



HAL
open science

Approche système pour la conception d'une méthodologie pour l'élicitation collaborative des exigences

Jacqueline Konaté

► **To cite this version:**

Jacqueline Konaté. Approche système pour la conception d'une méthodologie pour l'élicitation collaborative des exigences. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2009. Français. NNT: . tel-00435878

HAL Id: tel-00435878

<https://theses.hal.science/tel-00435878>

Submitted on 25 Nov 2009

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

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

APPROCHE SYSTÈME POUR LA CONCEPTION D'UNE
MÉTHODOLOGIE POUR L'ÉLICITATION COLLABORATIVE
DES EXIGENCES

PAR
JACQUELINE KONATÉ

THESE PREPAREE AU LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTÈMES
EN VUE DE L'OBTENTION DU DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE
UNIVERSITÉ TOULOUSE III PAUL SABATIER

SPECIALITÉ SYSTÈMES INDUSTRIELS

Jury :

<i>Rapporteurs :</i>	Jean-Pierre BOUREY	-	Ecole Centrale de Lille (LGIL)
	Zineb SIMEU-ABAZI	-	Institut Polytechnique de Grenoble (G-SCOP)
<i>Directeur :</i>	Abd-El-Kader SAHRAOUI	-	Université de Toulouse (LAAS)
<i>Président :</i>	Jean-Pierre BOUREY	-	Ecole Centrale de Lille (LGIL)
<i>Examineurs :</i>	Mario PALUDETTO	-	Université de Toulouse
	Pascale ZARATÉ	-	Institut Polytechnique de Toulouse (IRIT)
	Gwendolyn KOLFSCHOTEN	-	Technology University of Delft (TBM), Pays-Bas

À El Shaddai.

*Comment rendrai-je à l'Éternel tous ses bienfaits envers moi ?
J'élèverai la coupe des délivrances, et j'invoquerai le nom de l'Éternel.
J'accomplirai mes vœux envers l'Éternel, en présence de tout son
peuple.*

PS116 :12-14.

Avant-propos

Ce mémoire synthétise les trois années de recherche effectuées au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) et pour une part à la faculté TPM (Technology Policy and Management) de l'Université Technologique de Delft aux Pays-bas.

Ces travaux sont le résultat d'efforts conduits à la fois par l'équipe de recherche "Ingénierie Système et Intégration" du LAAS dirigée par DEMMOU Hamid et le groupe de recherche en Ingénierie Système de l'Université Technologique de Delft dirigé par VERBRAECK Alexander.

Cette section s'adresse aux personnes ayant contribué au bon déroulement des travaux réalisés dans le cadre de la thèse d'un point de vue technique, mais également d'un point de vue personnel et social.

Mes remerciements vont tout d'abord à l'endroit de Monsieur SAHRAOUI Abd-El-Kader qui a accepté de diriger cette thèse de recherche, et de l'avoir fait avec soin car il n'a ménagé aucun effort pour être là quand il le fallait. Je le remercie beaucoup pour sa disponibilité et l'encadrement dont j'ai bénéficiés auprès de lui.

Mes remerciements vont également à l'endroit de Mademoiselle KOLFSCHOTEN Gwendolyn qui a encadré mes travaux à l'Université de Delft et qui m'a beaucoup aidé dans l'apprentissage de l'approche de l'Ingénierie de la Collaboration et l'intégration de cette approche dans mes travaux. Je la remercie pour sa sollicitude et sa disponibilité qui ont été d'un grand apport pour l'avancement de mes travaux.

Mes remerciements concernent ensuite plus particulièrement les personnes ayant accepté de faire partie du jury. J'aimerais donc adresser mes sincères remerciements à :

- Monsieur le président du jury : M. BOUREY Jean-Pierre, professeur à l'Ecole Centrale de Lille
- Madame SIMEU-ABAZI Zineb, maître de conférences à l'Institut Polytechnique de Grenoble
- Madame ZARATE Pascale, maître de conférences l'Institut Polytechnique de Toulouse
- Mademoiselle KOLFSCHOTEN Gwendolyn, professeur assistant à l'Université Technologique de Delft, Pays-Bas
- Monsieur PALUDETTO Mario, professeur à l'Université de Toulouse
- Monsieur SAHRAOUI Abd-El-Kader, professeur à l'Université de Toulouse

Outre sa participation au jury de la thèse, je remercie encore Monsieur PALUDETTO Mario pour son aide dans les travaux de modélisation au cours de la thèse et son aide pour la correction de mon manuscrit.

J'exprime ma reconnaissance envers Messieurs TEMBINE Hamidou, SADOU Nabil et MFOUTOU Rivain d'avoir accepté de relire mon mémoire en faisant des critiques pour son amélioration. Je n'oublie pas mes chères amies Doreen BROWN et Brenda NAMUZIBWA qui m'ont aidé à corriger mes textes en anglais.

Je suis particulièrement reconnaissante à FAUVET Marie-Christine, professeur à l'Université Josphe FOURIER, qui m'a introduite dans la voie de la recherche par deux années successives d'encadrement dans mes travaux de recherche en master. Je la remercie pour ses conseils et ses encouragements qui m'ont été d'une grande utilité.

Je remercie tous les personnels du laboratoire LAAS et son directeur Raja CHATILA pour leur accueil pendant toute la durée de mes travaux de recherche. Un grand merci s'adresse également à l'équipe d'Ingénierie Système de l'Université Technologique de Delft pour leur accueil pendant mon séjour à Delft et pour la mise à disposition de l'outil ThinkTank. Les étudiants de la faculté TBM de l'Université Technologique de Delft ayant participé à la réalisation des travaux d'expérimentation et d'évaluation menés au cours de la thèse méritent également ma reconnaissance.

J'exprime ma gratitude envers les institutions qui ont financé mes études depuis mon accession aux études supérieures jusqu'à la recherche. Il s'agit :

- du gouvernement malien,
- de la fondation PATHFINDER du Docteur Cheick Modibo DIARRA,
- du laboratoire LAAS,
- de l'Université Paul SABATIER pour la bourse de mobilité (ATUPS),
- de l'Ecole Doctorale Systèmes (EDSYS) pour la bourse de mobilité

Je les remercie particulièrement pour leurs soutiens financiers qui m'ont permis de travailler dans de bonnes conditions.

Je ne saurais oublier non plus mes collègues doctorants qui m'ont tenu compagnie pendant les trois années de thèse et avec qui j'ai passé de bons moments : EL JAMAL Hani, MESSAADIA Mourad, ALBERT Vincent, GUILLERM Romaric, GOMEZ Carlos, KONE Oumar, YOBOUE Paméla, KAROUI Wafa, GUEYE Fallou, GACIAS Bernat, VERRIES Jean, BRAHIMI Houda, BANIEL Frédérique, AYALA Maria et bien d'autres encore.

Un grand merci au groupe les Chérubins et les associations CJE (Club des Jeunes pour l'Entrepreneuriat) et CPD (Cellule Pour le Développement) dont je fais partie pour leur soutien tout au long de la thèse et en particulier dans les moments difficiles.

Toute ma gratitude s'exprime envers mes amis qui m'ont prêté main forte pour l'organisation de mon pot de thèse : Fassély DOUMBIA, Mamadou TOUNKARA, Kpoto ONIVOGUI, Paméla YOBOUE, Fallou GUEYE, Aminata KONTE, Bintou KONATE, KAROUI Wafa.

Le dernier mot s'adresse à mes parents et toute ma famille, les membres des Eglises et communautés auxquelles j'ai appartenue, mes amis et proches (que je ne pourrai malheureusement pas tous citer car il me manquerait de la place et je crains d'en oublier, je m'en excuse) pour leurs soutiens permanents à tous les niveaux, notamment le spirituel et l'affectif et pour leur fidélité dans leur engagement à mes côtés.

Laboratoire LAAS, Toulouse
Le 5 novembre 2009

Jacqueline KONATÉ

Résumé

Résumé : La présente thèse porte sur la collaboration dans la conception d'un système dans un cadre Ingénierie Système (IS) et plus spécifiquement, nous sommes intéressés à la phase de définition des besoins du système ou processus d'Ingénierie des Exigences, qui est la toute première phase dans l'Ingénierie Système.

L'Ingénierie des Exigences est un processus assez complexe au cours duquel les exigences qu'un système doit satisfaire sont définies à partir de besoins provenant des différentes parties prenantes concernées de près ou de loin par la réalisation du système. Nous faisons la distinction entre le besoin qui est la perception qu'un utilisateur final a du système et l'exigence qui est la vision en termes techniques qu'un concepteur ou un développeur a du système. Le processus d'élicitation des besoins et de leur transformation en exigences techniques est un travail assez critique et demande l'implication de toutes les parties prenantes. Sur la base de ce constat, nous avons adopté une approche collaborative pour traiter la complexité de ce processus. Etant donné la nature du problème, nous avons distingué deux domaines de définition de nos travaux : l'Ingénierie des Exigences à travers l'Elicitation des Exigences et la Collaboration. Nous avons ainsi adopté une démarche dans laquelle nous faisons la distinction entre les problèmes d'Ingénierie et ceux de la collaboration.

Nous proposons une méthodologie pour l'Elicitation Collaborative des exigences qui distingue deux types de processus : les processus d'Ingénierie des exigences et les processus de collaboration. Les processus de collaboration sont définis à l'aide de l'Ingénierie de la Collaboration en s'appuyant sur les tâches d'Ingénierie identifiées aux travers de processus d'Ingénierie fournis par des normes, en l'occurrence la norme EIA-632. Des expérimentations de notre méthodologie ont été réalisées avec des étudiants en utilisant l'outil ThinkTank de GroupSystems et un prototype de spécification collaborative des exigences appelé SPECJ que nous avons développé.

Mots clés : Ingénierie Système, Ingénierie des Exigences, Collaboration, Ingénierie de la Collaboration, Innovation.

Abstract : This thesis treats the collaboration issues of design teams in the context of System Engineering (SE). Systems are more and more complex; their design requires the involvement of various skills, i.e., several stakeholders. This also involves team work between different stakeholders. Since this needs to be done correctly, it is necessary to define the methods required. In order to accomplish this, we were interested in collaboration that we considered more elaborated than simply group work. Indeed, collaboration is intended to be a group work that is better organized and structured, with clearly defined rules. To understand this problem more clearly, we were particularly interested in the process of identifying the needs of the system also called Requirements Engineering (RE), which is the first phase of the Engineering System.

RE is a very complex process during which system requirements have to be defined based on needs from different stakeholders concerned in one way or another by the realization of the system. We make a distinction between the need that is the perception of a final user of the system, and the requirement that is the vision, in technical terms, that a designer or a developer has of the system. Indeed, requirement is the technical expression of a need and it will be recorded in the specifications book in order to be transferred to the system realization team. The process of needs collection and the transformation of these needs into technical requirements is critical and requires involvement of all stakeholders. As a result of this, we decided to adopt a collaborative approach to deal with the complexity of this process. Thus, in order to define the boundaries and scope of our research work, we made a literature review on RE. We have more focused on the phase of Requirements Elicitation, the first phase of RE, because it requires the committed participation of all stakeholders. Given the nature of the problem, we have distinguished two domains in which our research work is located : RE through Requirements Elicitation and Collaboration. We have thus adopted an approach in which we made the distinction between engineering and collaboration problems. In this context, we realized a state of the art on RE, in which we presented some existing works by comparing them to ours. We followed this by including other present day research on Collective Intelligence and Collaboration Engineering.

We then proposed a methodology for Collaborative Requirements Elicitation. We separated the area of engineering from the area of collaboration by defining two types of processes : Requirements Engineering processes and Collaboration processes. Collaboration processes are defined using Collaboration Engineering based on engineering tasks. These are identified through the process provided by SE standards, namely the standard EIA-632. We also presented the tooling of the methodology and the results of the empirical studies we made.

Key words : System Engineering, Requirements Engineering, Collaboration, Collaboration Engineering, Innovation.

Table des matières

Avant-propos	v
Résumé	ix
Table des matières	xi
I Problématique et cadre de travail	5
1.1 Problématique	5
1.2 Cadre de travail : Ingénierie des Exigences	7
1.2.1 Définitions	7
1.2.2 Ingénierie des Exigences (<i>IE</i>)	8
1.2.3 Processus d'Ingénierie des Exigences	10
1.2.4 Méthodes et techniques pour l'Elicitation des Exigences	15
1.2.5 Outillage des méthodes et techniques	19
1.2.6 Synthèse	20
II État de l'art	23
2.1 La collaboration en général	23
2.2 Pourquoi la collaboration ? Dimension sociale	25
2.2.1 Intelligence Collective (IC)	25
2.2.2 Des formes d'intelligence collective	27
2.2.3 L'Intelligence Collective dans les organisations	32
2.2.4 Les forces et faiblesses de l'Intelligence Collective : exemples d'illustration	34
2.3 Ingénierie de la Collaboration	36
2.3.1 Quelques concepts clés	37
2.3.2 Peut-on faire l'ingénierie de la collaboration ?	39
2.3.3 Bases formelles	42
2.3.4 Bases conceptuelles	43
2.3.5 Concevoir des processus de collaboration	48
2.4 Conclusions	52
III Proposition d'un modèle pour la collaboration	55
3.1 Introduction	55
3.2 Modèle de Collaboration	57
3.2.1 Contraintes de la collaboration : <i>CON</i>	58
3.2.2 Conception de Processus de Collaboration : <i>PROC_C</i>	58
3.3 Un modèle conceptuel	60
3.4 Environnement de Travail Collaboratif pour l'Elicitation	63
3.4.1 Modélisation des Acteurs Collaboratifs	63
3.4.2 Modèle d'Interaction avec UML	64

IV Vers une méthodologie pour l'Elicitation Collaborative des Exigences	67
4.1 Présentation générale	67
4.2 Identification des parties prenantes pertinentes et présentation de la liste initiale des besoins	69
Illustration	70
4.3 Evolution et affinement de la liste des besoins	72
Illustration	73
4.4 Analyse des besoins	76
4.4.1 Suppression des ambiguïtés	76
4.4.2 Suppression des redondances	76
4.4.3 Suppression des incohérences	77
Illustration	78
4.5 Transformation des besoins en exigences techniques	81
Illustration	83
4.6 Hiérarchisation des exigences techniques	89
Illustration	90
4.7 Vérification et Validation des exigences	94
4.7.1 Identification du problème	94
4.7.2 Proposition d'options aux problèmes	94
4.7.3 Validation des options	95
Illustration	95
V Outillage et travaux expérimentaux de la méthodologie	103
5.1 Introduction	103
5.2 Les Systèmes Supports de Groupe	103
5.3 A propos de ThinkTank	104
5.4 SpecJ	104
5.4.1 Architecture fonctionnelle	105
5.4.2 Modèle conceptuel de données de l'application	107
5.4.3 Interface de SPECJ avec scénario	107
5.5 Travaux Expérimentaux et études de cas	115
5.5.1 Démarche	115
5.5.2 Résultats	118
Conclusion générale	121
Références	123
	132
Liste des tableaux	133
	134
Table des figures	135
Annexes	137
Annexe 1 : Description de l'approche ThinkLet	137
Annexe 2 : ThinkTank	139
Annexe 3 : Choix des outils utilisés dans SPECJ	143
Annexe 4 : Description du modèle conceptuel de données avec MySql	144

Annexe 5 : Signature de quelques méthodes utilisées dans SPECJ . 145

Introduction

La réussite d'un projet de développement d'un système dépend d'une identification réelle des besoins que le système est censé satisfaire. Dans les années 70 à 80, les systèmes étaient développés selon la vision des concepteurs et des ingénieurs, sans l'implication effective des utilisateurs et autres parties prenantes. Cela a abouti à l'abandon de nombreux systèmes qui étaient très bien construits d'un point de vue technologique, mais qui ont été considérés comme des échecs car ils ne correspondaient pas aux besoins des utilisateurs. Plus tard, le marketing a trouvé plus de succès parce qu'il a réalisé que les clients et les utilisateurs finaux devaient être impliqués dans le processus de développement et spécialement durant la définition des fonctionnalités du système. De nombreux éléments doivent être pris en compte tels que : les besoins des utilisateurs, le budget, l'économie, les aspects politiques et finalement la définition des solutions du problème. La prise en compte de tous ces problèmes nécessite la mobilisation de plusieurs compétences et, par là, l'implication de nombreux participants. Pour avoir l'implication et l'efficacité de toutes les parties prenantes concernées par le système, l'une des solutions possibles est la collaboration. Il faut rappeler que la collaboration n'est pas qu'un simple travail d'équipe, elle est bien plus que cela car elle se veut structurée et organisée tout en respectant un certain nombre de règles [SBS05].

