspond à une occurrence. Les occurrences sont hiérarchisées, tout comme le sont actuellement les exigences dans *GenSpec*.

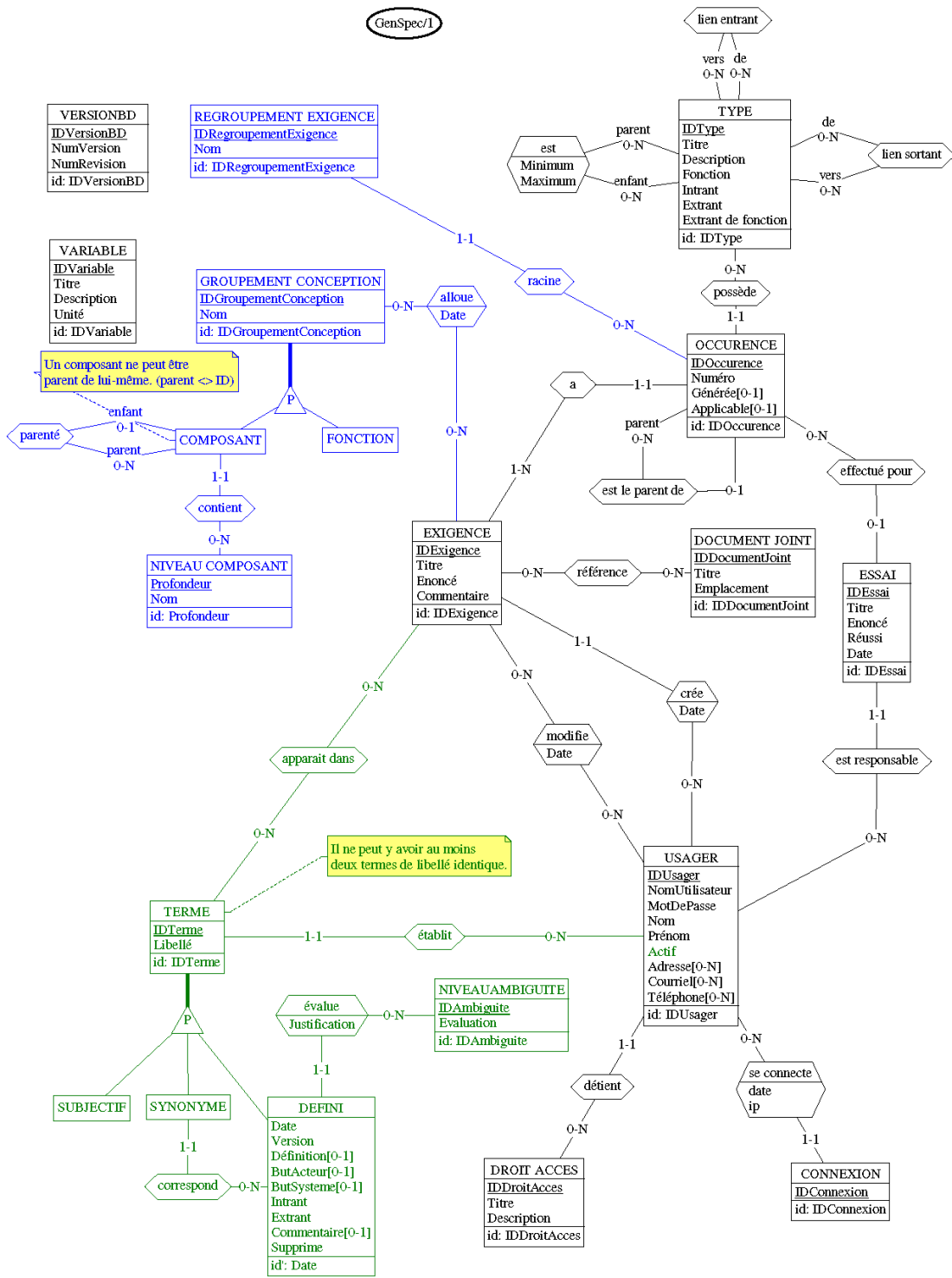
Chaque occurrence d'une exigence peut avoir un type différent en fonction de son emplacement dans la hiérarchie et des contraintes relatives au type d'exigences. Les essais sont réalisés vis-à-vis d'une occurrence et non pas d'une exigence et ce, car chaque occurrence peut avoir une procédure de résolution qui lui est propre. L'occurrence issue d'une exigence racine constitue l'occurrence racine d'un regroupement d'exigences. Notons également qu'un document joint peut être référencé par une exigence. Il s'agit de tout document susceptible d'apporter une quelconque forme d'information à l'exigence (traçabilité, digramme associé à l'exigence, etc.).

## Les relations avec le TE Terme

Un terme est soit :

- un terme subjectif;
- un terme défini dans le glossaire, c'est à dire un terme auquel des informations sont associées.
- un synonyme d'un terme défini.

Ces trois types de termes ont un libellé. Il ne peut y avoir au moins deux termes avec un libellé identique. Un terme défini peut posséder différents synonymes. Un terme défini peut être remplacé par un de ses synonymes. Le synonyme devient alors un terme défini et le terme défini peut devenir un synonyme, un terme subjectif ou être retiré du glossaire.





# Annexe K

## Article

La dernière modification du contenu de ce document date du 12 avril 2007.



# Ingénierie des exigences – Solution logicielle pour l’amélioration de la qualité et de la gestion des exigences

## Résumé

L'ingénierie des exigences (IE) est un ensemble d'activités très importantes relatives à tout processus de développement. Elle permet de faire le lien entre les besoins du client et leurs réalisations par le fournisseur. Dès lors, une gestion efficace de l'ensemble des exigences ainsi que la qualité des exigences définies sont des éléments majeurs dans la réussite d'un projet. Cet article a pour but de présenter une solution logicielle avancée permettant de répondre à ces besoins. Cette solution comprend un glossaire évolué et une administration centralisée des exigences, permettant respectivement de gérer l'ambiguïté lexicale des exigences, de supporter automatiquement tout type de liens entre toutes exigences d'un projet et d'allouer des exigences à des composants (sous-systèmes) établis par le concepteur. Cette solution logicielle a été intégrée à GenSpec, l'outil gratuit de support à l'IE développé à Hydro-Québec.

## Abstract

Requirements engineering (RE) is a very important set of activities carried out during the development process. RE links customers' needs and their realisation by the provider. An effective management of requirements and requirements quality are essential for a project to succeed. This article presents an advanced software solution answering these needs. This solution contains an evolved glossary and a centralised management of requirements. It allows dealing with lexical ambiguity in requirements, supporting automatically every kind of links between requirements and allocating requirements to designer made components (subsystems). The free RE tool GenSpec, developed at Hydro-Québec, supports this software solution.

## 1 Introduction

Cet article présente une solution logicielle améliorant la qualité et la gestion des exigences d'un projet.

Il présente de manière plus précise :

1. Le domaine de l'ingénierie des exigences (3.0) ;
2. Quelques problèmes majeurs en matière de qualité de rédaction et de gestion des exigences (4.0) ;
3. Un état de l'art des outils de support à l'ingénierie des exigences vis-à-vis de ces problèmes (5.0)
4. Deux solutions informatisées à ces problèmes : un glossaire et une gestion multi-documents d'exigences incluant l'allocation des exigences à des composants (6.0) ;
5. Les améliorations apportées par ces solutions (7.0).

Cet article s'adresse à toute personne concernée par l'établissement d'exigences vis-à-vis de tout type de produits ou services.