La collaboration est un concept et une pratique ancienne qui est devenue actuellement l'objet de recherches actives par l'intégration des technologies modernes de l'information. Cependant, il paraît normal de se poser la question : pourquoi tant de références à la notion de collaboration actuellement ? En effet, le contexte économique actuel encourage les compagnies à la fusion-acquisition et à l'externalisation de nombreuses activités. Par conséquent, les compagnies sont obligées de mieux communiquer avec leurs clients, partenaires, fournisseurs et filiales, pour produire les meilleurs produits et services plus rapidement et à moindre coût [MS04]. Compte tenu de la complexité, la diversité des savoir-faires et des compétences que requièrent les projets, les organisations adoptent des solutions de collaboration. L'importance de la collaboration s'accroît en fonction de la complexité du système à développer et du nombre de parties prenantes. Ceci implique la nécessité d'une approche collaborative au vu des compétences requises dans le développement des systèmes complexes. Parmi ces systèmes peuvent être cités les voitures qui ont aujourd'hui plus de 4000 pièces [AFI07b], les systèmes avioniques, etc. Une seule personne ne peut pas maîtriser la complexité d'un tel système dans son ensemble. Plusieurs études ont révélé l'importance que revêt la collaboration dans les organisations. Une étude de l'organisation *Parametric Technology Corporation* montre que les ingénieurs passent deux tiers de leur temps à collaborer [Par06]. Une autre étude de l'organisation *Frost and Sullivan* montre que plus de 36% de la performance d'une entreprise est due à son indice de collaboration, ce qui représente plus de deux fois l'impact

de son orientation stratégique (16%) et plus de cinq fois l'impact du marché et des influences de la turbulence technologique (7%) [Fro06]. De plus, les organisations se réfèrent de plus en plus aux équipes collaboratrices pour l'exécution des tâches récurrentes et collaboratives. Cependant, quand la collaboration dépasse un contexte limité et quand elle concerne un projet d'une taille considérable, il devient nécessaire d'en imaginer une nouvelle forme qui doit être structurée et organisée dans le but d'obtenir des résultats satisfaisants.

La collaboration a une dimension humaine et sociale très importante dont les facteurs ne sont pas toujours perceptibles et contrôlables. Par exemple, l'ingénierie des exigences est considérée comme une démarche de réconciliation des problèmes sociaux et techniques [Gog94]. La collaboration *Ad hoc* mène à la conception d'outils dédiés à la collaboration qui présentent beaucoup de faiblesses ; en effet, beaucoup d'outils considérés comme des chefs d'œuvre technologiques représentent des échecs car ils ne s'appuient pas sur des fondements théoriques et conceptuels solides prenant en compte les éléments essentiels [Bri06]. En plus, ces outils sont continuellement en développement à cause des améliorations des technologies qui les supportent ; ils sont liés à la technologie. De ce fait, un aspect important est la conception et la gestion explicites des processus de collaboration intégrée au développement et au déploiement des technologies pour supporter ces processus de collaboration [HDKC06].

Dans ce contexte, les études de recherche présentées dans ce mémoire visent à proposer un modèle de collaboration pour l'ingénierie système, notamment la phase de définition des besoins connue sous le nom d'Ingénierie des Exigences. Le choix porté sur cette phase s'explique par le fait qu'elle requiert l'implication de plusieurs participants et se prête bien à une étude sur la collaboration. Pour ce faire, nous avons adopté une approche qui sépare le problème en difficultés inhérentes à l'ingénierie des exigences d'une part et en difficultés liées à la collaboration d'autre part. Nous identifions d'abord un ensemble de processus d'ingénierie, en particulier ceux proposés par la norme EIA-632 [All99] et nous leurs associons des processus appelés processus de collaboration. Ensuite, ces processus dits de collaboration sont définis selon les principes de l'approche de l'Ingénierie de la Collaboration.

L'Ingénierie de la Collaboration est un nouveau domaine qui vise à concevoir et à déployer des processus pour les tâches collaboratives et récurrentes, et à concevoir ces processus dans le but de permettre aux praticiens n'ayant pas des compétences de facilitateurs d'exécuter eux-mêmes ces processus sans le concours de facilitateurs professionnels. Ainsi, à partir de processus d'ingénierie, nous soutenons qu'il est possible de définir des processus de collaboration qui seraient prévisibles, réutilisables et transférables aux novices ou aux praticiens [BdVJ03]. Comme déjà annoncé, le caractère multidisciplinaire de l'ingénierie des exigences justifie la pertinence de ce choix. En effet, un processus de développement efficace de produits complexes nécessite une collaboration efficace entre les différentes compétences. De fait, nous avons retenu comme facteur de qualité l'*efficacité*, dont l'amélioration constitue l'un des axes majeurs de nos travaux.

Ce mémoire présente le modèle de collaboration pour l'Elicitation des Exigences et est organisé en sept chapitres. Le premier chapitre présente la problématique et le contexte des travaux. Le deuxième chapitre présente l'état de l'art sur la collaboration en général, l'Intelligence Collective et l'Ingénierie de la Collaboration. Le troisième chapitre présente la proposition d'un modèle de collaboration pour l'élicitation. Le quatrième chapitre présente la méthodologie que nous proposons pour l'élicitation collaborative des exigences. Le chapitre cinq présente l'outillage de la méthodologie et les travaux expérimentaux sur cette méthodologie. Enfin, nous présentons les conclusions et les perspectives de nos travaux dans la dernière partie.

Problématique et cadre de travail

Ce chapitre situe le problème de la collaboration de façon générale et introduire le contexte spécifique à nos travaux, l'Ingénierie Système (IS), plus particulièrement l'Elicitation qui est l'étape initiale de l'Ingénierie des exigences. Après la problématique, nous présentons plus en détail la phase de l'Ingénierie des Exigences qui est la cible de notre étude.

1.1 Problématique

Il existe déjà de nombreux travaux de recherche portant sur la problématique de la collaboration. La spécificité du problème faisant l'objet de ce travail tient au fait qu'il vise la collaboration dans le cadre de l'ingénierie système, en particulier l'ingénierie des exigences. Il existe par ailleurs des processus d'ingénierie définis par des normes internationales. La structuration de ces processus laisse entrevoir la possibilité d'envisager un modèle de collaboration pour l'Ingénierie Système. En d'autres termes, le travail consiste à considérer un ensemble de processus et à déterminer comment ils peuvent être exécutés dans un cadre de collaboration. La figure I.1 résume cette idée.

Les travaux existants qui portent sur l'Elicitation Collaborative des exigences, ne traitent pas de façon particulière la dimension collaborative de ce processus [LC93], [Gul99], [Aka90]. En effet, ces travaux, sans perdre de vue l'importance de la collaboration pour la réussite du processus d'élicitation, ne proposent pas une structuration et une organisation de l'équipe qui permettent, lorsque celle-ci augmente en taille, de dépasser les limites qui émergent. Ces limites sont essentiellement dues à des facteurs humains qui sont difficilement contrôlables. Parmi elles, il y a la perte de vue de l'objectif de la collaboration qui peut se manifester par des distractions ou des problèmes au sein du groupe. En d'autres termes, ce n'est pas parce que les gens collaborent qu'ils réussissent toujours leur mission. Il est donc important de considérer la collaboration comme un moyen efficient permettant de surmonter des problèmes trop complexes pour un individu tout en évitant qu'elle soit un obstacle à la résolution de ces problèmes au sein d'un groupe d'individus.

À l'heure des technologies avancées et notamment celles de l'information et de la communication, la collaboration recouvre un domaine à deux dimensions fortement liées ; la dimension sociale et la dimension technique. Il n'est donc pas recommandé d'en choisir une et de négliger l'autre, mais il est possible de se focaliser davantage sur l'une tout en considérant ses effets sur l'autre. La dimension contextuelle nous intéresse dans la mesure où la raison d'être

de la technique est de satisfaire les besoins des utilisateurs. Ce constat constitue le point de départ et la raison d'être de la discipline de l'Ingénierie des Exigences. En effet, d'après Goguen, l'ingénierie des exigences est une approche de réconciliation des problèmes sociaux et des problèmes techniques [Gog94].

Dans ce contexte, le coeur du problème que nous traitons se situe donc au niveau de la spécification et de la définition de la partie intitulée *Élicitation Collaborative des Exigences* sur la figure I.1. À travers cette démarche, nous nous inscrivons dans la logique de Goguen pour faire face à la réconciliation du social et du technique par la collaboration.

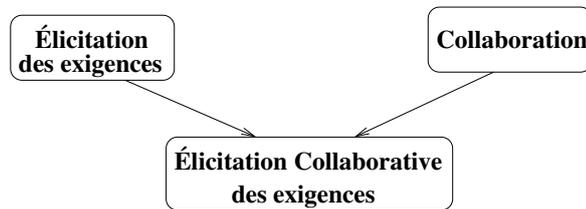


FIGURE I.1 – Collaboration et Elicitation des Exigences.

D'après ce qui précède, la collaboration et l'ingénierie des exigences ont en commun qu'elles ont toutes les deux une dimension technique et une dimension sociale. Sur la base de ce constat, nous pensons que la collaboration peut contribuer à améliorer la qualité du processus d'ingénierie des exigences.

Comme facteur de qualité, le facteur efficacité a retenu notre attention du fait qu'elle prend en compte le temps du point de vue de la durée d'exécution sans oublier la qualité du résultat. De ce fait, l'affirmation selon laquelle la collaboration peut améliorer l'efficacité du processus d'ingénierie des exigences étant en soi une hypothèse fondamentale, les questions qui en découlent et se posent à nous sont :

- *Comment la collaboration peut-elle améliorer l'efficacité du processus d'Elicitation des Exigences ?*
- *Comment définir et mesurer l'efficacité de l'Elicitation des Exigences ?*
- *Comment supporter l'Elicitation des Exigences pour en améliorer l'efficacité ?*
- *Comment peut-on améliorer l'efficacité du processus d'Elicitation Collaborative des exigences ?*

Les questions ci-dessus résument la problématique de nos travaux d'étude et de recherche. Avant de répondre à ces questions dans la suite du document, le reste de ce chapitre est consacré à une présentation dans en détail de l'Ingénierie des Exigences.

1.2 Cadre de travail : Ingénierie des Exigences

Nous présentons tout d'abord quelques définitions de concepts importants pour la compréhension du domaine. Ces définitions proviennent de normes d'Ingénierie Système.

1.2.1 Définitions

Il existe plusieurs normes d'Ingénierie Système : l'ISO-15288 [fSI03], IEEE-1220 [oEEE99] et l'EIA-632 [All99]. Dans le cadre de nos travaux, nous faisons référence à la norme EIA-632.

L'Ingénierie Système (IS) ou l'ingénierie des systèmes est une démarche méthodologique, coopérative et interdisciplinaire qui englobe l'ensemble des activités adéquates pour *concevoir, développer, faire évoluer et vérifier* un système apportant une solution optimisée sur tout le cycle de vie aux besoins d'un client tout en étant acceptable par tous [oEEE99].

Acquéreur : une entreprise, une organisation ou un individu qui obtient un produit d'un fournisseur.

Client : une entreprise, une organisation ou un individu qui : (1) commissionne l'ingénierie d'un système ; (2) est un acheteur potentiel du système ou d'une partie du système en construction ; (3) est un acquéreur du système.

Développeur : toute personne ou organisation qui prend part à la réalisation d'un système en y apportant une valeur ajoutée.

Une partie prenante est une entreprise, une organisation ou un individu ayant un intérêt ou une partie dans le résultat de l'ingénierie d'un système.

Selon l'AFIS (Association Française de l'Ingénierie Système), une partie prenante constitue une partie intéressée par l'utilisation et l'exploitation du système (voire par ses impacts sur son environnement), mais aussi un agent participant à sa conception, sa production, son déploiement, sa commercialisation, son maintien en condition opérationnelle et son retrait de service [AFI07b].

Besoin : la perception que l'utilisateur a du système [Ess02]. Ce besoin s'exprime souvent sous forme de problèmes que rencontrent les utilisateurs auxquels le système est destiné.

Exigence : il n'existe pas une définition de l'exigence, mais plusieurs telles que celle donnée par Davis qui dit que l'exigence doit être une caractéristique visible, vue de l'extérieur, du système désiré [Dav00]. Selon IEEE, une exigence est une *condition ou une capacité qui doit être satisfaite par un système ou un composant d'un système pour satisfaire un contrat, une norme, une spécification, ou autres documents formellement imposés* [IEE90]. Comparativement au besoin, qui est lié à l'utilisateur, l'exigence est la vision que le concepteur ou le développeur a du système [Ess02].

Le cycle de vie d'un système est l'ensemble des phases par lesquelles passe le système depuis l'émission des besoins le concernant jusqu'à son retrait de service. Il s'agit des phases de *Conceptualisation*, de *Conception* du système, de *Réalisation* des constituants, d'*Intégration* du système, de *Transfert vers l'exploitation*, d'*Exploitation* et enfin de *Retrait de service* [AFI07a].

Après les définitions de ces concepts clés du domaine, exposons le processus d'Ingénierie des Exigences avec quelques méthodes et des outils utilisés pour l'exécuter.

1.2.2 Ingénierie des Exigences (*IE*)

L'ingénierie des systèmes concerne la recherche d'une solution satisfaisant au mieux aux innombrables attentes et contraintes. Ces contraintes ont besoin d'être formalisées et suivies en termes d'impacts tout au long de la définition de la solution en prenant en compte les contraintes complémentaires. Celles-ci résultent des choix techniques ou organisationnels successifs. Enfin, il faut se donner les moyens de vérifier la conformité de la solution. Tout cela constitue le rôle de l'Ingénierie des Exigences. Elle est donc la première étape fondamentale de n'importe quel projet de développement de système.

Selon le GTIE (Groupe de Travail sur l'Ingénierie des Exigences) de l'AFIS (Association Française de l'Ingénierie des Exigences), l'*Ingénierie des Exigences* désigne l'ensemble des activités destinées à découvrir, analyser, valider et faire évoluer un ensemble d'exigences relatives à un système. Elle permet de montrer la satisfaction des besoins et des engagements durant tout le cycle de vie [dTIdE07].

Les éléments de base de cette discipline sont les *besoins* et les *exigences* précédemment définis dans la section 1.2.1. Les *exigences* représentent la vision du système du point de vue des concepteurs (point de vue technique); les *besoins* représentent la vision du système uniquement d'un point de vue utilisateur. Une exigence est un besoin qui est techniquement satisfaisable ou dont la solution peut être implémentée. Les exigences formalisent l'expression des besoins et des engagements des parties prenantes comme le montre la figure I.2. Une exigence bien définie permet d'assurer la réussite du projet de construction d'un système. Cependant, le passage des besoins aux exigences est une phase très critique ce qui fait de l'IE un processus assez complexe à exécuter. En effet, l'IE s'occupe de la définition des exigences depuis la phase d'émission des besoins et durant tout le cycle de vie du système.

Les exigences expriment des caractéristiques du système que les utilisateurs doivent apprécier. Par exemple, un besoin exprime que "la machine à café doit accepter au moins deux modes de paiement". L'analyse des exigences obtenues conduit à distinguer les exigences de *référence* des exigences *induites* [Ess02]. Les premières apparaissent à partir des besoins en termes techniques et prennent en compte l'expertise métier. Par exemple, "la machine à café

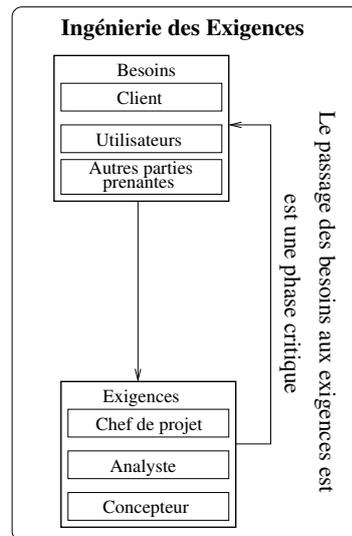


FIGURE I.2 – Besoins et Exigences [Ess02]

doit être équipée d'un lecteur de carte pour le paiement par carte bancaire". Les secondes apparaissent à partir de l'analyse des exigences et incluent tous les détails techniques que donnent les concepteurs pour répondre aux besoins exprimés à travers les exigences de référence. Par exemple, en plus de l'exigence de référence, les concepteurs rajouteront un numéro d'exigences, une justification, un niveau de priorité, une catégorie, etc. Pour des exemples bien détaillés, se référer au chapitre IV.

Comme indiqué sur la figure I.2, l'ingénierie des exigences assure le processus de transformation des besoins exprimés par les clients en exigences systèmes qui seront techniquement exploitables. Cette transformation constitue une étape assez critique de l'ensemble du processus de l'ingénierie des exigences car la réussite d'un produit dépend de la satisfaction de ses utilisateurs. Dans le domaine des systèmes qui doivent s'intégrer dans des environnements complexes, l'enjeu est d'abord de satisfaire tous les besoins et contraintes des parties intéressées de près ou de loin par l'utilisation et l'exploitation du produit. Cette satisfaction des besoins vise à répondre aux attentes et contraintes des parties prenantes impliquées dans tout le cycle de vie du produit. Elle a également comme but de conférer au système une acceptabilité et de répondre aux attentes ou aux inquiétudes de tous ceux qui sont ou seront concernés par ses impacts.