## 2 Domaine

L'ingénierie des exigences (IE) est un ensemble d'activités relatives à l'identification et à la communication du but d'un système d'informations, et au contexte dans lequel il sera utilisé. [7]. L'objectif de l'IE est de déterminer les services qu'un système doit offrir et les contraintes sous lesquelles il doit fonctionner [8]. L'IE se base sur les besoins du client pour produire des documents d'exigences tels que des normes, appels d'offres, contrats, devis, cahiers des charges ou spécifications. L'IE est souvent négligée, les attentes des clients concernés ne sont pas ou entièrement satisfaites. Les conséquences d'une telle négligence peuvent alors entraîner :

1. Une augmentation des coûts et délais de livraison ;
2. Une diminution de la qualité du système d'information. [5]

L'IE est la cause la plus fréquente d'échecs et la source de la plus grande quantité d'erreurs et ces erreurs sont les plus coûteuses. Dès lors, afin de répondre aux besoins des clients, il est essentiel, d'une part, de bien spécifier les exigences le plus tôt possible et avant tout nouveau développement et d'autre part, de maintenir à jour un document d'exigences complet. [8]

## 3 Problèmes

Cette section présente plusieurs problèmes rencontrés, vis-à-vis de la rédaction et la gestion des exigences en IE :

- 1) Ambiguïté des exigences ;
- 2) Incohérence des liens entre les exigences ;

3) Incohérence des allocations d'exigences à des composants.

### **3.1 Ambiguïté des exigences**

Lors de la rédaction d'exigences, l'emploi d'un langage naturel – tel que le français ou l'anglais – implique inévitablement différentes formes d'ambiguïté :

- a) Synonymie (3.1.1) ;
- b) Homonymie (3.1.2) ;
- c) Polysémie (3.1.3) ;
- d) Subjectivité (3.1.4).

REMARQUE – Nous traitons ici essentiellement l'ambiguïté lexicale. Pour obtenir d'avantage de détails relatifs aux différents types d'ambiguïté, consultez Berry D.M. and Kamsties [1].

#### **3.1.1 Synonymie**

La synonymie est « une relation entre deux termes ou deux expressions synonymes » [12]. Un synonyme est un terme ou une expression qui possède une signification très voisine voire identique à d'autres termes ou expressions [12].

L'emploi de synonymes dans des exigences est très fréquent car lié à l'utilisation d'un langage naturel. La référence à une certaine sémantique dans un document peut donc être faite par le biais de différents termes ou expressions. Cette utilisation non uniformisée peut alors entraîner confusions et erreurs lors de la lecture des exigences vu l'emploi de termes ou expressions différentes pour référer à une même signification.

EXEMPLE – *Document d'exigences* et *document de spécifications* sont synonymes [9].

#### **3.1.2 Homonymie**

L'homonymie survient lorsque deux termes possèdent la même représentation écrite et phonétique mais sont de sens et d'étymologies différentes [1]. "*Les homonymes sont des mots dont les différents sens ne sont pas reliés entre eux*" [6].

EXEMPLE – « Avocat » renvoie à la définition d'un fruit ou d'une profession. Il n'existe aucun lien sémantique apparent entre ces deux mots [6].

#### **3.1.3 Polysémie**

La polysémie survient lorsqu'un terme ou une expression possède plusieurs sens [12] apparentés mais seulement une seule étymologie [1]. Un terme peut être employé de différentes façons dans un document d'exigences. Cette utilisation multiple peut donc entraîner de nombreuses incompréhensions entre les différentes exigences d'un projet étant donné qu'un même terme peut supposer des sens distincts selon son utilisation.

EXEMPLE – « Louer » peut avoir le sens de « donner en location » et « vanter les mérites de quelqu'un ». [10]

#### **3.1.4 Subjectivité**

Un terme subjectif ou une expression subjective est un terme ou une expression dont la compréhension repose sur l'affectivité personnelle.

EXEMPLE – *Correct, efficace, rapide, complet* sont des termes subjectifs. Ces termes peuvent être interprétés de différentes manières selon l'affectivité du lecteur ou de l'auteur. En effet, la valeur que peut associer un acteur X à l'*efficacité* par rapport à un acteur Y peut entraîner une interprétation différente, en termes d'efficacité, d'une exigence.

### **3.2 Incohérence des liens entre les exigences**

Il est fréquent de retrouver de multiples liaisons entre des exigences. Ces liaisons établissent la traçabilité entre les exigences et permettent de gérer efficacement les changements. Les exigences peuvent appartenir, suite à une découpe effectuée par l'analyste, à des regroupements d'exigences différents, faisant partie de documents d'exigences différents. Un mécanisme de liaison permet alors de mettre en relation ces exigences entre elles.

En l'absence d'un tel mécanisme, lors d'une modification apportée à une exigence référencée, toutes ses références doivent être corrigées manuellement, entraînant une perte de temps conséquente. De plus, cette situation est susceptible d'engendrer de nombreuses incohérences par l'inexactitude des relations.

EXEMPLE – L'exigence E1 du projet P1 fait référence à l'exigence E2 du projet P2. Si E2 doit être supprimée, il faut alors manuellement supprimer le lien existant entre E1 et E2. Cette situation ne se produit pas si une liaison automatique existe entre ces deux exigences.

### **3.3 Incohérence des allocations d'exigences à des composants**

Premièrement, l'analyste organise les exigences du client. Celles-ci, bien que devant respecter un ensemble de contraintes, sont donc organisées de manière subjective en fonction des subdivisions de l'analyste.

Deuxièmement, le concepteur découpe le système en composants (ou sous-systèmes) pour réaliser la découpe architecturale du système. Cette découpe consiste à l'allocation des exigences, définies par l'analyste, en différents niveaux de composants.

La structure des exigences dépend donc de l'analyste et du concepteur. La répartition des exigences pour le concepteur, sans outil de support, peut être ardue et susceptible de générer des erreurs lorsque la tâche est réalisée manuellement.

## **4 État de l'art des outils de support à l'ingénierie des exigences**

Cette section présente les analyses préalables, relatives aux problèmes considérés ci-dessus, effectuées parmi des outils concurrents de support à l'IE. Les outils analysés sont :

- 1) Artisan Studio 6.1.21 (SysML) ;
- 2) IRqA 3.5.0 ;
- 3) REM 1.2.2 ;
- 4) Requisite Pro 7.0.0.0 ;
- 5) Telelogic Doors 8.1.

Les fonctionnalités qui entrent dans le cadre de la problématique décrite ci-dessus dans les différents outils sont les suivantes :

- 1) Ambiguïté des exigences (4.1) ;
- 2) Incohérence des liens entre les exigences (4.2) ;
- 3) Incohérence des allocations d'exigences à des composants (4.3).

L'analyse des fonctionnalités décrites ci-dessous s'est basée, d'une part, sur des tests réalisés dans chacun des outils cités, et d'autre part, sur l'analyse du site [<http://www.incose.org/>] qui présente les fonctionnalités intégrées dans de nombreux outils de support à l'IE [11]. Notons que ne sont mentionnés que les principes importants des outils en rapport avec les fonctionnalités rentrant dans le cadre de la problématique.

### **4.1 Ambiguïté des exigences**

**IRqA 3.5.0** présente un glossaire proposant de multiples fonctionnalités. De nombreuses informations, telles que l'auteur ou la date de modification, peuvent être recueillies pour un terme. Une gestion de l'historique des modifications d'un terme est également présente. Cela permet une gestion efficace de la traçabilité des termes utilisés dans les exigences.