Les enjeux majeurs étant la satisfaction du client ou de l'utilisateur et la réponse aux attentes et contraintes des parties prenantes, deux espaces respectivement appelés *espace du problème* et *espace de la solution* sont définis. L'utilisateur n'est concerné que par la partie du processus définie dans l'espace du problème car son rôle est central dans la définition du problème. Les autres parties prenantes interviennent également dans cette phase et poursuivent le processus jusqu'à l'étape de la définition de la solution. Ce dernier espace voit la

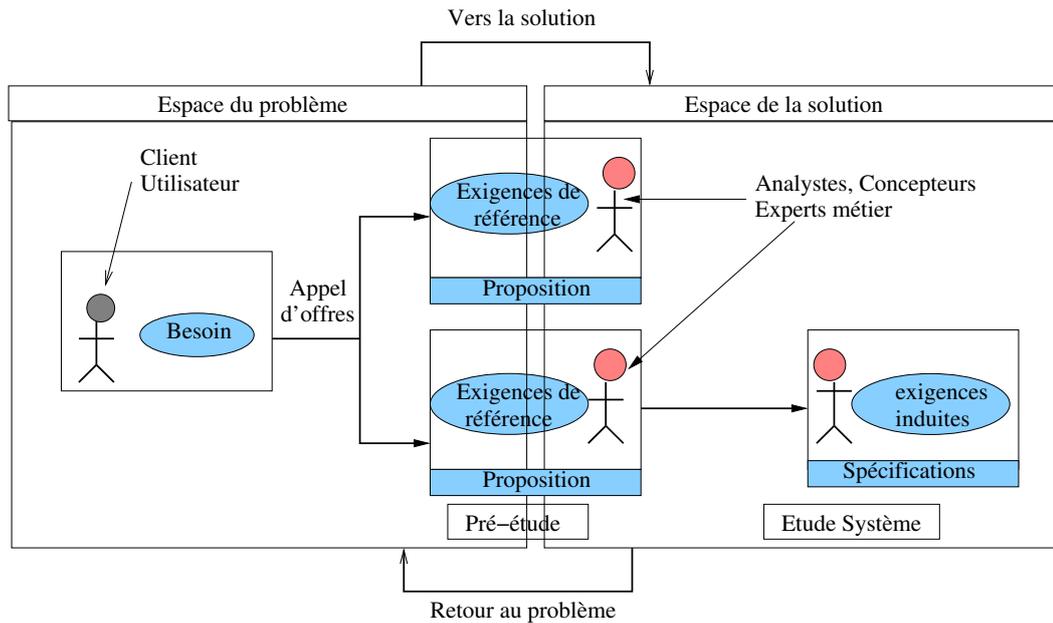


FIGURE I.3 – Les enjeux de l'ingénierie des exigences. [Ess02]

naissance des exigences à partir des besoins clients ainsi que tout le processus de transformation jusqu'à leur affinement en exigences techniques implémentables. Ce travail est mené par les concepteurs et experts métier qui ont les compétences requises pour l'assurer.

Puisque l'IE couvre tout le cycle de vie d'un produit, elle s'applique non seulement dans le cadre de la définition des fonctionnalités d'un nouveau système [KG04, KABC93], mais aussi aux cas d'évolution des systèmes existants vers de nouveaux systèmes. Par exemple, pour la résolution des questions d'interopérabilité des systèmes, l'IE est également appliquée [GBB07, BP08]. Dans la problématique de la maintenance des systèmes [SAS01, LSADG07], l'IE revêt une grande importance, notamment pour accroître la fiabilité et la performance des systèmes.

Il est un paradigme dans l'Ingénierie Système appelé *Processus-Méthodes-Outils* selon lequel toute démarche d'ingénierie doit commencer par la définition de processus, suivie des méthodes qui s'appuient sur ces processus et enfin le développement d'outils qui supportent ces méthodes. L'Ingénierie des Exigences s'est développée suivant cette démarche et les sections suivantes en présentent les différentes étapes.

1.2.3 Processus d'Ingénierie des Exigences

Selon Nuseibeh et Easterbrook [NE00], un processus d'Ingénierie des Exigences inclut les phases d'*élicitation*, de *Modélisation*, d'*Analyse*, de *Spécification*, de *Validation* et de *Gestion* comme le montre la figure I.4. D'autres auteurs proposent des découpages différents du processus d'IE comme Kotonya et Sommerville [KS98] qui ont mis en évidence toutes les phases précitées sauf celle de la modélisation. Ces différences ne s'expliquent pas

forcément par une omission de phases, mais peuvent être considérées comme différentes façons de voir le processus comme par exemple Kotonya et Sommerville qui incluent la modélisation dans la phase d'analyse. De ce fait, nous proposons les définitions ci-après provenant de l'une ou l'autre des deux visions sachant qu'au delà de ces différences apparentes, elles partagent un fond commun.

L'**Elicitation** des exigences consiste en la collecte, la capture, la découverte et le développement des exigences à partir d'une variété de sources y compris les parties prenantes humaines.

La **Modélisation** est une abstraction du problème par des représentations parfois graphiques.

L'**Analyse** se focalise sur l'examen, la compréhension des exigences élicitées et leur vérification pour la qualité en termes d'exactitude, de complétude, de clarté et de consistance.

La **Spécification** n'est autre que l'enregistrement et la documentation des exigences de sorte que celles-ci soient utilisables par les parties prenantes, et en particulier, par les développeurs qui doivent concevoir et construire le système.

La **Validation** est la confirmation de la qualité des exigences et de leur conformité aux besoins et désirs des parties prenantes. Quant à la gestion, elle est exécutée tout au long du processus d'ingénierie des exigences. Elle inclut les activités de contrôle de l'évolution et de changement des exigences, de contrôle de version, de traçabilité des exigences et les statuts des exigences.

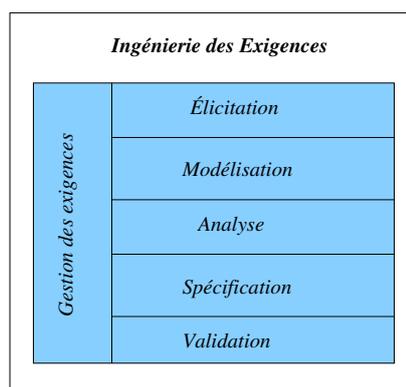


FIGURE I.4 – Les phases de l'Ingénierie des Exigences. [KS98, NE00, Cou07]

Nous détaillons ci-après et à l'aide de la figure I.5 les différentes étapes du processus d'Ingénierie des Exigences :

Phase de préparation - l'Elicitation commence par une phase de préparation qui demande l'implication de toutes les parties prenantes. Nous pouvons identifier une étape *Début* dans

laquelle les participants se présentent eux-mêmes, c'est-à-dire les premiers contacts sociaux, ils présentent leurs rôles, compétences, domaines, ..., ainsi que leurs attentes en spécifiant le but de la réunion. Pour détendre l'atmosphère, des "icebreakers" qui sont des techniques de détente peuvent être utilisées pour créer une atmosphère de collaboration [Uni08]. Den Hengst et al. ont appelé cette étape *Warm-up* [dHvdKA04]. Ensuite vient la phase *choisir/focaliser* qui permet de fixer les limites du domaine de l'environnement dans lequel le système ciblé sera utilisé, et tous les domaines reliés à ce domaine. Cette étape est très importante dans le processus d'élicitation des exigences car les étapes suivantes, que sont l'identification des parties prenantes et des classes d'utilisateurs, des buts et des tâches, des scénarios et cas d'utilisation, dépendent de la façon dont les limites sont définies [NE00]. L'étape *Identification des parties prenantes* consiste à déterminer, compte tenu des limites définies, les représentants de toutes les parties concernées par le système. La phase de préparation est aussi liée au travail initial qui permet d'obtenir les connaissances basiques du domaine de l'organisation [MBG08].

Définir les besoins - cette étape consiste en l'identification de participants ou de parties prenantes pertinentes. Par exemple, dans l'approche Easy Win Win, les utilisateurs, les parties prenantes et les développeurs sont invités [GHB⁺04], alors que dans d'autres approches telles que User-Centred Design [Gul99] seuls les utilisateurs sont impliqués. Les problèmes, les besoins, les buts et les perspectives concernant la conception du système sont "élicités". C'est une étape clé dans laquelle de nombreuses méthodes et techniques peuvent être utilisées. Des exemples d'approches sont le *brainstorming* [Osb48] qui consiste à produire le maximum d'idées possible, le *Story telling* en groupe qui consiste à raconter des récits à travers lesquels on peut identifier des besoins [AG06], ainsi que l'*Interview* [NS03]. Le choix des approches et des techniques dépend du temps et des ressources dont disposent les ingénieurs des exigences et dépend aussi du type d'information à "éliciter" [NE00]. Cette phase poursuit le travail de détermination de la portée du système, en d'autres termes le du cadrage système. Certaines méthodes pour le cadrage sont le *cadrage de la ligne du produit* [Sch02] et le *défi de cadrage du projet* [LJ97]; les deux méthodes permettent de déterminer le contour du domaine du système à construire ou du projet à réaliser.

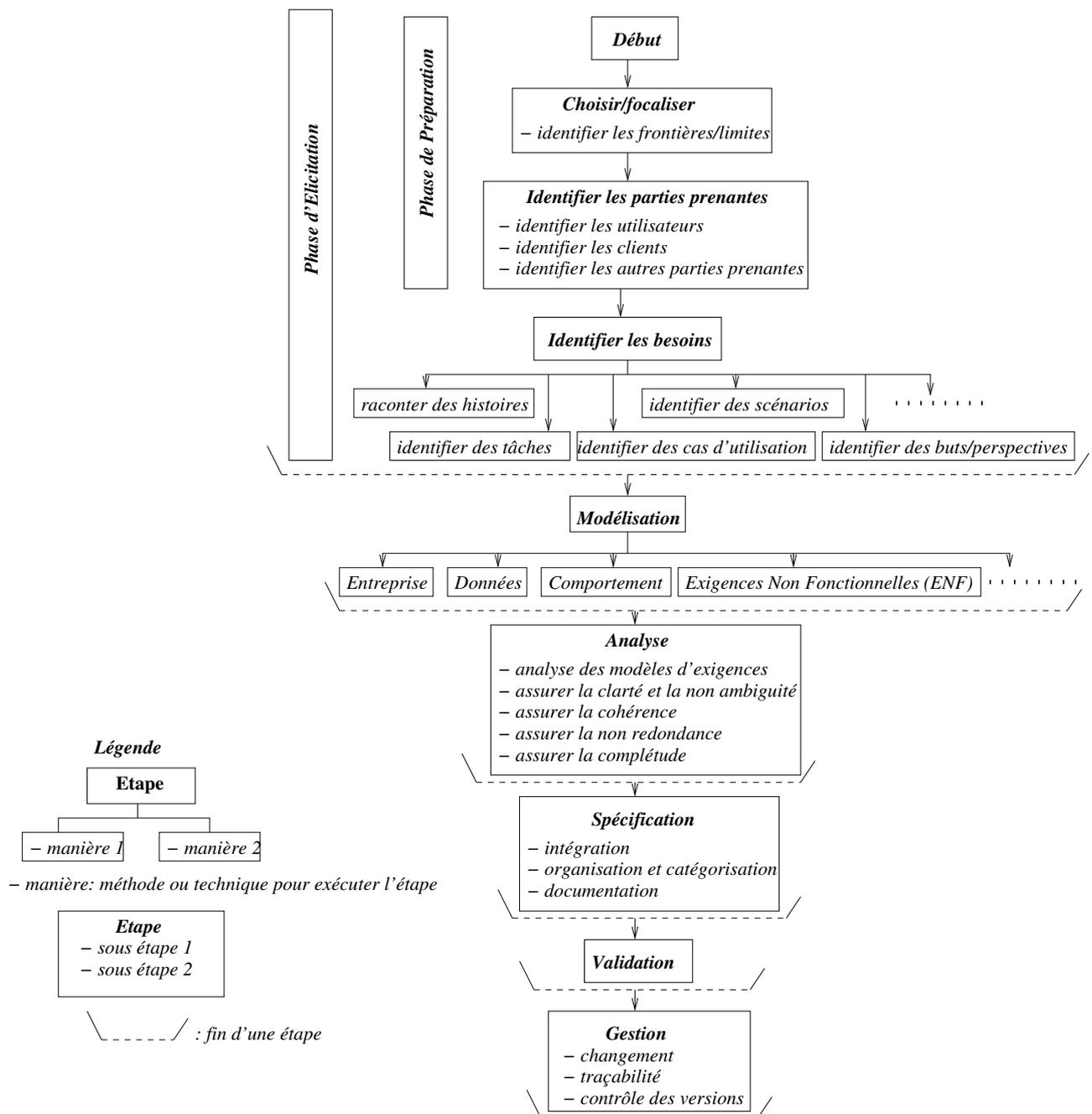


FIGURE I.5 – Vue détaillée des phases de l'Ingénierie des Exigences.

Modélisation - c'est la construction d'une description et d'une représentation abstraite d'un système pour les besoins de l'interprétation [NE00]. Diverses aspects d'un système peuvent être modélisés pour avoir une compréhension approfondie du contexte ou de l'environnement. Par exemple, quand la modélisation porte sur les données utilisées et générées par le système, pour avoir plus de compréhension par rapport à la structure des informations sur le système, la *modélisation de données* peut être utilisée. Un exemple de modèle de données est le modèle Entité-relation [Ng81]. Lorsque la modélisation est utilisée pour comprendre les parties prenantes ou la dynamique du système, ou encore le comportement fonctionnel, nous parlons de *modélisation comportementale* dont un exemple connu est la modélisation de *Cas d'utilisation* [KG04]. La modélisation peut aussi concerner le processus, dans ce cas, on parle de *modélisation de processus*. Par exemple la modélisation de processus de collaboration [BdVJ03] et la modélisation de Workflow [YB01]. Nous pouvons aussi citer la *modélisation* orientée objet [SIN08].

Analyse - c'est l'examen et l'interprétation des résultats issus de la phase de modélisation dans le but de clarifier les exigences, de supprimer les incohérences, et d'assurer la complétude et la non redondance. Les méthodes d'analyse sont entre autres la critique des connaissances [Wat97] et les techniques d'animation des exigences [AV93].

Spécification - Elle consiste à établir la liste finale des exigences en les organisant suivant des catégories. Les exigences selon différentes perspectives doivent être intégrées dans le but d'avoir une vision commune du système. Un exemple d'approche pour la spécification est la fiche *Volere* [RR07]. L'organisation des exigences peut être faite sur la base de méthodes orientées aspects [RMA03]. Des exemples d'intégration des exigences sont *la gestion de l'intégration conjointe des exigences* [Cen84] et l'intégration horizontale [GO01].

Validation - selon la norme EIA-632, la validation est focalisée sur la vérification de la version finale du document des exigences pour détecter les conflits, les omissions et les déviations par rapport aux normes [All99]. La validation cherche à certifier que les exigences satisfont les attentes des parties prenantes et à assurer qu'elles définissent les fonctionnalités attendues du système. En cas de conflits, une négociation est envisagée pour amener les gens à un accord. D'après l'EIA-632, un accord est un arrangement, non nécessairement contractuel, entre deux parties (un acquéreur et un fournisseur) qui définit les tâches à exécuter, les parties à délivrer, les critères d'acceptation à appliquer aux parties délivrées et d'autres exigences affectant le développement ou l'acquisition d'un produit. Quand il n'y a pas de conflits, les participants acceptent et valident tout simplement la spécification.

Gestion - cette étape consiste à suivre l'évolution ou le changement des exigences, à faire la traçabilité et le contrôle des différentes versions de ces exigences [Sah05] durant tout leur cycle de vie de l'exigence, c'est-à-dire durant tout le processus d'Ingénierie des Exigences [FD91].

1.2.4 Méthodes et techniques pour l'Elicitation des Exigences

Après cette présentation du processus d'ingénierie des exigences et de ses différentes étapes, nous abordons ici les méthodes et techniques utilisées pour exécuter ces étapes.

Comme nous avons vu dans la section précédente, il existe différentes méthodes et techniques pour exécuter les étapes de l'Ingénierie des Exigences. Dans cette section, nous présentons certaines d'entre elles et des outils qui les supportent.

Les méthodes et les techniques pour l'ingénierie des exigences sont inspirées des sciences de la *psychologie cognitive*, l'*anthropologie*, la *sociologie* et la *linguistique* [NE00]. De façon générale, une méthode est définie comme "*une façon de considérer quelque chose*" et, une technique comme "*une façon d'exécuter une activité avec habileté, ou les qualifications nécessaires pour la faire*" [Cam08].

Pour les besoins de l'ingénierie des exigences, nous définissons une méthode comme une façon de considérer les processus d'ingénierie des exigences, et une technique comme une procédure ou une façon d'exécuter les processus de l'Ingénierie des Exigences. De ce fait, nous considérons que les méthodes pour l'Ingénierie des Exigences utilisent les techniques pour l'Ingénierie des Exigences, et chaque technique est supportée par des outils.

Dans cette section, nous présentons d'abord les méthodes à travers une catégorisation, puis nous présentons les différentes classes de techniques avec des exemples et finalement quelques outils.

Méthodes collaboratives

Il existe selon la littérature [Cou07] des méthodes de *Modélisation*, de type *Méthodologique* et des méthodes *Collaboratives* pour l'Ingénierie des Exigences. Compte tenu du contexte de notre étude, nous choisissons de présenter ici seulement quelques méthodes collaboratives.

Dans les méthodes collaboratives, un groupe de participants travaille de manière conjointe. La différence entre les méthodes collaboratives et les méthodes de groupes est que les méthodes collaboratives sont plus structurées et organisées avec des rôles et des tâches spécifiques. Cela présente l'avantage de faciliter le contrôle et la gestion du processus. En voici quelques exemples :

Développement Conjoint d'Application (JAD) [LC93] - est une méthodologie de développement de systèmes pour supporter la collaboration parmi les membres d'une équipe de projet dans le processus de développement d'applications, spécialement pour la définition des exigences. JAD implique les utilisateurs finaux dans un atelier structuré et intensif avec le personnel des systèmes d'information et un leader de session expérimenté. Les participants

sont capables de faire ressortir des connaissances tacites. Le rôle du leader est de définir, avec les managers et les utilisateurs, la portée et les objectifs du projet et de déterminer les participants d'un atelier JAD. Durant les sessions, le leader présente les informations et capture les exigences qu'un scribe documente sous une forme standardisée. En cas de conflits entre les exigences, le leader doit jouer le rôle de médiateur. Donc, les qualités du leader déterminent grandement la réussite des sessions JAD.

Déploiement de la Fonction Qualité (QFD) [Aka90] - qui est aussi appelée "Maison de la Qualité" est une approche visant à accroître la satisfaction des clients. Les besoins des clients (le quoi) sont collectés et transformés en spécifications techniques ou solutions (le comment) qui répondent aux besoins. Une matrice de corrélation est créée pour montrer les relations entre les besoins et les vues techniques (*quoi-comment*) et aussi entre les solutions (*comment1-comment2*). Ces relations sont pondérées pour déterminer si une solution satisfait ou pas un besoin. Les solutions non pertinentes, incohérentes et redondantes sont identifiées en vue de faciliter la prise de décision tout en considérant le coût du développement [PK98]. Cette méthode est vue comme la voix du client en entreprise parce qu'elle est centrée sur la satisfaction de l'utilisateur.

Conception Centrée sur l'Utilisateur (UCD) [Gul99] - est une approche qui vise à assurer l'unité entre le système technique et le système social en impliquant activement les utilisateurs comme une partie d'une équipe interdisciplinaire du projet. Les utilisateurs sont consultés, mais les décisions finales reviennent aux développeurs. *Participatory design* [Com90] est un type d'approche UCD pour l'évaluation, la conception, et le développement des systèmes technologiques et organisationnels qui met l'accent sur l'implication active des utilisateurs potentiels ou actuels du système dans les processus de conception et de prise de décision. Cette approche est basée sur l'idée selon laquelle, l'implication des utilisateurs dans le processus de conception contribue à rendre le système plus adapté aux besoins et aux procédures de travail des utilisateurs. Un avantage majeur de l'approche est qu'elle aide les participants, et spécialement les utilisateurs, à prendre part personnellement au projet. Ils sont susceptibles de travailler pour la réussite du projet [MS00].

WinWin [GB01] - cette méthode est axée sur la négociation des exigences en groupe. Elle a été introduite par Barry Boehm [BBHL94]. Il y a quatre concepts essentiels dans cette approche :

- les *win-conditions* qui représentent les objectifs des parties prenantes,
- les *problèmes* qui sont les conflits entre les parties prenantes,
- les *options* qui sont des alternatives pour résoudre les problèmes,
- et les *accords* qui capturent les engagements partagés des parties prenantes par rapport aux win-conditions ou les options adoptées.

WinWin utilise ces concepts pour engager les gens dans un processus de prise de décision à partir d'une situation conflictuelle vers une situation de consensus.

Techniques

On trouve dans la littérature, quatre classes de techniques d'Ingénierie des Exigences [NE00] : les *techniques traditionnelles*, les *techniques cognitives*, les *techniques contextuelles* et les *techniques de groupe*. Tout comme pour les méthodes, nous décrirons quelques techniques de groupe et citerons quelques exemples des autres catégories de techniques.

Techniques Traditionnelles : ce sont des techniques utilisées depuis fort longtemps en génie logiciel pour déterminer les besoins et les souhaits des utilisateurs. Elles ont vu le jour bien avant que l'Ingénierie des exigences ne soit une discipline à part entière de l'Ingénierie Système [NE00, Cou07]. Parmi ces techniques, nous avons l'*Interview* [LG96], l'*introspection* [GL93] et l'*Analyse des tâches* [TFS07].

Techniques de la Psychologie cognitive : la psychologie cognitive est l'examen de la façon dont les gens comprennent, diagnostiquent et résolvent les problèmes qui les concernent eux-mêmes avec des processus mentaux qui interviennent entre le stimulus et la réponse [Nei67]. Les techniques pour l'Ingénierie des Exigences basées sur la psychologie cognitive ont été, à l'origine, développées pour l'acquisition de la connaissance [NE00]. Certaines d'entre elles sont l'*analyse des protocoles* [vML99] et le *Tri de carte* [SP05].

Techniques Contextuelles : elles sont vues par Nuseibeh et Easterbrook [NE00] comme alternatives aux techniques traditionnelles et aux techniques basées sur la psychologie cognitive. Comme leur nom l'indique, elles mettent un accent particulier sur le contexte et l'environnement dans lesquels le système visé sera utilisé, c'est-à-dire, l'environnement du monde réel. Parmi ces techniques figurent l'*Ethnographie* [RLD02], les *Cas d'utilisation* [KG04] et le *Prototypage* [LS92].

Techniques de groupe : ces techniques cherchent à favoriser l'accord des parties prenantes tout en favorisant la dynamique de groupe [NE00]. Elles engagent au moins quatre participants :

- *Équipes expertes* [Kru94] - Cette technique amène les membres du groupe à discuter ensemble de certains sujets intéressants ou à se centrer sur des problèmes particuliers. Dans le développement de logiciels, les "équipes expertes" ont souvent recours aux matériels tels que les questionnaires, les prototypes etc, pour provoquer et encourager le dialogue dans le groupe. Ces "équipes expertes" ont l'avantage de suivre les interactions naturelles entre les gens plus que les techniques traditionnelles parce que le facilitateur prend un rôle passif durant la session, et également parce que la structure requise est flexible. Cependant, les groupes ne sont pas habituellement des communautés naturelles, mais un collectif ad-hoc constitué pour une occasion spécifique. Bien que relativement pratique et économique à mettre en œuvre, cette technique n'est pas très appropriée pour trouver les solutions aux problèmes complexes [Cou07].

- *Brainstorming* - le *Brainstorming* est une technique pour la génération d'idées en groupe qui a été proposée par Osborn [Osborn48]. Osborn a remarqué que la génération d'idées pouvait être améliorée si les gens suivaient un protocole à quatre règles [Bri06] :
 - "ne pas critiquer les idées des autres",
 - "être ouvert aux idées absurdes et inutiles",
 - "générer beaucoup d'idées autant que faire se peut",
 - "bâtir sur et étendre les idées des autres".

Ce processus implique plusieurs participants de diverses origines et ayant différentes perspectives. Dans la première étape, un grand nombre d'idées sont produites, puis on passe à une étape de consolidation durant laquelle le nombre d'idées décroît. Un des grands avantages de cette technique est qu'elle permet la réflexion et l'expression libre qui favorisent l'émergence d'idées nouvelles, créatives et innovatrices.

- *Construction d'une histoire collective* [AG06] - Il s'agit d'une activité collective de construction d'une histoire à laquelle participent plusieurs personnes qui contribuent, avec leurs souvenirs et interprétations, aux expériences vécues. Un contexte partagé d'une tâche exécutée par un groupe peut être "élicité" et représenté à travers cette technique car elle aide à identifier, représenter et rendre explicite les éléments contextuels liés aux événements de la tâche décrite dans l'histoire, et cela, dans le but d'établir les bonnes relations entre les éléments [SB05]. Pendant que les membres d'une équipe contribuent à créer une histoire concernant un travail ou une situation expérimentée par eux, ils peuvent progressivement en rajouter, poser des questions, faire des corrections et des commentaires et aussi faire des objections. Une phase de négociation peut être envisagée, si cela est nécessaire, pour faire des contre-propositions, les accepter ou les rejeter.
- *Conception collaborative* [AEF⁺00] - Elle est une approche pour la conception avec des acteurs multiples. Cette conception est faite à travers l'interaction, la coopération, la communication, l'harmonisation et la négociation entre différents acteurs [JCLW08]. Dans le processus de la conception collaborative, plusieurs concepteurs complètent l'objet à concevoir (artéfact). Il s'agit d'une conception dite "en même temps" ou "en parallèle". Une conception complète d'un artéfact inclut la capture des exigences pour cet artéfact basées sur des retours des participants, la spécification de l'artéfact lui-même, le processus de création de l'artéfact et ainsi de suite à travers tout son cycle de vie [KSFBY03].

La plupart des approches citées ci-dessus ne sont pas exclusivement dédiées à l'Elicitation des Exigences. Par contre, elles sont ou peuvent être utilisées pour les besoins d'Elicitation des Exigences. En outre, chacune d'elles a ses points forts et ses points faibles [GL93], si bien qu'un débat sur la suprématie de l'une par rapport à l'autre de manière générale ne serait

que spéculation. En revanche, une telle discussion aurait un sens dans un contexte spécifique, car certaines approches peuvent s'avérer mieux adaptées que d'autres selon les situations.

Chacune des techniques et méthodes précédemment citées peut être supportée par un outil. En effet, selon Nuseibeh et Easterbrook, l'utilisation d'un outil automatisé est essentiel pour une exécution efficace des activités du processus d'Elicitation des Exigences [NE00]. Ainsi, nous présentons dans la section suivante, quelques outils pour l'élicitation.

1.2.5 Outillage des méthodes et techniques

La plupart des outils existants pour l'Elicitation des Exigences sont des outils pour les phases de modélisation et de gestion. Selon Coulin, cela peut s'expliquer par le fait que la modélisation et la gestion sont souvent la cible de développement d'outils alors que les autres phases telles que l'élicitation sont considérées comme des activités plus sociales [Cou07]. Cependant, les outils jouent un rôle important dans chaque phase y compris les moins techniques car, les outils sont essentiels pour l'efficacité du processus [NE00].

Les outils existants sont regroupés en catégories selon des similitudes en termes d'usages. Il n'existe pas de classification communément adoptée comme dans le cas des techniques précédemment vues [Cou07]. Nous nous focaliserons davantage sur des outils qui permettent la collaboration tout en donnant quelques exemples sur d'autres types.

Outils de collaboration - Ils supportent des tâches de groupe et spécialement la phase d'Elicitation des Exigences. Ils sont souvent une collection de supports basiques tels que les outils de messagerie instantanée et les outils de vote [BGB01]. Ils peuvent aussi inclure la vidéo conférence, tout un environnement virtuel spécialement conçu pour les projets de groupe. Les outils de collaboration visent à promouvoir une compréhension partagée de leurs utilisateurs et la résolution conjointe des problèmes. Des exemples sont CRETA (Collaborative Requirements Engineering Support Tool) [TdAFdM02] et WikiWinWin [YWK⁺08] qui est un outil pour le modèle de négociation des exigences prôné par l'approche WinWin [BBHL94]. WikiWinWin est basé sur le principe de contribution collective des idées appelé le wiki. Nous avons également les GSS (Group Support Systems) dont GroupSystems est un leader dans la construction de tels systèmes [Gro08]. Les GSS sont en général des applications Web supportant les principales activités collaboratives et qui sont souvent utilisés pour l'Elicitation des Exigences. Ils permettent aux utilisateurs d'enregistrer des rendez-vous, de faire des échanges "d'Emails", d'inclure d'autres dispositifs tels que l'agenda, les supports de messagerie instantanée et de vote. Ils permettent aussi le chargement et la recherche de document et de faire également des questionnaires. Les GSS sont basées sur quelques patrons de collaboration comme "générer", "réduire", "organiser" et "le consensus" [BdVJ03] que nous présenterons en détail dans le chapitre suivant. Des études ont révélé que l'utilisation des solutions telles que celles proposées par GroupSystems est très utile pour l'efficacité du processus et permet

même de réduire le temps de travail à 50% et faire un gain sur le temps du projet entre 70% à 90% [FSHD07, DHNV90].

Nous avons réalisé deux cas d'études avec l'outil ThinkTank de GroupSystems que nous présentons en détail dans la section 5.5 à la page 115.

D'autres types d'outils sont des *outils de modèles ou templates* qui sont essentiellement centrés sur la modélisation des exigences dans une perspective de documentation. Par exemple, la fiche Volere [RR07] pour la spécification et ScenarioPlus pour les cas d'utilisation [Ale97]. Il y a aussi les *outils de gestion des exigences* qui sont, comme leur nom l'indique, principalement utilisés pour identifier, organiser et enregistrer les exigences selon un format spécifique et de manière hiérarchique et aussi assurer la traçabilité et le contrôle de versions. On peut citer les outils AnalystPro [IBM08] et Truereq [Tru08]. Il y a également des *outils pour les études* comme QuestionPro [Que08]. Il existe bien d'autres outils encore, mais nous nous limiterons à ceux déjà cités.

1.2.6 Synthèse

Le but de ce chapitre n'était pas de faire un état de l'art de la discipline de l'Ingénierie Système, il vise simplement une présentation succincte permettant de comprendre le cadre de nos travaux, à savoir l'Ingénierie des Exigences (IE).

Dans cette section, nous avons abordé la discipline de l'IE dans le but de faire ressortir les principales étapes avec les défis qui leurs sont associés d'un point de vue sociologique et technologique. Nous avons également vu les méthodes et les techniques actuelles de l'IE et leurs spécificités ainsi que les outils existants pour les supporter. Cette suite dans la présentation des étapes suit le principe : *Processus-Méthodes-Outils*.

De toute évidence, cette discipline est très vaste. Il existe de nombreux travaux à réaliser malgré le nombre incalculable de travaux déjà effectués dans ce domaine. L'Ingénierie des Exigences demeure un domaine de recherche très actif et très prometteur.

En conclusion, nous constatons que la phase d'Elicitation des Exigences intègre les aspects sociaux et qu'elle requiert essentiellement beaucoup de collaboration. De plus, la réussite des autres étapes dépend beaucoup d'une bonne réalisation de l'élicitation. Sur la base de ce constat et considérant l'objet de nos travaux, c'est-à-dire la collaboration, notre contexte de travail cible est désormais l'Elicitation des Exigences. Ainsi, eu égard à l'intérêt que nous portons à l'Elicitation Collaborative des Exigences, nous nous focalisons particulièrement sur la dimension collaborative du processus.

Toutes les techniques et méthodes dites de collaboration que nous avons présentées dans la littérature ont des atouts indéniables de même qu'elles présentent l'avantage de permettre à plusieurs personnes de travailler ensemble pour résoudre des problèmes. Cependant, elles

ne s'appuient pas sur une construction de la collaboration, c'est-à-dire une architecture avec des règles. Or, la collaboration n'est pas qu'un travail de groupe, elle est aussi structurée, organisée pour permettre à une équipe de travailler de manière harmonieuse et singulière afin d'atteindre un objectif commun comme s'il s'agissait d'un individu. Cela vise l'accroissement de l'efficacité de l'équipe. Dans cet esprit, notre objectif est de proposer une méthodologie pour que l'Elicitation Collaborative des Exigences emprunte une démarche de construction qui intègre la prise en compte de la dimension collaborative du processus.

Ainsi, puisqu'une nouvelle approche appelée Ingénierie de la Collaboration considère la collaboration comme un système à part entière – d'où le mot ingénierie –, et étant donné la proximité de notre vision à cette approche, nous avons choisi la démarche de l'Ingénierie de la Collaboration pour réaliser la dimension collaborative de la méthodologie que nous nous proposons de développer.