**Requisite Pro 7.0.0.0** permet d'établir une discussion sur des termes définis. L'avantage que présente cette fonctionnalité est que plusieurs acteurs peuvent intervenir et débattre de l'exactitude du terme ou des informations – telle que sa définition – qui sont associées à ce terme.

**Telelogic Doors 8.1** permet, lors de la procédure de vérification orthographique des exigences, de proposer une « liste de mots alternatifs » à un terme défini préalablement et ce, afin de donner la possibilité à l'analyste de choisir parmi un de ces mots alternatifs proposés dans cette liste. Cette fonctionnalité se révèle intéressante afin de fixer un vocabulaire commun dans la définition des exigences.

EXEMPLE – Suggérer le terme « bassins » au lieu du terme « liste des groupes de charges par bassins » dans des exigences.

## 4.2 Incohérence des liens entre les exigences

**Telelogic Doors 8.1** fournit un mécanisme avancé en matière de traçabilité entre les exigences. Il permet de détecter des « liens suspects » avertissant que des exigences possédant un lien ont été modifiées. Ces exigences demandent dès lors une vérification de la part de l'analyste pour en assurer de nouveau la vraisemblance.

Tous les autres outils, y compris Telelogic Doors 8.1, fournissent une fonctionnalité relative à la traçabilité qui permet de mettre en évidence les relations des exigences entre elles, ou bien les relations qu'entretiennent les exigences avec d'autres éléments du système.

## 4.3 Incohérence des allocations d'exigences à des composants

**Artisan Studio 6.1.21** permet de mettre en relation des exigences avec n'importe quel élément défini, soit dans un modèle interne au projet, soit avec des éléments externes définis dans des applications extérieures, par exemple Telelogic Doors.

**Telelogic Doors 8.1** permet d'allouer des exigences à des éléments du système, tels que des composants. Les allocations peuvent se réaliser, soit à partir des exigences vers les composants, soit à partir des composants vers les exigences. Cette double approche propose beaucoup de souplesse à cette fonctionnalité.

## 5 Solutions

Cette section présente les solutions logicielles aux problèmes identifiés :

1. Glossaire (5.1) ;
2. Gestion multi-documents d'exigences (5.2) ;
3. Allocation d'exigences à des composants (5.3).

### 5.1 Glossaire

Les fonctionnalités principales du glossaire sont les suivantes :

- a) Obtention des informations relatives à un terme (5.1.1) ;
- b) Définition d'intrants et extrants (5.1.2) ;
- c) Gestion de la polysémie (5.1.3) ;
- d) Gestion de la synonymie (5.1.4) ;
- e) Gestion des raccourcis (5.1.5) ;
- f) Gestion des termes subjectifs (5.1.6).

**5.1.1 Obtention des informations relatives à un terme :** cette fonctionnalité permet de mémoriser les informations relatives à un terme dans le glossaire :

- a) *But pour l'acteur – personne ou autre système* : il correspond à l'utilité du terme pour tout acteur lié au projet en cours de spécification.
- b) *But pour le système* : il correspond à l'utilité du terme pour le système dans sa globalité.
- c) *Définition* : elle correspond à la sémantique du terme défini.
- d) *Niveau d'ambiguïté* : il correspond à l'évaluation de l'ambiguïté des informations relatives au terme.
- e) *Occurrences du terme* : elles correspondent à une liste d'exigences où le terme est employé.
- f) *Propriétés du terme* : elles établissent si le terme définis peut constituer ou non un intrant et/ou un extrant.
- g) *Liste des synonymes* : elle correspond à un ensemble de termes ou d'expressions de sémantique identique ou proche du terme défini.

La [Figure 1](#) présente une interface personne-machine reprenant l'ensemble des informations relatives à un terme.

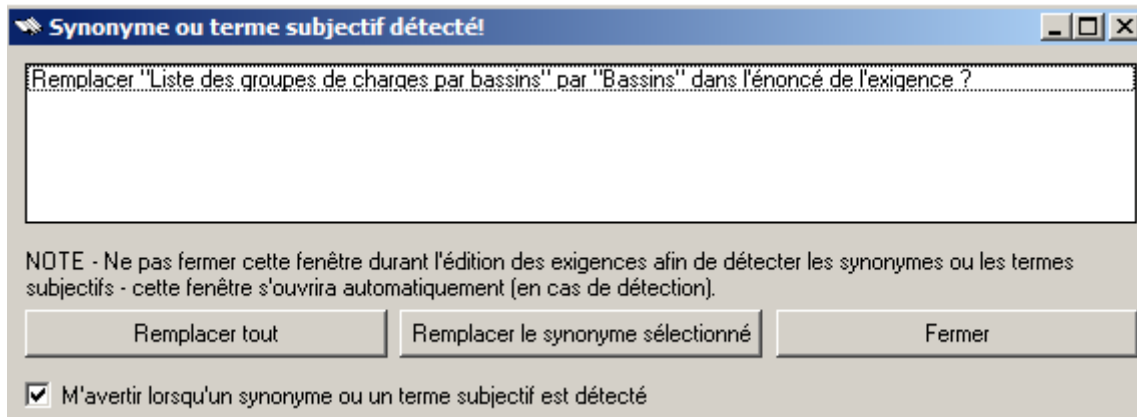
The screenshot shows a software window titled "Glossaire" with a close button in the top right corner. The window is divided into several sections:

- Libellé du terme :** A text box containing "Bassin".
- Version du terme :** A section containing "Auteur : olivier pire - (Administrateur)" and "Date de modification : 2007-01-03 15:35:19".
- Termes du glossaire :** A list box with "Bassin" selected.
- Propriétés du terme :** Two checkboxes, "Intrant" and "Extrant", both of which are unchecked.
- Identifiant du terme :** A text box containing "N° ID : 1".
- Définition :** A checkbox labeled "Implicite" which is checked, and a text area containing the text: "Ensemble de GC regroupés fonctionnellement ou régionalement, et ordonnés ou non selon les besoins."
- Le but pour l'acteur est de :** A text box followed by a checked "Implicite" checkbox.
- Le but pour le système est de :** A text box followed by a checked "Implicite" checkbox.
- Synonyme(s) | Ambiguïté | Occurences du terme | Commentaires :** A tabbed interface with the "Synonyme(s)" tab selected.
  - Aucun synonyme correspondant
  - Synonyme :** A text box.
  - Liste de synonymes :** A list box containing "Liste des groupes de charges par bassins".
  - Buttons: "Ajouter", "Modifier", "Supprimer", "Supprimer tout", and "Remplacer".
- Bottom bar:** Four buttons: "Nouveau", "Supprimer", "Enregistrer", and "Annuler".

Figure 1 - Fenêtre du glossaire

- 5.1.2 Définition d'intrants et extrants :** cette fonctionnalité permet de définir un terme, à partir d'un arbre hiérarchique des exigences présentant les relations de parenté entre ces dernières et ce, dans le glossaire. Le terme peut être réutilisé comme intrant ou extrant de toute exigence, offrant ainsi une uniformisation générale dans un projet.
- 5.1.3 Gestion de la polysémie :** cette fonctionnalité permet, de par l'utilisation du « cliquer-glisser » dans l'arbre hiérarchique des exigences, de gérer la polysémie. La polysémie est soumise à deux limitations :
- Une seule définition ne peut être associée qu'à un terme ;
  - Il ne peut exister plusieurs termes identiques dans le glossaire associés à des sémantiques différentes.