CHAPITRE II

État de l'art

Nous distinguons deux dimensions de la collaboration ; l'une plutôt sociale et l'autre technique. L'état de l'art concerne ces deux dimensions, mais il est surtout axé sur la seconde au regard de la problématique énoncée. Précisons que les deux dimensions sont indissociables pour une étude efficace de la collaboration. De nombreux travaux ont été réalisés sur la collaboration suivant les deux axes, ainsi que de nombreux outils conceptuels et automatisés. Dans ce chapitre, nous présentons d'abord la collaboration dans ses aspects généraux, puis abordons le concept d'Intelligence Collective et exposons enfin l'Ingénierie de la Collaboration qui est une base importante pour la suite de nos travaux.

2.1 La collaboration en général

Il y a une collaboration lorsqu'il y a au moins deux personnes qui visent le même objectif. La collaboration se réfère à une large sphère d'activités, elle peut ainsi être définie de différentes façons. La définition donnée par Briggs et al. est plus générale et elle est valable pour n'importe quel type de collaboration [BdVJ03] :

La collaboration est un effort conjoint vers la réalisation d'un but commun.

Certains domaines d'application dans lesquels la recherche sur la collaboration est en effervescence sont notamment celui des services publics [PRS⁺00] et celui de la communication [Wal00], sans oublier les domaines techniques avec les concepts tels que la *Conception Collaborative* [TH08], l'*Ingénierie Collaborative* [WdGM98], la *Prise de Décision Collaborative* [JZBC08, KS08], la *Maintenance Collaborative* [SAZ06], etc. Cela révèle l'intérêt de la thématique et la pertinence d'une étude portant sur ce sujet.

En effet, la motivation que suscite l'étude de la collaboration tient à l'hypothèse implicite suivante : *La contribution résultant de la collaboration des individus est supérieure à la somme de leurs contributions individuelles sans collaboration.* Cette hypothèse est une transposition de la théorie de l'*émergence* qui repose sur la constatation selon laquelle dans un ensemble formé de parties différentes, le tout est davantage que la somme des parties [Lau97]. Par exemple, un organe du corps comme le cœur est plus que la somme des cellules qui le compose. Il n'y a qu'à observer le fonctionnement de l'organe pour s'en convaincre.

Par analogie, nous considérons que l'ensemble des personnes collaborant à la réalisation d'un travail peut être vu comme un système complexe lorsque le nombre croît et pour lequel chaque individu constitue le sous-système élémentaire avec ses propriétés. Ces propriétés

représentent les *compétences* des personnes qui représentent les sous-systèmes dans ce cas. Puisque selon les domaines de collaboration les compétences ne sont pas les mêmes, il en découle plusieurs types de systèmes de collaboration. Ainsi, il existe des systèmes de collaboration dans un contexte sportif comme le football, dans l'économie, le domaine militaire, etc. Etant données la double dimension sociale et technique de la collaboration, un système de collaboration doit tenir compte des êtres humains de façon individuelle et collective dans leur environnement de travail et de l'objet de leur collaboration qui est le but à atteindre ainsi que les moyens techniques pour atteindre ce but. Autrement dit, la collaboration peut être vue comme l'art de faire accomplir un travail par une équipe de personnes de façon efficace et harmonieuse. Cette efficacité tient essentiellement à la bonne mise en œuvre de la collaboration.

Dans la littérature, la collaboration est surtout évoquée avec la *coopération*, la *coordination* et la *communication*. La distinction entre ces notions n'est pas très courante ni évidente. Cependant, cette distinction est essentielle parce qu'il est question de choses bien différentes.

La *coopération* est le processus de raisonnement et/ou de mise en commun des connaissances dans le cadre de la résolution de problèmes [SBC96]. De Terssac et Maggi disent que la coopération est un moyen pour dépasser les limites individuelles [dM96]. D'autres définitions sont données par Schmidt et Banon [SB92] et Soubie [Sou98] entre autres.

La *coordination* est définie par Maggi comme le complément indispensable de l'activité de coopération [Mag97]. En effet, elle est l'ensemble des règles d'actions qui structure la coopération. Ainsi, la coopération et la coordination constituent les deux dimensions de l'action sociale et collective : l'une étant la finalisation et l'autre la régulation [GS04]. Dans le cadre du travail coopératif, il y aura une répartition claire du travail entre les participants. Concrètement, il sera affecté à chacun une tâche claire et précise dont il est responsable, puis les travaux individuels seront rassemblés pour former le travail final [Reb07, Pot09]. Cependant, tous interagissent pour la cohérence du travail final.

La *collaboration* est un effort conjoint vers un but commun [BdV03]. Elle est aussi vue par Gray comme un processus dans lequel les parties qui voient différents aspects d'un problème peuvent l'explorer de manière constructive et peuvent chercher les solutions qui vont au delà de leurs visions limitées [Gra89]. Contrairement au cas de la coopération, il n'y a pas de répartition du travail entre les participants dans un contexte de collaboration. Tous travaillent ensemble à chaque étape de l'exécution du travail de sorte que le travail individuel ne soit pas identifiable [Pot09]. Chaque étape est entièrement réalisée ensemble, le travail peut être divisé mais une représentation commune est maintenue tout au long des interactions [Reb07]. La collaboration est donc une co-construction et les rapports inter-participants y sont souvent qualifiés d'horizontaux.

La *communication* représente un élément essentiel de la coopération, de la collaboration et de la coordination. Cependant, la particularité de la communication est qu'elle n'a pas nécessairement un but et elle est utilisée comme moyen pour coopérer, collaborer et coordonner. Ainsi, la communication n'est pas une finalité en soi, mais un moyen pour atteindre un but [GS04]. La communication dans la collaboration est plutôt synchrone même si la communication asynchrone n'est pas impossible. Nous avons l'inverse dans la coopération [Pot09].

Pour finir avec les généralités sur la collaboration, voici quelques éléments importants de la collaboration que Soliman et al. appellent des ingrédients essentiels de la collaboration [SBS05] :

1. la participation d'au moins deux personnes ou plus est requise, sans quoi aucune collaboration n'est envisageable,
2. un espace de travail partagé qui peut être réel ou virtuel,
3. une bonne gestion du temps qui engendre une meilleure productivité,
4. un objectif commun pour tous les participants. En effet, quelles que soient les raisons qui ont amenés les participants dans le groupe de travail, ils doivent être d'accord sur un même objectif,
5. une focalisation sur l'objectif du groupe pour éviter toute distraction et toute divergence,
6. un langage commun qui n'est pas forcément le langage parlé, mais qui peut simplement être symbolique car n'ayant pour but que de permettre les interactions,
7. une maîtrise du domaine de l'objectif visé est requise pour les participants,
8. enfin l'interaction, sans laquelle il n'est pas possible de parler de collaboration.

Abordons maintenant les aspects sociaux de la collaboration.

2.2 Pourquoi la collaboration ? Dimension sociale

"La seule voie qui offre quelque espoir d'un avenir meilleur pour toute l'humanité est celle de la coopération et du partenariat" [Ann01].

"Les grands enjeux de l'humanité ne sont pas la faim, la pauvreté, le développement durable, la paix, la santé, l'éducation, l'économie, les ressources naturelles ... mais notre capacité à élaborer de nouvelles organisations capables de les résoudre. Notre enjeu principal est l'intelligence collective" [Nou07].

2.2.1 Intelligence Collective (IC)

Le terme intelligence est défini par le dictionnaire Larousse comme *l'ensemble des fonctions mentales ayant pour objet la connaissance conceptuelle et rationnelle* [Lar09]. Les recherches dans le domaine de la neuro-chirurgie et de l'éducation ont prouvé que l'intelligence

n'est pas héréditaire et qu'elle n'est pas statique non plus, c'est-à-dire qu'elle ne se détermine pas par le nombre de neurones [DD97]. En revanche, elle provient du nombre de synapses qui interconnectent ces neurones et de l'acquis d'informations accumulées dès la naissance. L'intelligence découle d'un processus dynamique basé sur l'acquisition des connaissances en passant par des interconnexions et des échanges. Plus ces derniers sont riches, plus l'intelligence s'accroît. À l'inverse, lorsque qu'ils diminuent, cela conduit à une perte de l'intelligence qui peut être due à la maladie ou la vieillesse.

De nos jours, des recherches s'intéressent à ce phénomène observé au niveau individuel en vue de le transposer au niveau collectif (un groupe d'individus). On y fait parfois un parallèle avec une organisation en mettant l'accent sur la nécessité d'accroître les interconnexions entre les membres et les entités de cette organisation. Cela a pour conséquence de développer l'intelligence de l'organisation et de capitaliser les informations qui sont nécessaires à aux interconnexions qui résident en son sein [Zar05, BMA⁺96]. En effet, les insectes sociaux tels que les abeilles, les fourmis et les termites ont pendant longtemps été considérés (à tort) plus intelligents que les insectes solitaires étant donnée la complexité des tâches qu'ils accomplissent, par exemple une termitière [CD05]. Quand bien même les membres de ces sociétés ne peuvent être individuellement qualifiés d'intelligents, le collectif l'est à cause de sa capacité à réaliser des constructions sophistiquées, à s'adapter à des environnements changeants, à trouver le plus court chemin à une source de nourriture, etc. Autant d'activités collectives dont la complexité va bien au delà des simples capacités de chacun des individus : on parle alors d'*intelligence collective*.

L'expression Intelligence Collective a été introduite pour la première fois par le sociologue Pierre LEVY dans son ouvrage intitulé "*L'Intelligence Collective : Pour une Anthropologie du Cyberspace*" [LEV97]. Il l'a définie comme étant *une intelligence partout distribuée, sans cesse valorisée, coordonnée en temps réel, qui aboutit à une mobilisation effective des compétences*.

Les termes *partout distribuée, sans cesse valorisée, coordonnée en temps réel et qui aboutit à une mobilisation effective des compétences* signifient respectivement que : personne ne sait tout, tout le monde sait quelque chose, le savoir est réparti ; l'être humain est la richesse centrale du collectif humain organisé et l'économie des qualités humaines est très fondamentale car chaque membre du collectif est porteur d'une richesse non négligeable qui lui garantit au sein du collectif intelligent une place et une contribution uniques ; un outil de support à l'intelligence collective, le cyberspace, qui seul permet une communication médiatique à grande échelle est nécessaire ; l'intelligence collective, au delà des aspects théoriques et philosophiques prône une nouvelle organisation sociale effective et efficace soutenue par les compétences, le savoir et les connaissances [Cai01].

Il existe plusieurs types d'intelligences collectives que nous présentons dans la section suivante.

2.2.2 Des formes d'intelligence collective

Il est très important de noter que l'intelligence collective existe depuis toujours car elle est à l'origine des organisations sociales (groupes, associations, ...). Selon Noubel, il en existe trois formes que nous présentons ci-après [Nou07] :

Intelligence collective originelle

Cette forme d'intelligence est celle qui découle d'un petit groupe d'individus (10-20 personnes, pas plus) qui collaborent en partageant le même espace physique. Tout individu a d'une manière ou d'une autre expérimenté ce type d'intelligence collective. Il peut s'agir entre autres d'un groupe de travail, d'une association, d'une équipe sportive, ... Dans le cas particulier d'une équipe de sport, tout joueur a un rôle spécifique qu'il joue au sein du collectif à un moment donné. L'équipe fonctionne au niveau global comme un tout coordonné et harmonieux sans que l'ordre ou l'information ne suive un quelconque chemin hiérarchique pour atteindre les autres membres, et cela dans des contextes plus ou moins complexes. Ainsi, tous les joueurs ont une représentation globale du problème à résoudre et sont capables de raisonnements. L'équipe acquiert ainsi une identité propre faisant d'elle une équipe différente des autres dans la même discipline, le football par exemple. Il en va de même pour les orchestres de musique au sein desquels chaque musicien perçoit la mélodie d'ensemble et par conséquent réagit instantanément de manière parfois improvisée. Cela confère à l'orchestre son style qui est unique.

Quelques-unes des principales caractéristiques de l'intelligence originelle sont :

1. ***Un Tout émergent*** : chaque groupe d'individus (orchestre, équipe de sport, ...) a un caractère, un style et un esprit différents, ce qui pousse à identifier le groupe à un individu avec une personnalité. La réussite du groupe (le Tout) devient éclatante lorsque son individualité s'affirme.
2. ***Un espace "holoptique"*** : il consiste en un espace physique ou virtuel dont l'architecture est intentionnellement conçue pour donner à ses acteurs la faculté de voir et percevoir l'ensemble de ce qui s'y déroule. Ils vont donc pouvoir agir en s'accordant. L'holoptisme est défini comme l'ensemble des propriétés incluant la transparence "horizontale" (perception des autres participants) et la communication "verticale" avec le collectif. Dans le cadre de l'intelligence collective originelle, l'espace holoptique est réel et non virtuel permettant aux individus d'interagir directement (nos cinq sens) sans le concours d'une technologie quelconque.
3. ***Un contrat social*** : Il représente la réglementation tacite et explicite du travail ou du jeu qui assure la cohérence du groupe.
4. ***Une architecture polymorphe*** : étant donné que les relations au sein du groupe changent constamment, le polymorphisme fait allusion à une relation flexible qui peut s'adapter selon les situations. Le polymorphisme s'articule autour des expertises qui

déterminent les rôles à jouer selon les cas. Autrement dit, un participant devient leader lorsque son expertise entre en jeu à un moment donné. Ce même participant peut être amené à jouer un rôle qui n'est pas lié à ses compétences si la situation l'impose. Par exemple, un attaquant dans un match de football peut prendre la place d'un milieu de terrain et inversement si la nécessité venait à s'imposer.

5. *Un objet-lien en circulation* : c'est l'objet de la collaboration pour lequel il est nécessaire d'avoir un accord car les efforts du groupe doivent converger vers cet objet qui peut être matériel ou symbolique. Des exemples d'objets matériel et symbolique sont le ballon et un morceau de music tel une mélodie.
6. *Une organisation apprenante* : l'individu aussi bien que le collectif est engagé dans un processus d'apprentissage. Ce processus intègre les erreurs pour en apprendre plus sur elles en vue d'améliorer le fonctionnement du groupe.
7. *Une économie du don* : il s'agit là de concilier l'intérêt individuel avec l'intérêt collectif. Il est évident que l'intérêt collectif ne peut être satisfait que lorsque chaque individu du collectif s'investit comme il se doit. Cet investissement n'est effectif que lorsque l'individu est personnellement motivé, d'où un intérêt individuel. Chaque participant doit donc tirer un bénéfice individuel au travers du bénéfice collectif.

Les points de l'intelligence collective ci-dessus sont complémentaires car ils découlent l'un de l'autre. Lorsqu'ils sont bien développés et coordonnés au sein d'un collectif, ce dernier devient capable d'évoluer et de créer le futur en contexte complexe, inattendu, incertain [Nou07].

Ce type d'intelligence collective présente, outre tous les avantages qu'on peut en citer, quelques limites qui sont essentiellement numériques et spatiales. En effet, cette forme d'intelligence collective n'est adaptée qu'aux groupes avec un nombre restreint de membres. Lorsque le nombre de membres est grand, cela entraîne plus de problèmes au sein du groupe et influence négativement les interactions, empêchant ainsi le groupe d'être efficace. Quant à la limite spatiale, elle exige des personnes qu'elles soient proches afin qu'elles aient toutes une vue d'ensemble de la situation en cours à chaque instant (holoptisme), et cela dans le but de permettre à chacune d'elles de réagir en fonction. Ces limites justifieraient pourquoi il n'y a pas d'équipes sportives de quatre-vingt joueurs [Nou07].

Intelligence collective pyramidale

Cette forme d'intelligence collective dite pyramidale vient en réponse aux deux limites de l'intelligence collective originelle. Lorsqu'un collectif d'un très grand nombre d'individus est engagé dans les activités telles que bâtir, planifier, cultiver, transporter, fabriquer, etc, et qui ne s'effectuent pas toujours sur le même site, il n'est plus possible de s'en remettre aux seules capacités d'auto-régulation "sur le terrain", comme dans le cas d'un match de football. Il devient nécessaire d'avoir un responsable pour prendre des décisions et par qui

l'équipe se laisse conduire. L'intelligence collective pyramidale utilise comme moyen l'écriture sans laquelle, il ne serait pas possible au chef de transmettre des directives, de compter, d'administrer, et d'organiser des collectivités très larges, à l'échelle d'une ville ou d'un pays entier. Cette forme d'intelligence collective est très courante car elle est à la base du fonctionnement des entreprises, des administrations, des gouvernements, des armées. Ses quatre principes fondamentaux sont la *division du travail*, l'*autorité*, une *monnaie rare* et des *normes et standards*. Le premier principe qu'est la division du travail veut que chacun ait un rôle prédéfini. Cela a pour conséquence l'établissement d'une hiérarchie dans la transmission de l'information, et s'oppose au principe de l'holoptisme vu dans d'intelligence collective originelle. L'autorité quelle que soit sa nature (filiation, étatique, expertise, etc.) est source de discordance dans la transmission d'informations entre l'émetteur et le récepteur. L'autorité contrôle en établissant des règles et attribue des droits et prérogatives. Le principe de la monnaie rare repose sur la hiérarchisation qui est le propre de l'intelligence collective pyramidale, et dont usent les autorités par l'arbitraire, pour faire de la monnaie une valeur d'échange et de réserve rare dans le but d'en tirer davantage de profits. Quant aux normes et standards, ils permettent de concrétiser, de faire circuler et de faire interopérer des savoirs au sein du collectif. Le langage en est un exemple.