**5.1.4 Gestion de la synonymie :** cette fonctionnalité permet de limiter l'utilisation de synonymes dans un ensemble d'exigences. L'utilisateur qui définit un terme a la possibilité d'associer un ensemble de synonymes au terme défini. L'emploi d'un de ces synonymes lors de la rédaction d'une exigence sera marqué préalablement par la possibilité de remplacer le synonyme par son terme défini. La [Figure 2](#) présente une interface personne-machine permettant le remplacement d'un synonyme par son terme associé.



**Figure 2 - Gestion de la synonymie**

**5.1.5 Gestion des raccourcis :** cette fonctionnalité permet d'accroître la vitesse de rédaction des exigences. Un raccourci constitue une abréviation représentative d'un terme. Un raccourci peut être associé à un terme de par la fonction de synonyme. En ajoutant un raccourci comme synonyme d'un terme, cela permet, lors de la rédaction des exigences, d'employer et de remplacer ce raccourci, comme tout synonyme, par son terme associé.

**5.1.6 Gestion des termes subjectifs :** cette fonctionnalité permet d'éviter l'utilisation de termes subjectifs lors de la rédaction d'exigences. L'utilisateur est averti du caractère ambigu de l'exigence qu'il rédige lorsqu'il emploie un terme subjectif dans l'exigence.

## **5.2 Gestion multi-documents d'exigences**

La gestion multi-documents d'exigences permet à un utilisateur de créer un arbre hiérarchique d'exigences regroupant la totalité des exigences d'un projet. Chaque exigence racine de la structure d'arbre constitue un nouveau regroupement d'exigences. L'ensemble des exigences d'un projet, même décomposé en de multiples regroupements d'exigences, peut alors se retrouver dans un seul et même projet.

Cette fonctionnalité permet de répondre aux besoins suivants :

- a) Traçabilité entre des exigences de regroupements différentes (5.2.1) ;
- b) Génération de multiples documents d'exigences (5.2.2).

### **5.2.1 Traçabilité entre des exigences de regroupements différents**

Les exigences des différents regroupements peuvent être liées entre elles, quel que soit le regroupement dans lequel elles apparaissent. La possibilité de relier les exigences entre elles montre le type de relation qu'elles entretiennent. Le nombre de liens entre exigences à spécifier manuellement est donc fortement réduit. La gestion des exigences peut être réalisée plus rapidement et le risque d'erreurs se voit par conséquent amplement diminué.

### **5.2.2 Génération de multiples documents d'exigences**

Différentes possibilités relatives à la génération de documents sont offertes à l'analyste. Ces possibilités sont les suivantes :

- 1) La génération de toutes les exigences d'un projet dans un document unique ;
- 2) La génération de documents distincts présentant des regroupements d'exigences différents.

La génération de documents - spécification, arbre hiérarchique, etc. -, vis-à-vis de différents regroupements d'exigences peut être réalisée en une seule opération. Cela améliore donc la souplesse et la rapidité de la

procédure de génération de documents d'exigences. La Figure 3 présente une interface personne-machine : dans la partie supérieure, on retrouve les différents regroupements d'exigences ainsi que les différents types de regroupements qu'il est possible de générer ; dans la partie inférieure, on retrouve l'évolution de la génération par document.

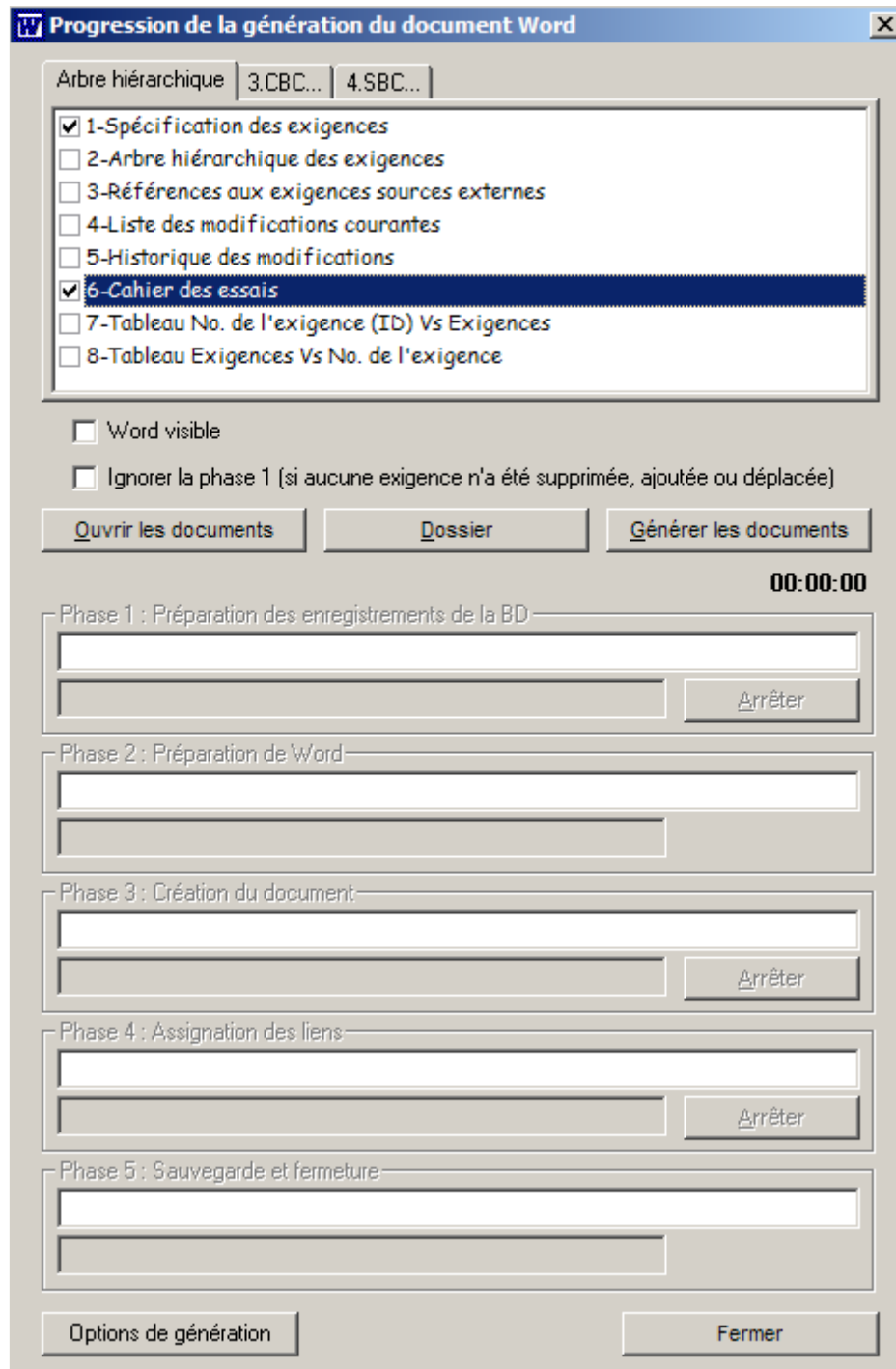


Figure 3 - Génération de documents d'exigences

### 5.3 Allocation d'exigences à des fonctions et composants

Les exigences définies par l'analyste peuvent être allouées par le concepteur à des composants qu'il a préalablement définis. Un tableau de conception peut alors être généré dans un document Microsoft Excel. Ce dernier présente les allocations d'exigences réalisées dans un format tabulaire et ce, afin de mettre en évidence les allocations d'exigences selon les composants définis.

Cette fonctionnalité permet d'étendre les activités de l'IE pour le concepteur du système. La Figure 4 présente une interface personne-machine : à gauche, on remarque la hiérarchie de composants établis par le concepteur ; à droite, apparaît la liste des allocations des exigences au composant sélectionné.

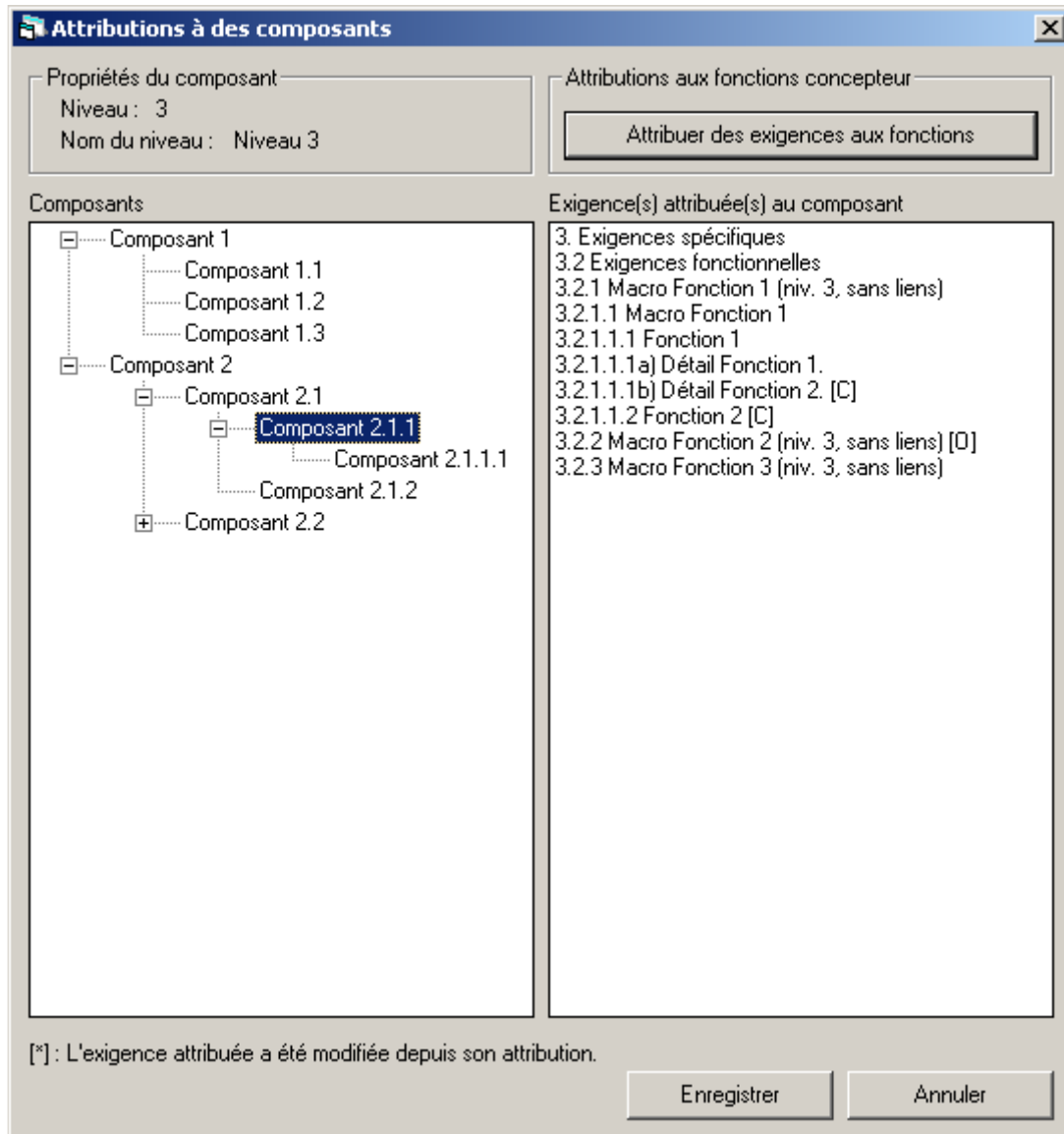


Figure 4 - Allocations d'exigences à des composants

## 6 Améliorations

Cette section présente les améliorations, classées selon l'ordre des solutions décrites ci-dessus, suite à l'apport de ces nouvelles fonctionnalités :

- 1) Compréhension uniformisée (6.1) ;
- 2) Diminution de l'ambiguïté lexicale des exigences (6.2) ;
- 3) Rédaction plus rapide (6.3) ;
- 4) Utilisations d'un terme (6.4) ;
- 5) Gestion des exigences entre regroupements d'exigences (6.5) ;
- 6) Génération de regroupements d'exigences (6.6) ;
- 7) Allocation d'exigences pour la phase de conception (6.7).

### 6.1 Compréhension uniformisée

Le glossaire permet d'établir une sémantique pour chaque terme. Une compréhension uniformisée du terme, et donc une diminution de l'ambiguïté, en résulte pour l'ensemble des acteurs d'un projet.



## 6.2 Diminution de l'ambiguïté lexicale des exigences

Le glossaire permet de réduire l'ambiguïté lexicale des exigences :

- a) Chaque terme, défini dans le glossaire, est associé à une seule sémantique ;
- b) Chaque sémantique, définie dans le glossaire, est associé à un seul terme ou une seule expression.

## 6.3 Rédaction plus rapide

Le glossaire permet d'accroître la vitesse de rédaction des exigences par l'utilisation de raccourcis. L'utilisation d'intrants et d'extrants définis préalablement dans le glossaire permet une réutilisation multiple, donc plus rapide, de ces derniers.

## 6.4 Utilisations d'un terme

Le glossaire permet de lister, pour chaque terme, les exigences dans lesquels ce terme apparaît. Cela fournit à l'utilisateur une vue d'ensemble quant à l'utilisation, l'importance et l'impact d'un terme dans un projet.

## 6.5 Gestion des exigences entre regroupements d'exigences

La possibilité de créer plusieurs regroupements d'exigences dans un même projet permet une gestion plus complète et plus efficace de l'ensemble des exigences. Les liens entre les différents regroupements d'exigences étant réalisés automatiquement. Cette amélioration diminue ainsi le risque d'incohérences entre les exigences.

## 6.6 Génération de regroupements d'exigences

La gestion multi-documents d'exigences fournit la possibilité à l'utilisateur de générer, en une seule fois, les regroupements d'exigences qu'il souhaite et ce, qu'il s'agisse de regroupements différents ou de documents présentant la totalité des exigences.

## 6.7 Allocations d'exigences pour la phase de conception

L'allocation d'exigences à des composants permet au concepteur d'utiliser un outil afin de supporter la découpe architecturale du système. Le concepteur réalise ses découpes selon ses besoins à partir des exigences définies préalablement par l'analyste.

## 7 Conclusion

La production d'exigences de qualité est essentielle dans la réussite d'un projet. Les outils de support à l'IE, tel que «GenSpec», fournissent un support pour l'ensemble de ces activités. L'ajout d'un glossaire, d'une gestion multi-documents d'exigences et d'un mécanisme d'allocations d'exigences à des sous-systèmes, permettent respectivement de répondre aux problèmes de l'ambiguïté lexicale, des liens entre les exigences et de la conception architecturale du système. Il en découle une gestion plus efficace des exigences ainsi qu'une rigueur accrue vis-à-vis de la rédaction des exigences. L'outil de support à l'IE GenSpec apporte les solutions à tous les problèmes identifiés (Tableau 1) [4].