L'intelligence collective pyramidale présente elle aussi ses limites. En effet, contrairement à l'intelligence collective originelle qui est très évolutive, la hiérarchie pyramidale est plus ou moins inflexible et résiste au changement, d'où son inaptitude à s'adapter aux contextes instables et de complexité imprévisible.

Intelligence en essaim

Il s'agit là de l'intelligence observée chez les insectes sociaux qui opèrent par divisions de centaines, de milliers et voire de millions d'individus. Ils sont très nombreux et considérés comme intelligents en collectif et individuellement "non intelligent". Nous l'avons un peu évoqué en ouvrant ce chapitre. Il s'agit de fourmis, termites, abeilles, etc. Cette forme d'intelligence confère au collectif d'individus d'incroyables capacités d'organisation, de résistance et d'adaptation. Etant donné l'absence d'holoptisme dans ce cas d'intelligence, elle est qualifiée d'*aveugle*. En effet, aucun individu du collectif n'a une perception de la situation globale du collectif. Les sociétés d'insectes sont régies par l'instinct et s'appuient sur des conditions extérieures (température, météorologie, danger, nourriture,...), comme des contenants naturels qui dictent la marche à suivre auprès d'individus interchangeable. Cela veut dire clairement qu'il n'est pas possible de faire une identification des parties d'un tout, donc qu'il y a uniformité. Or, si ce fait est admissible pour les insectes, il n'en est pas de même que pour les humains dont l'individualité, la particularité et la supérieure capacité de raisonnement ne sauraient être ignorées. C'est pour cette raison que Bô affirme que l'intelligence en essaim n'est pas à la portée des humains [Bô08]. Cependant, cette forme d'intelligence fait l'objet de nombreuses recherches, notamment dans le domaine de l'Intelligence Artificielle

(IA) [CD05, WdLL07, JMM08].

En effet, l'IA est inspirée au départ du comportement de l'humain d'un point de vue individuel en imitant son raisonnement. Elle s'appuie sur la représentation des connaissances d'un expert et la modélisation de son processus de décision en vue de faire des systèmes appelés *systèmes intelligents*. L'Intelligence Artificielle Distribuée (IAD) est une branche de l'IA qui transpose le même phénomène à l'échelle collective (plusieurs individus). Les recherches sur l'intelligence en essaim entrent donc dans le cadre de l'IAD. Des exemples de systèmes issus de la recherche sur ces phénomènes existent déjà et parmi eux peuvent être cités les Systèmes Multi-Agents (SMA) [Fer95, Woo02, Wei99], la Simulation [Gre04, DTZ08], la Robotique Collective [Sai05] et les Réseaux [AD06].

Lévy écrit que *"la meilleure chose qu'on puisse faire avec les nouvelles technologies, ce n'est pas de faire de l'Intelligence Artificielle (IA), mais, au contraire, de l'Intelligence Collective (IC) : que les ordinateurs n'imitent pas les humains, mais les aident à penser et à faire évoluer collectivement leurs idées"* [LEV97].

En d'autres termes, au lieu de chercher à substituer les humains (mission de l'IA), il faut leur permettre de penser et d'agir ensemble (mission de l'IC).

Après ce bref exposé sur les trois types d'intelligence collective, il semble que seule l'intelligence collective originelle, en dépit de ses limites, soit la seule forme satisfaisante jusque là.

D'après [Nou07], l'intelligence collective originelle a ceci d'important qu'elle transcende et inclut l'individu. En d'autres termes, bien que l'entité émergente qui dépasse l'individu est bien distinct, cette entité inclut également l'individu dans sa relation harmonieuse en lui conférant plus de valeur (capacité). A l'inverse, les types d'intelligences collective pyramidale et en essaim ne semblent ni transcender ni inclure l'intelligence collective originelle. Elles sont donc considérées comme un passage dans l'évolution vers une autre forme d'intelligence collective qui étend les limites des trois formes que nous venons de voir. Pour ce faire, Noubel propose des caractéristiques supplémentaires aux sept caractéristiques de l'intelligence collective originelle déjà vues dans l'intelligence originelle (supra section 2.2.2) dans le but d'aller vers une Intelligence Collective globale (plusieurs dizaines à plusieurs millions de personnes) qui transcende et qui inclut l'individu. Il s'agit des éléments :

8. ***Une monnaie suffisante*** : puisque nous sommes dans le cas d'un grand nombre de personnes, comme dans l'intelligence collective pyramidale, le principe de monnaie s'applique toujours en tant que valeur d'échange et de réserve ; mais cette fois-ci en quantité suffisante. Après les temps où les propriétés domaniales avaient cours, vinrent progressivement l'âge de l'industrie avec des matières premières telles que le charbon, l'acier, ... qui ont servi à construire la diversité de produits industriels de l'époque comme les trains, les automobiles, etc. Aujourd'hui, nous en sommes à une société de l'information

qui, d'après certains, fait allusion à toutes les sphères de l'activité humaine et englobe notamment le langage, le processus d'information et la connaissance. L'information devient donc la monnaie suffisante de ce nouvel âge rendue accessible par les nouvelles technologies de l'information et de la communication (NTIC). Les ordinateurs portables peu coûteux, les PDA (Personal Digital Assistant) et autres dispositifs informatiques avec les télécommunications à haute vitesse permettent à tous, sans aucune formation spéciale excepté la capacité de lire, d'écrire et d'utiliser un clavier, de récupérer et de manipuler facilement et sans frais excessifs, des données provenant de sources variées. Ainsi, nous sommes dans un "univers numérique" avec de nouvelles catégories de communications et d'échanges qui, pour la plupart, se réalisent entièrement dans l'espace virtuel qu'est le cyberspace. Le codage numérique est passé de l'ordinateur au réseau téléphonique et à la radiodiffusion. Au même moment, la technologie réduit considérablement le coût supplémentaire du traitement de l'information informatisée ou de sa transmission le long d'un réseau. Cependant, les règlements publics, les contrats privés, les pratiques restrictives et les taxes imposent souvent aux usagers des frais quelquefois importants [HV97]. Ce dernier point est encore dû à l'influence de l'organisation pyramidale qui nous entoure.

9. ***Des normes et des standards*** : toujours comme dans l'intelligence collective pyramidale, il est nécessaire d'avoir des standards et normes dans l'Intelligence Collective globale ayant pour fonction d'assurer la cohésion, de maximiser le degré de perméabilité et d'interopérabilité des grands collectifs. Ces standards et normes ne visent aucunement à établir de pouvoirs hiérarchiques.
10. ***Un système d'information*** : il permet d'organiser et d'optimiser l'espace partagé par le collectif. Il est le moyen de liaison de nos sens à travers des interfaces, il opère des calculs, simulations et anticipations que ni nos sens, ni nos intelligences ne sont capables de réaliser, il organise et indexe la mémoire collective, il reconstruit des espaces holoptiques artificiels là où l'espace réel de proximité ne suffit plus, il met en relation les personnes suivant les besoins du polymorphisme, il nous relie au cyberespace.
11. ***Une interpénétration permanente avec le cyberespace*** : le cyberspace est la base de connaissances la plus étendue que les organisations peuvent disposer de nos jours. On peut tirer des bénéfices énormes et faire profiter les autres de nos contributions. Les espaces holoptiques offerts par les technologies restent, pour l'instant, limités aux seules facultés relatives à la vue et à l'audition.
12. ***Un développement personnel*** : enfin, pour aller vers une Intelligence Collective à grande échelle, il faut une transformation individuelle et sociétale profonde. En effet, il est nécessaire que les individus et donc les sociétés ayant appris à vivre selon un certain schéma, change désormais d'approches. Notamment, il faut une intelligence comportementale et une intelligence relationnelle. Ceci s'étend au domaine de la psychologie humaine et l'analyse comportementale ; mais ces aspects dépassent le contexte de nos

travaux.

2.2.3 L'Intelligence Collective dans les organisations

L'intelligence collective (IC) est une notion qui commence à s'intégrer dans le vocabulaire des managers qui réalisent peu à peu le besoin de réorganiser les entreprises. Ainsi, de nouvelles expressions comme *décision collective*, *réflexion collective*, *entreprise intelligente*, ... font leur apparition. L'IC devient donc la capacité d'une organisation, d'un collectif à se poser des questions et à chercher les réponses ensemble [Zar05]. Une *organisation intelligente* est une entité qui intègre l'IC dans sa politique d'organisation. Actuellement, peu d'entreprises exploitent l'IC ; et si elle est utilisée, c'est surtout sous forme de coopérations intellectuelles en particulier pendant les réflexions collectives.

Selon Pierre Lévy, *"la réflexion collective est un sous-ensemble de l'intelligence collective plus explicite, discursif et conversationnel. L'intelligence collective comprend, en effet, l'organisation et le fonctionnement dynamique de tous les éléments d'une culture"* [LEV97].

Malheureusement, pour des raisons culturelles, certaines dues aux habitudes managériales et d'autres qui s'expliquent par la défaillance des technologies de l'information et de la collaboration, la réflexion collective n'est pas très présente dans les entreprises. Pour une organisation qui se veut intelligente, un accent particulier doit être mis sur la distinction entre la réflexion collective et la communication. En effet, tandis que la communication permet d'échanger des informations sans qu'il y ait forcément des coopérations intellectuelles, la réflexion (en collectif), quant à elle, implique des coopérations intellectuelles qui permettent de créer l'information, de lui donner du sens et d'interagir sur l'information existante pour la transformer en une nouvelle information.

Le plus souvent, on pense coopérer alors qu'on communique tout simplement. La valeur créée par l'information dépend surtout de la qualité de l'interaction des personnes autour de l'information. Cette valeur résultant du collectif est meilleure à la somme des contributions de chacun.

Une autre distinction à faire est celle entre la réflexion collective et la décision collective. Olivier Zara, auteur du livre *"Management de l'intelligence collective"* [Zar05], pense que contrairement à la perception générale au sujet de l'IC : une approche voulant supprimer toute hiérarchie dans l'entreprise et menant à une organisation de l'entreprise dans laquelle toutes les décisions sont prises à la majorité, l'IC n'a pas un rapport direct avec le fait de décider. En revanche, elle est fortement liée à la réflexion, la coopération, l'innovation, la création,...

De ce fait, elle participe au processus d'émergence de la décision mais elle n'influe pas de façon systématique la prise de décision, le plus important étant le caractère collégial du processus de la construction de la décision car celle-ci a requis l'intelligence collective et les connaissances de chacun.

La réticence de bon nombre de managers quand il s'agit d'intelligence collective tient de la crainte de perdre leur pouvoir. Or, l'IC ne vise pas une repartition du pouvoir, mais un changement dans l'exercice du pouvoir dans les modes de management. Une nouvelle gouvernance des organisations est à envisager, le *management de l'intelligence collective*. Par ailleurs, une réflexion collective ne donne pas lieu à une décision intelligente à tous les coups. La qualité d'une décision ne dépend pas du caractère individuel ou collectif de la réflexion dont elle découle car, de même qu'une réflexion collective peut aboutir à une mauvaise décision, une réflexion individuelle peut conduire à une bonne décision. Ce n'est pas parce que c'est collaboratif que c'est intelligent. Par contre, si ce n'est pas collaboratif, le risque que ça ne soit pas intelligent est très élevé. Puisque mettre ensemble des personnes s'avère insuffisant, le management de l'intelligence collective vise à obtenir une décision intelligente au moyen d'outils, de méthodes, de processus et de technologies [Zar05].

Selon les initiateurs du concept, le défi auquel sont confrontées les organisations aujourd'hui est de *vouloir* (culture, valeurs, croyances), de *savoir* (compétences relationnelles) et de *pouvoir* (organisation, fonctionnement) mobiliser l'intelligence collective et les connaissances. Le management de l'intelligence collective vient donc soutenir les organisations dans ce sens en proposant comme l'indique la figure II.1 trois fondements pour une organisation intelligente. Ces piliers sont la gestion de la connaissance, les technologies de l'information et, bien sûr, l'Intelligence Collective.

Parler d'intelligence organisationnelle revient donc à assurer en permanence l'équilibre complexe de ses représentations. L'intérêt de ces représentations est de permettre de spécifier simultanément l'organisation comme système global et comme communauté d'individus collaborant au sein de ce système global. Les processus de coopération, de collaboration, de coordination et de communication deviennent alors intelligibles s'ils sont perçus comme des échanges de symboles et de désignations de symboles dans le but de construire des représentations partagées (gestion de connaissances) qui sont des éléments nécessaires à la mise en œuvre collective des projets [BMA⁺96].

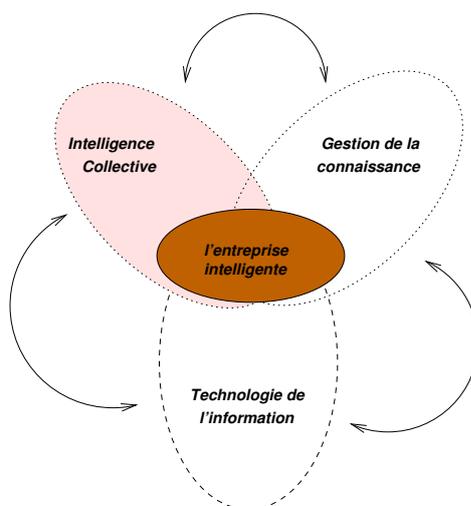


FIGURE II.1 – Organisation d’une entreprise intelligente. [Zar05]

Selon les auteurs, la notion de *contrat collaboratif* naît de cette nouvelle forme d’organisation des entreprises. Ce contrat est différent du contrat de travail en bien des points. Premièrement, sa raison d’être est la mission de l’organisation et non d’autres motivations ; deuxièmement il vise la stratégie de l’organisation et non les objectifs individuels ; enfin, troisièmement, il concerne uniquement un cadre de coopérations intellectuelles. En d’autres termes, il est un support de la convention à l’échelle collective de même que le contrat de travail l’est à la l’échelle individuelle.

Ce contrat fait office de régulateur social en ce sens qu’il incite à une coopération de manière à surpasser les conflits entre personnes.

Nous présentons quelques forces et faiblesses actuelles de l’Intelligence Collective dans la sous-section suivante.

2.2.4 Les forces et faiblesses de l’Intelligence Collective : exemples d’illustration

Nous présentons dans cette section sous formes d’exemples les réussites et les échecs de l’Intelligence Collective.

Les réussites

- *les ONG (Organisations Non Gouvernementales) et associations humanitaires* : elles reposent sur l’Intelligence Collective non pyramidale et attirent un grand nombre de personnes qui s’investissent avec dévouement. En effet, à travers ces organisations, les personnes accèdent directement à la vie de la cité et agissent efficacement sur des sujets concrets [Nou07].

- *les logiciels libres* : les communautés du logiciel libre (Freeware) arrivent à développer des produits aussi performants que les logiciels propriétaires encore appelés logiciels privés. Ces logiciels propriétaires sont issus d'organisations à structure pyramidale qui présentent des difficultés face à l'évolution et au changement. Au même moment, les collectifs du logiciel libre sont en permanence en train d'inventer des logiciels de différents types et de plus en plus performants. Un exemple bien connu est le système d'exploitation Linux [Tha08].

Les points faibles

"La masse n'a pas toujours raison, surtout s'il s'agit d'une masse moutonnaire et conformiste qui ne remet rien en question" [LEV97].

Parmi les cas non désirables d'intelligence collective il y a par exemples :

- les décisions de groupe non prises de façon libre à cause de la hiérarchie,
- les concertations sur des choix confus qui n'aboutissent pas,
- les votes démocratiques qui portent un dictateur à la tête de l'état.

L'intelligence collective est limitée par des effets de groupe (conformisme, crainte, ...), si bien que l'individu seul peut bien être plus intelligent que tout un groupe car il ne serait pas sous la pression d'un groupe qui absorbe sa capacité de discernement.

Il est important de noter, par ailleurs, que la notion d'intelligence, même pour un individu, n'est pas aisée à définir de façon précise car elle s'applique aux facultés cognitives, voire émotionnelles de ce dernier. L'application de cette notion à un système autre qu'humain pose des problèmes. Toutefois, on peut utiliser le terme intelligence pour d'autres systèmes, en l'occurrence un groupe d'individus, en précisant tout de même qu'il ne s'agit que d'une analogie [BMA⁺96].