Grâce au support que propose GenSpec pour répondre aux différents problèmes énoncés dans cet article, les besoins des clients peuvent alors être exprimés plus rigoureusement, augmentant ainsi les chances de répondre aux souhaits des clients efficacement et dans un délai adapté.

Tableau 1 - Liens problème-solution

Problème	Solution
3.1.1	5.1.1 ; 5.1.4 ; 5.1.5
3.1.2	5.1.1; 5.1.2 ; 5.1.3
3.1.4	5.1.6
3.2	5.2
3.3	5.3

## 8 Documents de référence

- [1] Berry D.M. and Kamsties E., « From contact drafting to Software Specification: Linguistic Sources of Ambiguity – A HandBook », November 2003.
- [2] Collection Microsoft Encarta 2004 DVD – Dictionnaire de la langue française, 2004
- [3] Bujold R. « Aide pour GenSpec », (Version 6.4.22 last revised 29/12/06) (Date of access 03/01/07)

- [4] Bujold R. , « Ingénierie des exigences – L’outil de support GenSpec », *IEEE Canadian Review*, pages 13-16, Fall 2005. ([http://iee.ca/canrev/cr51/CR51\\_GenSpec.pdf](http://iee.ca/canrev/cr51/CR51_GenSpec.pdf))
- [5] Bujold R. , « Ingénierie des exigences – Une méthode simple et systématique », *IEEE Canadian Review* , pages 6-10, Fall 2004. (<http://iee.ca/canrev/cr48/cr48.pdf>)
- [6] Deville G., « Introduction aux méthodes et concepts essentiels d’ingénierie linguistique», Technical report, University of Namur, Belgium, 2006-2007.
- [7] Easterbrook S., « Requirements Engineering», page 8, Fall 2006 (<http://www.cs.toronto.edu/~sme/CSC340F/readings/FoRE-chapter01-v7.pdf>)
- [8] Habra N., « Ingénierie du logiciel – Ingénierie des exigences», Technical report, University of Namur, Belgium, 2006-2007.
- [9] [http://fr.wiktionary.org/wiki/Aide:Synonymes et antonymes](http://fr.wiktionary.org/wiki/Aide:Synonymes_et_antonymes), (last revised 18/10/06) (Date of access 29/12/06).
- [10] <http://www.etudes-litteraires.com/figures-de-styles/polysemie.php>, (Date of access 03/01/07).
- [11] <http://www.paper-review.com/tools/rms/read.php> (Date of access 19/01/07).
- [12] Le nouveau Petit Robert, Montréal Canada, 1993

## 9 Remerciements

Heymans Patrick, Lacroix Marc, Levesque Claire, Pirmez Olivia, Ratté Harold, Sicard Alain, Vincelette Michel et le reste de l’unité Conception-Automatismes à Hydro-Québec.

## 10 A propos des auteurs

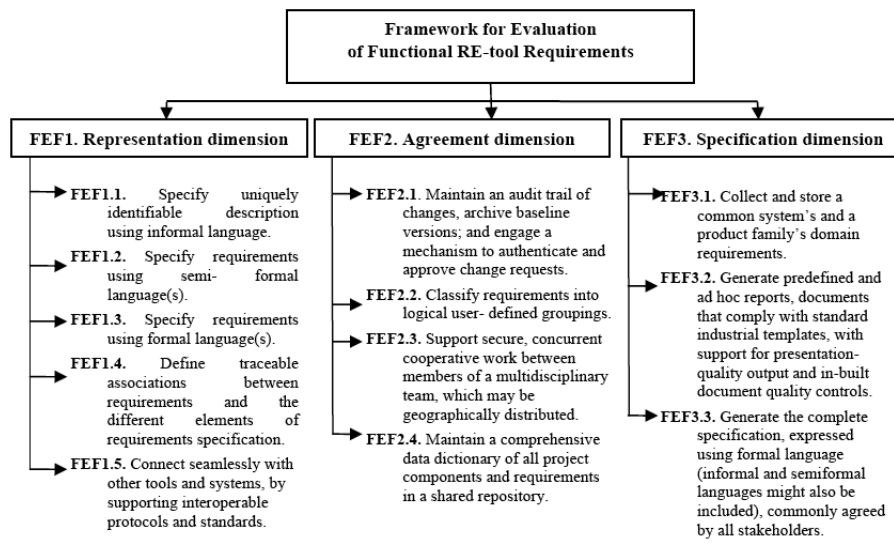
Nicolas Pirmez et Olivier Pire sont étudiants en troisième maîtrise en informatique aux facultés universitaires Notre-Dame de la Paix à Namur, Belgique. Dans le cadre de leurs études, leur stage à Hydro-Québec s’est porté sur l’ajout de nouvelles fonctionnalités à l’outil de support à l’ingénierie des exigences développé à Hydro-Québec : GenSpec.



## Annexe L

# Framework for functional and non-functional RE-tool requirements

Cette annexe reprend l'ensemble des tableaux définissant les différentes exigences fonctionnelles et non-fonctionnelles du cadre d'évaluation appliqué à l'OSIE *GenSpec*. L'ensemble de ces tableaux sont tirés de [Mat05].



	Features	Activities How does the RE-tool...	Activity description
Representation dimension	FEF1.1. Specify uniquely identifiable description using informal language.	FEF 1.1.1 provide natural language description.	<i>Natural language</i> is the language in which humans communicate in everyday life. The requirements could also be defined using <i>professional language</i> (Pohl, 1994, Krogstie and Solvberg, 1996), which is used by a set of persons working in a certain kind of area.
		FEF 1.1.2 allow specifying unique identification (ID) for each separate requirement.	Requirements ID is assigned when requirement is created and entered to a requirements database. It should be unique for each individual requirement.
		FEF 1.1.3 allow importing of requirements and their description from text document.	The initial user needs are usually stored in requirements documents that are text files. The tool should have means to import this information to the tool database, using semi or fully automated functionality.
	FEF1.2. Specify requirements using semi-formal language(s).	FEF 1.2.1 provide tools for semiformal language description.	<i>Semiformal language</i> (Pohl, 1994) is language with a precisely defined vocabulary and syntax, but without a precisely defined semantics (e.g., ER-diagrams, OMT, UML, and DFD). The particular languages maintained in the RE-tool, are specified using non-functional process requirements.
		FEF 1.2.2 provide forward/backward traceability between semiformal, and informal, formal descriptions.	While having requirements definition in different representation languages, it is important to keep requirements uniqueness and maintain traceability relationships between different representation forms.
	FEF 1.3. Specify requirements using formal language(s).	FEF 1.3.1 provide tools for formal language description.	<i>Formal language</i> (Pohl, 1994) is a language with a precisely defined vocabulary, syntax and semantics (e.g., Z-schemas, beta-notations, and aggregation models). The particular languages maintained in the RE-tool, are specified using non-functional process requirements.
		FEF 1.3.2 provide forward/backward traceability between formal and informal, semiformal descriptions.	While having requirements definition in different representation languages, it is important to keep requirements uniqueness and maintain traceability relationships between different representation forms.
	FEF 1.4. Define traceable associations between requirements and the different elements of requirements specification.	FEF 1.4.1 provide functions for testing traceability between informal, semiformal and formal requirement description.	By defining traceability between different representation forms, it is important to ensure that the requirements model is not changed using different views. The activity is partially duplicated with FEF1.2.2 and FEF1.3.2. However it is important to have it here as the RE-tool evaluation purposes could differ in different environments and previous activities would not be chosen for assessment.
		FEF 1.4.2 create parent-child traceable relations between requirements.	The activity defines hierarchical parent-child traceability. Parent here is understood like a higher abstraction level requirement, which consists of lower abstraction level requirements – children.
		FEF 1.4.3 maintain peer-to-peer traceable relations between requirements.	Peer-to-peer relationship defines requirements trace on the same hierarchical level (on the same abstraction level).

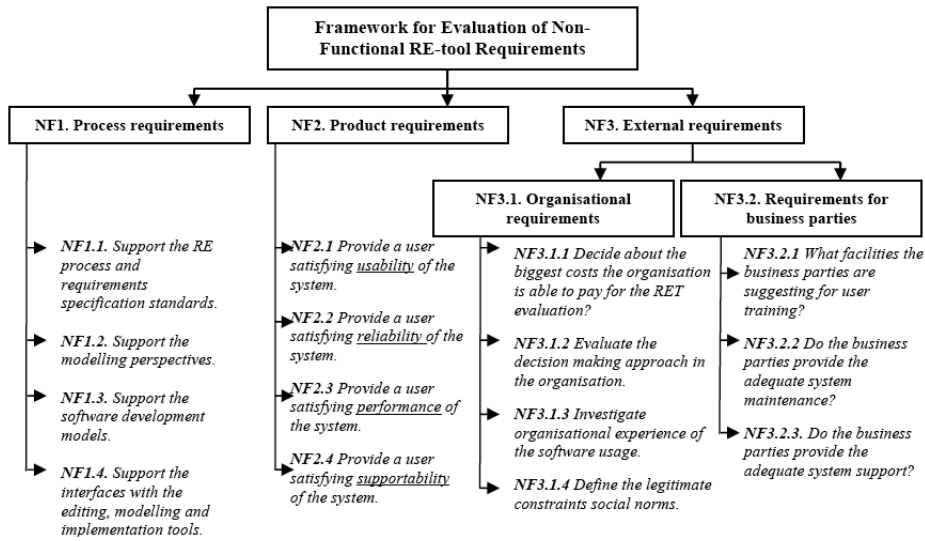
<i>Continuation of Table 5.1</i>		
	<b>FEF 1.4.4</b> maintain traceable relation between various related information.	Related information could be characterised as additional requirements views, figures, requirements documents, elicitation, or negotiation material.
	<b>FEF 1.4.5</b> maintain forward/backward traceability between a source of requirements, the requirements and design.	Requirements source is understood as the material (document, related system or person), from where (or whom) requirements are elicited. Design here is the system description, which specifies how the system should be built and implemented.
<b>FEF 1.5.</b> Connect seamlessly with other tools and systems, by supporting interoperable protocols and standards.	<b>FEF 1.5.1</b> allow importing/exporting requirements description from/to text documents.	Importing of textual information from text and/or graphical documents eases requirements model construction. Exporting of textual and/or graphical requirements information helps to analyse them using other software tools. Both import and export functionality could be fully or semi-automated. The activity includes analysis of the interoperability relationships and associations between the RE-tool and other tools used in an organisation.
	<b>FEF 1.5.2</b> allow importing/exporting requirements description from/to graphical documents.	

	Features	Activities How does the RE-tool...	Activity description
Agreement dimension	<b>FEF 2.1.</b> Maintain an audit trail of changes, archive baseline versions; and engage a mechanism to authenticate and approve change requests.	<b>FEF 2.1.1</b> maintain user authentication to the system (i.e. user name, password).	RE-tool users are stakeholders of the RE process. When speaking about the product produced with the RE-tool, users consist of different groups: those who are responsible for product development, introduction and maintenance; those with financial interest, responsible for the sale or purchase; and those who have an interest in product use.
		<b>FEF 2.1.2</b> allow grouping users into different groups.	The RE-tool has to allow definition of user groups, and assignment the users to them.
		<b>FEF 2.1.3</b> allow creating different views (according to documents, requirements, attributes) for different groups of stakeholders.	Different user groups could have different interest in requirements activities. The tool should allow display of various requirements views and tool functionality according to the user group or individual user needs.
		<b>FEF 2.1.4</b> register agreement/ rationale/ discussion/ negotiation/ changes/ history of requirements and by how it was achieved.	<i>Rationale</i> is a reference to a set of information which provides an explanation why the requirement has been included. The negotiation, discussion and history maintenance are the means to create and to maintain the requirements rationale.
		<b>FEF 2.1.5</b> call the earlier requirement description/ versions and register them into history context.	<i>History</i> is chronologically ordered states of individual requirements or requirements model. The RE-tool has to support history control mechanism and keep change track. The earlier requirements model (or requirement) versions should be possible to upload.
	<b>FEF 2.2.</b> Classify requirements into logical user-defined groupings.	<b>FEF 2.2.1</b> allow specifying attributes/ properties of the requirement.	Definition of requirements attributes and properties include requirements prioritisation activities, where different requirements or their groups are characterised according to the importance, stability, time to market and similar characteristics.
		<b>FEF 2.2.2</b> provide sorting according to different attributes/ properties.	Requirements sorting and filtering according to defined attributes and properties, helps to narrow analysis scope and to find the relevant requirements to a user.
		<b>FEF 2.2.3</b> provide filtering according to different attributes/ properties.	
	<b>FEF 2.3.</b> Support secure, concurrent cooperative work between members of a multidisciplinary team, which may be geographically distributed.	<b>FEF 2.3.1</b> provide platform independent interface for geographically distributed users.	Platform independent interface suggests means for cooperative geographically distributed teams to access the requirements model through Web browsers.
		<b>FEF 2.3.2</b> allow making a copy for modification of an already approved version of requirements description in different abstract levels (document, requirement).	Modification of requirements model versions leads to working on the same requirements data at the same time. Approval of requirements helps to reach agreement about the requirements model. The requirements model is the object of negotiation and agreement about the requirements, and their changes.
		<b>FEF 2.3.3</b> provide a change approval cycle for multiple change negotiation and approval before posting into common repository.	The RE-tool should ensure that different versions of the requirements model would not overlap but be artefacts to reach the agreement.

<i>Continuation of Table 5.2.</i>		
FEF 2.4. Maintain a comprehensive data dictionary of all project components and requirements in a shared repository.	FEF 2.4.1 provide the single repository or data and concept dictionary.	Data dictionary provides definitions of the main elements of the requirements model, RE-tool functionality, and related information.
	FEF 2.4.2 provide separate data dictionaries for non-technical users and technical users.	The RE-tool is used by users who have knowledge about the RE and software development process, and users who do not have such a knowledge. The RE-tool has to distinguish between both groups of users providing appropriate data dictionaries or suggesting process scenarios.
	FEF 2.4.3 provide the help system to the users.	The RE-tool has to provide help functionality, where all the RE-tool functions and features would be defined and explained.