Toutefois, les critiques ci-dessus s'appliquent plus au travail collaboratif de type humain qu'à l'intelligence en essaim (Intelligence distribuée). Contrairement aux insectes, chaque humain a son opinion et un intérêt personnel différent de celui du collectif [Nou07]. Face à ces critiques, Lévy affirme que *... l'intelligence collective consiste précisément à valoriser toute la diversité des connaissances, des compétences et des idées qui se trouvent dans une collectivité et à organiser cette diversité en un dialogue créatif et productif. La culture de l'intelligence collective travaille à établir de manière douce et pacifique un "multilogue" ouvert, qui est préférable aussi bien au cloisonnement et à l'isolement des intelligences, qu'à l'uniformité bien pensante* [LEV97].

Le but de cette présentation sur l'intelligence collective n'est pas de faire une étude approfondie de ce domaine, mais de montrer les principes fondateurs et les concepts clés de cette nouvelle discipline, les chantiers en cours et les horizons.

Nous terminons cette partie par cette affirmation de Noubel qui trace une perspective pour l'avenir de l'Intelligence Collective : *L'intelligence collective n'est pas une condition a priori, mais un état a posteriori, fruit d'un entraînement et d'un apprentissage constants ... un collectif, quelle que soit sa taille, passera par les mêmes phases complexes que celles de l'individu : une enfance, une adolescence, une phase adulte, des hauts et des bas, des crises et des victoires* [Nou07].

Pour en savoir plus sur l'état de la recherche actuellement dans cette discipline et les perspectives, l'ouvrage de Jean-François Noubel intitulé *"Intelligence collective : la révolution invisible"* est une référence en la matière [Nou07].

Après cette introduction sur la Collaboration et l'Intelligence Collective, il n'est plus nécessaire de souligner l'importance et les avantages du travail collaboratif. Cependant, les faiblesses énumérées conduisent à prendre conscience de la nécessité d'une organisation et d'une structuration pour la collaboration dans le but de la rendre fructueuse. Nous pouvons également remarquer que l'intelligence collective originelle est celle qui est adaptée à notre contexte d'étude en raison des conditions dans lesquelles elle se réalise. Aussi, il faut noter que l'utilisation d'outil du cyberspace ôte la contrainte numérique initialement fixée entre 10 à 20 personnes, permettant ainsi à un nombre plus important de personnes d'y participer. Pour ce faire, il est nécessaire de définir des règles de fonctionnement pour régir ce type de collaboration. C'est ainsi qu'une nouvelle approche intitulée *Ingénierie de la Collaboration* s'est intéressée essentiellement à la *"constructivité"* (une architecture et des règles [Sif09]) de la collaboration comme si elle était un système à part entière où les groupes de personnes à travers leurs activités collaboratives et les outils supports de groupe (*GSS : Group Support System*) sont étudiés dans le but d'améliorer la qualité de la collaboration.

2.3 Ingénierie de la Collaboration

L'Ingénierie de la Collaboration est une approche, pour la conception, des pratiques du travail collaboratif pour les tâches récurrentes de "haute valeur" (très importantes), et le déploiement de ces conceptions pour que les praticiens les exécutent eux-mêmes sans le concours de facilitateurs professionnels [BdVJ03].

L'Ingénierie de la Collaboration vise à fournir certains avantages de la facilitation professionnelle aux groupes qui n'ont pas accès aux facilitateurs professionnels.

De nos jours, l'Ingénierie de la Collaboration fait face à deux défis majeurs [BKGJdV06]. Le premier du développement d'une compréhension plus complète de la façon dont les équipes réalisent leurs tâches à travers plusieurs *patrons de collaboration* (termes définis dans la section 2.3.1) et la façon dont ces patrons peuvent être invoqués de manière prédictible. Pour le second, il faudra développer une meilleure compréhension de la manière dont les conceptions faites au moyen de l'Ingénierie de la Collaboration peuvent être transférées aux praticiens de

façon à produire des communautés de pratique autonomes et croissantes. Ces deux défis sont posés simultanément dans le cadre de notre travail sur la collaboration dans l'Elicitation des Exigences et plus particulièrement le premier. En effet, les pratiques dans l'Elicitation des Exigences sont essentiellement collaboratives et une exécution efficace du processus d'élicitation requiert effectivement plus de compréhension de la manière dont les différentes équipes participantes réalisent leurs tâches pour déterminer les *patrons de collaboration*.

Nous présentons, dans la section suivante, l'approche de l'Ingénierie de la Collaboration à travers ses concepts clés, puis la justification de l'approche elle-même, ses fondements théoriques et conceptuels, son langage de Patron de Conception : le *thinkLet*. Enfin nous terminons par la présentation de la manière de concevoir un processus de collaboration qui constitue un point important à la fois en Ingénierie de la Collaboration et en Elicitation Collaborative des Exigences.

2.3.1 Quelques concepts clés

Ici, nous précisons les concepts d'approche de l'Ingénierie de la Collaboration [BdVJ03, dVB05, BKGJdV06, KBdV⁺06] :

- La ***Collaboration*** est un effort conjoint vers un but commun. Il est nécessaire que les participants à la collaboration fassent l'effort de réaliser le but du groupe.
- Le ***But*** est un état ou un résultat désiré.
- Une ***pratique du travail*** est un ensemble d'actions réalisées répétitivement pour accomplir une tâche organisationnelle particulière.
- Une ***tâche est dite collaborative*** si la réussite de sa réalisation dépend des efforts conjoints de plusieurs individus.
- Une ***tâche est dite de haute valeur*** si l'organisation tire un avantage substantiel ou évite une perte substantielle par l'accomplissement réussi de cette tâche. Cela ne veut pas dire que l'organisation doit tirer une valeur substantielle à partir de chaque exécution de la tâche, mais plutôt que la pratique de la tâche produit une haute valeur dans le temps.
- Une ***tâche est dite récurrente*** si la tâche doit être conduite répétitivement, et si un processus similaire peut être utilisé chaque fois qu'elle est exécutée. Cela ne veut pas non plus dire que pour toute tâche et pour toute exécution, tous ses aspects de la tâche doivent être similaires ; mais cela veut dire tout simplement qu'une approche similaire peut être utilisée chaque fois que la tâche est exécutée, malgré les variations dans ses paramètres.

- Un ***Processus de Collaboration*** est une série d'activités exécutées par une équipe pour un but spécifique et dans un délai donné.
- Un ***Facilitateur*** est celui qui, à la fois, conçoit et conduit un processus dynamique qui implique la gestion des relations, des tâches et de la technologie, aussi bien que la structuration des tâches et la contribution à l'accomplissement effectif du résultat de la collaboration. Le travail du *facilitateur* est appelé la ***Facilitation***.
- Un ***Praticien*** est un spécialiste d'une tâche et qui doit exécuter des tâches collaboratives importantes telles que l'évaluation des risques ou la définition des exigences comme faisant partie de ses obligations professionnels. Il n'est pas nécessairement qu'un facilitateur professionnel qui conçoit les nouveaux processus pour les nouvelles situations, mais il est celui qui exécute un processus de collaboration spécifique sur une base récurrente, il n'a donc pas besoin de formation ni de compétence contrairement à un facilitateur.
- Un ***Ingénieur de la collaboration*** est celui qui conçoit et documente les processus de collaboration qui peuvent être facilement transférés à un praticien. Cela signifie que le praticien peut exécuter un processus sans l'aide d'un ingénieur de la collaboration ou d'un facilitateur.
- Un ***ThinkLet*** est la plus petite unité de capital intellectuelle nécessaire pour créer un patron de collaboration.
N'ayant pas pu trouver une traduction convenable du mot *thinkLet* en Français, nous recourons à ce mot comme tel dans nos lignes.
- La ***Réutilisabilité*** est la propriété qu'a un thinkLet d'être utilisé pour résoudre des problèmes autres que ceux pour lesquels ils ont été créés à l'origine.
- La ***Prédictibilité*** est la propriété qu'a un thinkLet, lorsqu'il est exécuté suivant les prescriptions, de créer des variations similaires aux patrons généraux de collaboration et des livrables avec une diversité d'équipes, de tâches et de conditions.
- La ***Transférabilité*** exprime le degré auquel les gens qui n'ont jamais créé un thinkLet peuvent l'apprendre, s'en rappeler, et l'exécuter avec succès.

Les termes ci-dessus définis sont les principaux concepts utilisés dans l'Ingénierie de la Collaboration. Leur assimilation est nécessaire à la compréhension de ce qui suit. La section 2.3.2 discute le pourquoi d'une ingénierie pour la collaboration.

2.3.2 Peut-on faire l'ingénierie de la collaboration ?

Considérons à nouveau la définition de la collaboration qui est un effort conjoint vers le but d'un groupe. L'effort est donc ici une ressource clé de la collaboration. Toutes les ressources dont notamment la connaissance ou compétence, le temps, les ressources physiques appartiennent aux participants à la collaboration et il est nécessaire que ces derniers manifestent une volonté d'engagement pour que ces ressources soient accessibles. Ainsi, le succès de la collaboration dépendra de la *volonté* de ses participants. Cependant, la volonté est quelque chose qui est propre à tout individu et elle est imprévisible, ce qui rend la collaboration difficile et complexe [KdVBS07]. Ce point est certainement la raison pour laquelle la collaboration est souvent faite de manière *ad hoc* et cela justifie pourquoi les collaborations sont souvent infructueuses en dépit de l'utilisation de technologies parfois sophistiquées. D'ailleurs, Den Hengst et al. soulignent bien l'importance de la conception et de la gestion explicites de *processus de collaboration* couplés au développement et au déploiement de technologies qui supportent ces processus [HDKC06].

La résolution de problèmes complexes a souvent donné lieu à de nouvelles disciplines. De même, eu égard aux problèmes observés dans la collaboration, une nouvelle discipline appelée Ingénierie de la Collaboration a été introduite. Nous tentons ici de justifier "l'ingéniérabilité" de la collaboration, en d'autres termes le bien fondé de la discipline.

De l'Ingénierie Système ...

Tout d'abord, il convient de définir ce que signifie le mot *ingénierie*. De manière générale, ce mot fait référence à la conception et à la construction d'un *système*. L'ingénierie est *l'ensemble des fonctions allant de la conception et des études à la responsabilité de la construction et au contrôle des équipements d'une installation technique ou industrielle* [J.O80].

Un *système* est défini dans la norme EIA-632 comme *une agrégation de produits finaux et de produits capacitants (enabling products) pour réaliser un but donné* [All99].

Un produit final est un produit qui exécute les fonctions opérationnelles du système tandis qu'un produit capacitant exécute le processus associé ou les fonctions non opérationnelles du système (supra, la figure II.2). Un avion et une voiture sont des exemples de produits finaux ; des exemples de produits capacitants sont les plans de développement, de production et de test.

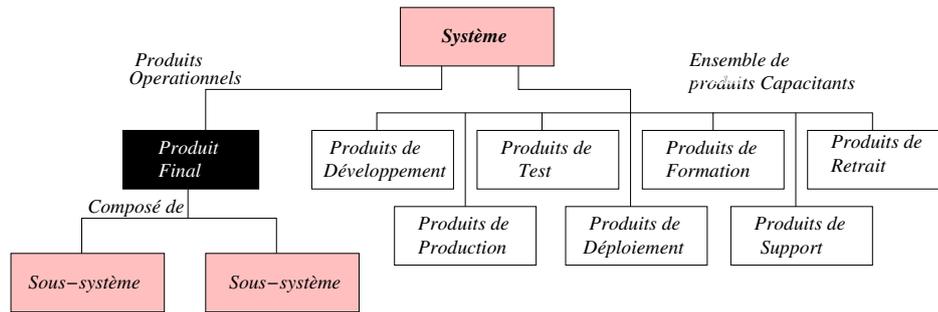


FIGURE II.2 – La représentation d’un système selon la norme EIA-632. [All99]

L’Ingénierie Système (IS) est basée sur un processus itératif constitué des étapes suivantes : (1) comprendre un problème avant de tenter de le résoudre ; (2) examiner les solutions alternatives ; (3) vérifier que la solution choisie est appropriée avant de poursuivre [INC00].

... Vers l’Ingénierie de la Collaboration

Par analogie à l’IS, l’*ingénierabilité* de la collaboration consiste en : (1) l’utilisation systématique d’une approche pour concevoir des processus de collaboration ; (2) la structuration et l’orientation de l’effort conjoint vers le but du groupe ; (3) l’évocation, avec une certaine prévisibilité de l’engagement des ressources pour atteindre le but du groupe [KdVBS07].

Dans notre démarche, la collaboration est considérée à la fois comme un *système* et un *processus*. Cela s’inscrit dans la logique de la norme EIA-632 (voir la figure II.2). En effet, l’objet de l’ingénierie est, la collaboration, le processus ou le système. Selon la norme EIA-632, un processus est *un ensemble de tâches étroitement liées qui transforment les entrées en sorties*. Cette définition demeure la même dans le contexte de l’Ingénierie de la Collaboration. Ainsi, au sens de la conception de la collaboration, la figure II.3 présente la collaboration sous forme d’un système ”entrée-processus-sortie”. L’entrée du processus est l’ensemble des

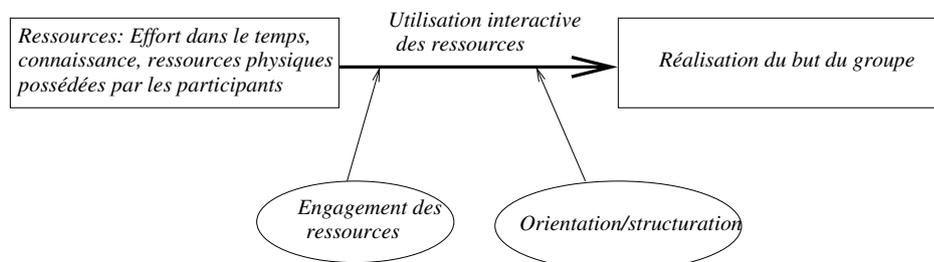


FIGURE II.3 – La représentation d’un processus de collaboration. [KdVBS07]

ressources qui peuvent être utilisées dans le processus. Ces ressources sont servies et consommées par les membres du groupe ou les participants au processus. Les ressources importantes

dans une activité collaborative sont l'effort dans le temps et la connaissance. La technologie, les outils et autres ressources physiques telles que l'argent ou une salle de réunion peuvent être utilisés dans ce processus. Les conditions qui permettent l'utilisation interactive de ces ressources conduisent à la réalisation du but du groupe. Les conditions essentielles sont que les ressources nécessaires soient engagées dans le processus et que ces ressources soient structurées ou orientées vers le but du groupe.

De nombreuses études théoriques et expérimentales soutiennent que l'*engagement* est très lié à la *performance*. En effet, lorsque le premier diminue, la seconde est fortement affectée [LL90, BKdV06]. Il existe donc différentes causes de l'engagement des ressources dans un contexte collaboratif. Parmi elles, citons la disponibilité, la difficulté d'une tâche, le sacrifice individuel et le sacrifice collectif, l'utilité de la tâche, etc. Kolschoten et al. [KdVBS07] proposent un modèle s'appuyant sur la théorie de l'instrumentalité de Briggs et al. [BKdV06] et expliquent l'engagement des ressources dans un but de groupe.

Pour structurer ou orienter l'utilisation des ressources, il est nécessaire de communiquer clairement la façon dont ces ressources doivent être déployées dans une activité avec des contraintes de telle sorte que cette activité soit efficace dans la réalisation du but du groupe. Par exemple, le comportement peut être structuré avec l'utilisation des règles. Ces règles sont incluses dans un langage de patron de conception qui est le thinkLet [KBdV⁺06].

Pour ce qui est du processus, sa conception nécessite de trouver un équilibre entre, réaliser le but, avoir l'engagement et l'acceptation des parties prenantes, diriger et gérer les ressources disponibles. Pour ce faire, des interventions sont requises pour orienter et structurer les ressources mises à disposition pour la collaboration du groupe. De telles interventions sont faites par un leader (facilitateur) de processus de groupe et elles peuvent être communiquées par écrit, ou par la formation et elles peuvent être également intégrées à la technologie à travers des restrictions fonctionnelles des actions permises par la technologie [KdVBS07].

Les processus sont aussi construits avec des thinkLets comme briques de base toujours dans le but d'accroître l'*ingénierabilité* de la collaboration. En effet, les thinkLets sont un langage de patron de conception pour l'ingénierie de la collaboration. Ils visent à augmenter l'efficacité, l'acceptation, la réutilisation, la transférabilité et la prédictibilité. Nous reviendrons sur ces points dans la section dédiée au thinkLet (section 2.3.4).

Actuellement, la littérature sur l'Ingénierie de la Collaboration comporte de nombreux travaux portant sur les différents aspects de cette nouvelle discipline, notamment les travaux sur l'approche de conception et les interventions de facilitation [BdVJT01, BdVJ03, dVB05]. Cela montre qu'il est bien possible de construire des "choses" dans la collaboration.