	Features	Activities How does the RE-tool...	Activity description
Specification dimension	FEF 3.1. Collect and store a common system's and a product family's domain requirements.	FEF 3.1.1 enable selection and extraction of common domain requirements and requirements which differentiate systems in product line.	The activity emphasises the requirements reuse from the requirements repository, which is maintained by the RE-tool. Requirements from the repository could be selected using, for instance, free-based or/and discriminant-based methods (Mannion <i>et al.</i> , 1999; Kaindl and Mannion, 2005)
		FEF 3.1.2 incorporate requirements to a concrete project.	Incorporation of requirements (and their groups) into appropriate places of the requirements model helps to create the requirements specification faster as these requirements are complete and agreed in similar projects or domains (or project).
		FEF 3.1.3 adapt/ spread changes in domain requirements to concrete projects within domain.	When the requirements are changing in the requirements repository, the changes are spread to all related requirements within the domain.
		FEF 3.1.4 provide comparison of domain requirements feasibility.	The RE-tool has to support functionality for requirements analyses in the requirements repository by comparing, changing and storing them here.
	FEF 3.2. Generate predefined and ad hoc reports, documents that comply with standard industrial templates, with support for presentation-quality output and in-built document quality controls.	FEF 3.2.1 provide wizards for report generation.	Wizards help to prepare reports, which are not designed together with the RE-tool functionality.
		FEF 3.2.2 provide possibility to print report according to views and sorting.	The RE-tool has to maintain requirements sorting, filtering, prioritisation requirements agreement, negotiation and rationale maintenance reports which would be possible to view, send by e-mail, and print them out.
		FEF 3.2.3 provide possibility to print results of rationale, brainstorm and etc.	
		FEF 3.2.4 provide techniques for error checking.	All the reports have to be syntactically correct. The RE-tool should have functionality for error prevention, detection and correction.
	FEF 3.3. Generate the complete specification, expressed using formal language (informal and semiformal languages might also be included), commonly agreed by all stakeholders.	FEF 3.3.1 correspond to standards of software documentation.	The requirements specification should correspond to the international standards (the exact standard maintained by the RE-tool, is determined by the non-functional process requirements).
		FEF 3.3.2 correspond to standards, defined by an organisation.	The RE-tool should have means to define internal organisational standards for requirements specification. The internal requirements specification standards could be adapted following the international ones, or created according the organisational needs.
		FEF 3.3.3 support formal languages for complete, commonly agreed requirements specification.	The output of the RE process is an agreed, complete, represented in formal language requirements specification. The RE-tool has to ensure functionality for completeness, agreement and formality check.





	Feature RE-tool should...	Working process characteristics
Process requirements	NF1.1. Support the selected RE and requirements specification standards.	IEEE std 830-1998, ISO 9126, NASA-DID-P200, Internal organisational standards.
	NF1.2. Support the selected modelling perspectives.	Structural, functional, behavioural, rule, object, communication, actor and role.
	NF1.3. Support the software development models.	Waterfall, spiral, prototyping, transformational, knowledge-based, domain-based, and RUP.
	NF1.4. Support the interfaces with the editing, modelling and implementation tools.	Editing, communication, modelling, implementation, testing tools, database management systems.

	Feature	Measures	Questions
Product requirements	NF2.1. Provide a user satisfying <u>usability</u> of the system.	Capability of the system to be: <ul style="list-style-type: none"> <li>- understood;</li> <li>- learned;</li> <li>- used and liked by the user;</li> <li>- used under specified conditions.</li> </ul>	How easy is the RE-tool to learn for various groups of users? How efficient is the RE-tool for the frequent user? How easy is the RE-tool to remember for the occasional user? How satisfied is the user with the RE-tool? How easy is it to understand what the RE-tool does?
	NF2.2. Provide a user satisfying <u>reliability</u> of the system.	Accuracy Error tolerance; Consistency; Recoverability; Availability.	What failure types are evaluated for the RE-tool? What is the availability of the RE-tool and its components? How long is the RE-tool allowed to be out of operation after it has failed?
	NF2.3. Provide a user satisfying <u>performance</u> of the system.	System performance; Required amount of resources; Data and result accuracy.	What is the minimum and average response time for certain data transaction or operation? What is minimum and maximum throughput for certain operations? What is the capacity of the RE-tool? What is resource utilization of the RE-tool? Do users receive the exact output?
	NF2.4. Provide a user satisfying <u>supportability</u> of the system.	Correctability Extensibility	What type (e.g., corrective, adaptive, preventive, perfective) of maintainability is performed most often? What are reasons for the maintainability activities? How many steps (activities, corrections) should be performed to reach the desirable state of the tool?

	Features	Measures	Description
Organisational requirements	NF3.1.1. Decide about the biggest costs the organisation is able to pay for the RE-tool evaluation?	Tool prices	Direct tool price, tool running costs (like new computers, RAM, etc.)
		Evaluation costs	Evaluation and transition costs (changing from the old to a new working practice), time and resources need to evaluate tool.
		Maintenance and support costs	Future operational costs, support costs, maintenance/ upgrades etc.
	NF3.1.2. Evaluate the decision making approach in the organisation.	Objectivistic (management requirements);	An objectivistic world-view describes an organisation, where stakeholders can map the required RE-tool without changing the organisational processes.
		Constructivistic (users' requirements)	A constructivistic approach supports stakeholder knowledge externalisation and internalisation processes, and in such a way makes the organisational environment part of the individual understanding. The RE-tool selection is the process of negotiation of the environment and RE-tool suitability to this environment.
	NF3.1.3. Investigate organisational experience of the software usage.	Degree of the RE-tool requirements	Different user experience contributes with different degree of abstraction for the tool functional and non-functional feature evaluation. More experienced users could require evaluating bigger number of requirements or only the particular requirements, which would contribute with evaluation of 'needed' tool features.
		Degree of RE-tool evaluation needed	Different user experience contributes with different degree of abstraction for the tool functional and non-functional feature evaluation. More experienced users could require the more detailed tool analysis of a particular feature.
	NF3.1.4. Define the legitimate constraints social norms.	Political constraints and laws	Political constraints and laws specify the norms, which exist in the organisation, and the country where organisation is situated.
		Cultural constraints	Cultural constraints describe the national traditions, customers, which could effect tool evaluation and usage.
		Social constraints	Social constraints characterise the organisational workers relationships, organisational workers hierarchy and relationships between management and workers.

	Features	Measures	Description
Requirements for business	NF3.2.1. What facilities the business parties are suggesting for user training?	Availability (costs) of training seminars;	Describes availability of tool training seminars organised by the tool vendor.
		Time spent on the phone before getting an answer; Average response time for online help; Average response time for help via e-mail;	Does the vendor or consultant provide phone, e-mail and online consultancy services using messenger systems (e.g.: MSN, ICQ and Skype). How long does it take to get answers about the tool usability and functionality?
		Availability of information in the vendors' Web page.	Does the vendor of the tool provide the appropriate support in the tool Web page? Is the information informative and relative to the emerging question? How often does the vendor update the Web information?
	NF3.2.2. Do the business parties provide the adequate system maintenance?	Quality of the maintainability services;	The basic supportability (maintainability) metrics: ratio of total change implementation time to total; number of changes implemented; number of unresolved problems; time spent on unresolved problems; percentage of changes that introduce new faults; number of modules modified to implement a change.
		Availability at any time;	Is vendor available at any time when maintainability problems of the tool emerge? How long does it take to get service form the vendor?
	NF3.2.3. Do the business parties provide the adequate system support?	Quality of support during new version releases;	Does vendor representative participate during tool updates? Does the vendor provide teaching services of new tool versions?
		Quality of updates of the current tool version	Do updates solve the old version problems? Are new tool versions more effective, easy to use, etc. than older ones?