En fin de compte, l'Ingénierie de la Collaboration a adopté, dès le départ, le repère à cinq dimensions proposé par Seligman et al. qui permet de fixer les axes de recherches dans

cette voie [SWS89]. En effet, ce repère soutient qu'une approche d'ingénierie doit avoir les dimensions suivantes :

1. les *Manières de Penser* qui représentent les concepts et les fondements théoriques,
2. les *Manières de Travailler* qui définissent le modèle de conception de manière structurée,
3. les *Manières de Modéliser* qui sont les conventions pour la représentation des aspects du domaine et de l'approche,
4. les *Manières de Contrôler* qui, quant à elles, sont des métriques sur les méthodes pour la gestion du processus d'ingénierie,
5. les *Manières de Supporter* enfin : celles-ci constituent les outils, les approches et techniques pour aider à la conception.

Les différents travaux menés dans le domaine s'inscrivent donc dans ce repère.

Enfin, l'*ingénierabilité* de la collaboration peut être ainsi résumée. Il est possible d'améliorer la qualité de la conception de processus de collaboration pour accroître son efficacité, son acceptabilité, sa réutilisation, sa transférabilité et sa prévisibilité. Le langage de patron de conception qu'est le thinkLet occupe une place assez importante dans cette conception. En effet, c'est le thinkLet qui permet de supporter, de structurer et d'orienter l'effort collaboratif ; et il peut être utilisé comme une approche de conception systématique [KdVBS07].

De nombreux travaux ont posé les fondements théoriques de l'approche de l'Ingénierie de la Collaboration. Cependant, nous n'allons pas tellement nous appesantir sur les fondements théoriques au risque de diverger de nos objectifs initiaux, cela n'éclipse pas leur valeur intrinsèque pour nous. En revanche, notre attention sera particulièrement portée sur les aspects conceptuels qui ont un lien avec nos travaux sur l'Elicitation des Exigences. Les sous-sections 2.3.3 et 2.3.4 sont consacrées à la présentation des bases formelles et conceptuelles de l'Ingénierie de la collaboration.

2.3.3 Bases formelles

Les travaux théoriques sous-jacents à l'Ingénierie de la Collaboration se sont portés sur les points suivants : (1) la détermination des moyens qui favorisent l'acceptation, l'adoption et l'utilisation des outils de collaboration ; (2) la raison pour laquelle les organisations cessent l'utilisation d'outils de collaboration, même ceux qui produisent des avantages économiques conséquents ; Enfin (3) la façon d'accroître le savoir-faire qui permet d'aider les organisations à éviter l'abandon des outils de collaboration [BdVJ03].

Il est important de rappeler que les recherches dans le domaine de l'usage des technologies pour la collaboration se sont longtemps focalisées sur les technologies elles-mêmes, ce qui correspondait à un niveau d'abstraction peu utile, et produisaient des résultats ambigus, équivoques et un déficit d'informations sur l'usage et l'adoption des technologies [BdVJT01]. Ainsi, Briggs et al. [BdVJ03] ont introduit et adopté le concept de thinkLet comme objet

d'étude plutôt que les technologies elles-mêmes. En effet, le thinkLet est à un niveau d'abstraction au dessus des technologies, mais qui tient compte des technologies et également la dimension sociale de la question. Par conséquent, les recherches sur les technologies de collaboration peuvent être plus contrôlables et capables de contribuer de façon significative au développement et à l'utilisation de ces technologies.

Dans ce contexte, l'Ingénierie de la Collaboration s'est appuyée sur des travaux théoriques réalisés dans le domaine de l'adoption et le transfert de technologie. Ces théories considèrent que le processus d'implémentation d'une technologie exerce une influence considérable sur l'éventuelle utilisation ou l'adoption de la dite technologie [BdVJ03]. Ainsi, elles tiennent compte des dimensions sociales et techniques du problème d'adoption et d'utilisation des technologies, notamment celles dédiées au travail collaboratif. Des exemples de théories sont la *Théorie de la Structuration* [Orl92] et la *Théorie de la Structuration Adaptative* [DP94]. Ces théories sont aussi appelées des *théories de processus*.

L'avantage de ces théories dites de processus est qu'elles permettent d'avoir des détails sur les descriptions et les prescriptions de l'organisation qui cherche à adopter une technologie. Par contre, ces théories ne permettent pas à elles seules d'expliquer les changements dans l'exécution d'un processus et les résultats de cette exécution. En d'autres termes, les modèles de processus n'expliquent pas les mécanismes causaux qui sont derrière la transition de la technologie [BdVJ03]. En réponse à ce problème, il existe des modèles à facteurs ou des modèles causaux qui permettent des prévisions sur la transition de la technologie, parfois en combinaison avec un modèle de processus. Des exemples de théories causales sont le *Modèle d'Acceptation de la Technologie* [Dav93, Dav86] et le *Modèle de Transition de la Technologie* [BAM⁺99]. Un des avantages majeurs de ces modèles est que la variation des facteurs permet d'observer le comportement du processus et de prévoir ses résultats.

La suite de cette section est axée sur les aspects conceptuels de l'approche de l'Ingénierie de la Collaboration.

2.3.4 Bases conceptuelles

L'élément central dans la conception de l'approche de l'Ingénierie de la Collaboration est le *processus de collaboration*, qui lui-même s'appuie sur le thinkLet comme brique de construction. De ce fait, le fondement conceptuel de l'approche de l'Ingénierie de la Collaboration repose le thinkLet.

La figure II.4 présente un méta-modèle conceptuel pour la conception de processus de collaboration basé sur les thinkLets. Les principaux travaux de conceptualisation du domaine sont présentés, entre autres, dans [dVKB06, KBdV⁺06] :

Identification : un thinkLet est identifié par un nom qui est le plus souvent captivant et quelque peu amusant pour permettre aux personnes de le mémoriser facilement. Ce nom est aussi une métaphore qui permet à l'ingénieur de la collaboration de se rappeler de la

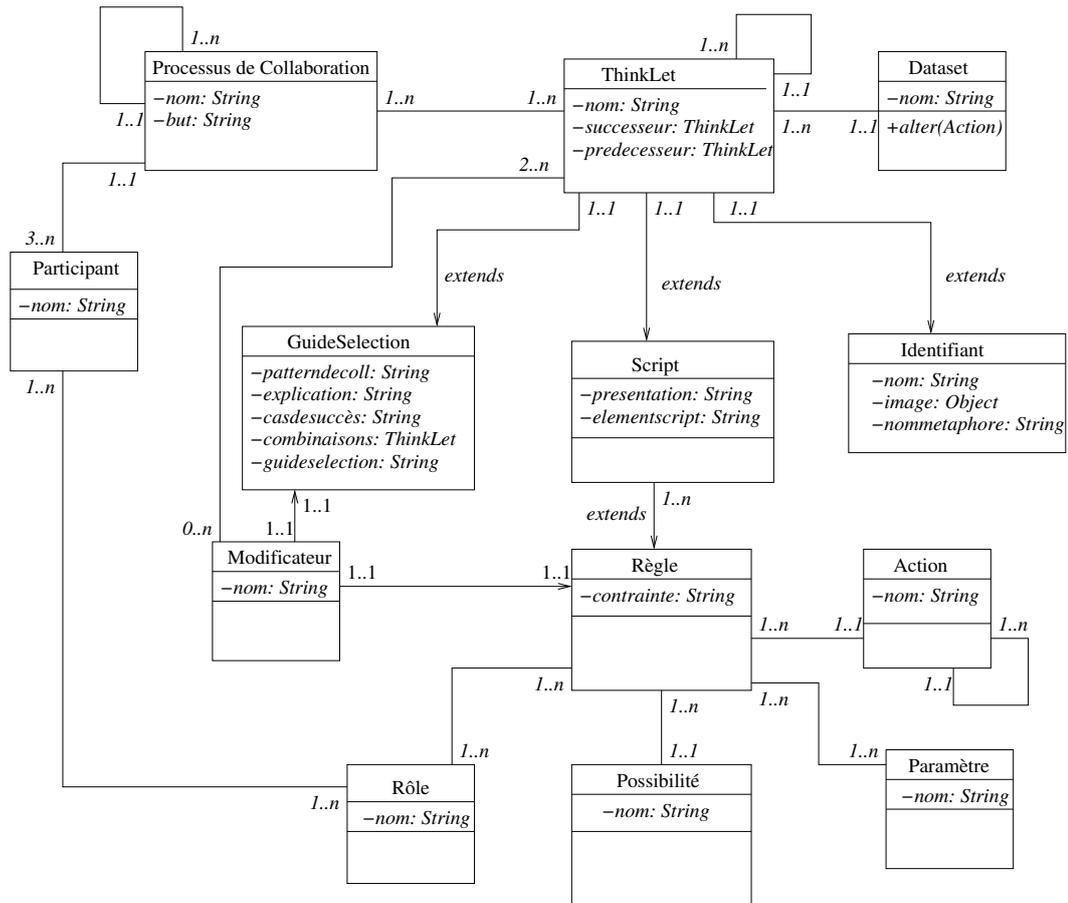


FIGURE II.4 – Diagramme de classes de processus de collaboration. [KBdV⁺06, dVKB06]

dynamique de groupe que le thinkLet invoque. Par exemple, dans FreeBrainstorming les participants contribuent librement à l’expression de leurs idées. Une image est aussi associée au thinkLet ainsi qu’une explication de la métaphore pour faciliter la remémoration du thinkLet pour les utilisateurs. Les concepteurs et les praticiens peuvent utiliser la métaphore une fois qu’ils l’ont comprise et le nom du thinkLet pour faire référence à un processus plus complexe.

Choix de sélection : la conception d’un processus de collaboration nécessite un choix des thinkLets pour une sous tâche et une sous étape spécifiques dans l’effort de collaboration. Pour faire ce choix, l’ingénieur de la collaboration doit être en mesure de prévoir les effets que le thinkLet va créer. De ce fait, un thinkLet décrit la dynamique qui émergera du groupe lorsqu’il est exécuté. Le comportement qui se produit pendant l’exécution du thinkLet et les cas de réussites sont également inclus. Une partie appelée ”approfondissement” fournit des astuces pour l’implémentation des thinkLets qui sont utiles aux nouveaux facilitateurs et aux ingénieurs de la collaboration en phase d’apprentissage. Une partie ”cas de réussites” donne des exemples concrets d’utilisation du thinkLet dans la vie réelle. En outre, des combinaisons réussies et prouvées avec d’autres thinkLets sont suggérées. Le choix de sélection est finalement offert ainsi : ”choisir ce thinkLet quand” et ”ne pas choisir ce thinkLet quand”.

Script : il constitue l'ensemble des instructions qu'un praticien ou un facilitateur donne au groupe pour créer les interactions désirées du groupe. Aussi, le script doit expliquer les moyens à l'équipe et les instruire quant aux actions qui doivent être entreprises et la façon de mener ces actions. Le script contient une présentation du thinkLet et un ensemble d'éléments que l'ingénieur de la collaboration élabore au moment de la conception.

Rôle : Il représente une collection de règles qui guident les actions d'un ensemble de participants. Certains thinkLets nécessitent que les participants jouent des rôles différents dans une activité et suivant des règles précises. Par exemple, dans les thinkLets de génération un participant peut jouer le rôle de *scribe* tandis que les autres ne font que proposer des idées ; ils sont donc des *contributeurs*.

Règle : une règle décrit une instruction pour exécuter une action avec un certain moyen et sous des contraintes spécifiques. Les règles pour un thinkLet sont la base pour les instructions aux participants quant à ce qu'ils doivent faire et dire pour réaliser l'activité avec succès. Par exemple, les règles pour un thinkLet FreeBrainstorming veulent que les contributions soient liées aux questions du sujet traité.

Possibilité : c'est l'ensemble des moyens nécessaires pour l'exécution des instructions du script. Par exemple, un thinkLet StrawPoll qui est un patron d'évaluation requiert un dispositif pour le vote permettant d'attribuer à chaque objet une valeur donnée. Ce dispositif peut être soit un outil automatisé comme les outils pour le vote électronique, soit tout simplement un dispositif plus traditionnel tel qu'un tableau ou une feuille de papier.

Action : Il s'agit d'une action individuelle faite par les participants en utilisant les possibilités qu'ils ont. Ces actions sont, entre autres, ajouter, éditer, enregistrer, supprimer, juger ou associer des concepts.

Paramètre : Tout thinkLet a un certain nombre d'informations qui doivent être envoyées à tous les membres du groupe afin qu'ils puissent travailler efficacement. Ces informations sont instanciées au moment de la conception ou de l'exécution du thinkLet. Par exemple, pour un thinkLet de génération, une question de brainstorming doit être instanciée, et pour un thinkLet de vote, des critères d'évaluation doivent être instanciés.

Modificateur : il est une règle réutilisable qui peut être appliquée à un ensemble de thinkLets pour créer un changement répétable et prévisible dans les dynamiques de groupe que ces thinkLets produisent.

Une **Contrainte** est une limitation ou une directive sur la façon dont une action doit être exécutée. Par exemple dans le thinkLet RichRelations, le nom d'une catégorie est doit être inspiré de celui d'une relation entre deux objets.

Langage de Patron de Conception pour la collaboration : *thinkLets*

Les patrons de conception encore appelés *design patterns* ont été introduits par Alexander, un architecte des années 1970 qui avait remarqué une récurrence des problèmes qui surviennent dans la phase de conception d'architecture. Il imagina le concept de patron comme suit : "un patron décrit un problème qui se répète et se répète encore et décrit donc le noyau de la solution à ce problème, de manière que vous pouvez utiliser cette solution plus d'un million de fois sans jamais l'avoir fait de la même manière deux fois" [AIS⁺77]. Le langage qu'il proposa comporte deux cents cinquante trois (253) patrons couvrant tous les aspects de la construction des bâtiments [Mat08].

Le concept fut repris plus tard pour les besoins de la conception logicielle par Gamma et al. [GHJV95] dont les travaux ont contribué non seulement à prouver l'intérêt du concept, mais aussi à faire de lui une référence dans le domaine de l'informatique.

Le concept de patron de conception a également été repris par les fondateurs de l'approche de l'Ingénierie de la collaboration pour proposer un nouveau langage de Patron de Conception pour la collaboration appelé *thinkLets*. Les *thinkLets* ont donc le même objectif que les autres langages de Patrons de Conception. Ils sont les meilleures pratiques des facilitateurs experts pour supporter les groupes dans leurs efforts collaboratifs pour réaliser les objectifs/buts. Les problèmes que les *thinkLets* sont censés résoudre sont les situations récurrentes pour lesquelles les *thinkLets* peuvent être utilisés systématiquement comme solutions déjà prédéfinies pour faire avancer le groupe vers son but [BKGJdV06].

Le *thinkLet* est défini comme la plus petite unité de capital intellectuel nécessaire pour créer un patron de collaboration [BdVJ03]. Les *thinkLets* sont des techniques de facilitation réutilisables, prédictibles et transférables qui peuvent être utilisées pour conduire un groupe à travers un processus vers son but. Ils peuvent donc être utilisés et réutilisés comme des blocs de construction pour les conceptions de processus de groupe dans plusieurs domaines où la collaboration est nécessaire [dVB05]. Chaque *thinkLet* est une instantiation de l'un des six patrons généraux suivants, eux aussi ayant des sous patrons [BKGJdV06] :

1. **Générer** : Ce patron permet de passer de moins de concepts à plus de concepts dans le lot de concepts partagés par le groupe.
 - *Rassembler*- Il s'agit de collecter et de partager les concepts connus provenant individuellement des membres du groupe.
 - *Créer*- Il s'agit de produire et partager les nouvelles idées qui n'étaient pas précédemment connues des membres du groupe.
 - *Elaborer*- Il s'agit d'ajouter des détails aux concepts qui sont déjà partagés par le groupe.
 - *Décomposer*- Il s'agit de caractériser un concept en termes de ses composants et sous composants.

- *étendre*- Il s’agit d’ajouter des détails pour expliquer et décrire plus complètement un concept.
2. **Réduire** : Ce patron permet de partir de beaucoup de concepts et arriver à se focaliser sur peu de concepts dont le groupe estime qu’ils méritent plus d’attention.
 - *Sélectionner*- Il s’agit de choisir un sous-ensemble des concepts existants.
 - *Abstraire*- Il s’agit de dériver des concepts plus généraux à partir d’instances spécifiques d’un ensemble existant.
 - *Résumer*- Il s’agit de capturer l’essence des concepts sans l’élimination de concepts uniques.
 3. **Clarifier** : Ce patron permet de passer de moins à plus de compréhension partagée des concepts et des mots et phrases utilisés pour les exprimer.
 - *Décrire*- Il s’agit de proposer des explications et des formulations alternatives d’un concept.
 4. **Organiser** : Ce patron permet de passer de moins à plus de compréhension des relations entre les concepts que le groupe est en train de considérer.
 - *Classifier*- Il s’agit d’arranger les concepts dans un *cluster* labellisé.
 - *Structurer*- Il s’agit de créer des arrangements spatiaux parmi les concepts pour représenter leurs relations conceptuelles.
 5. **Évaluer** : Ce patron permet de passer de moins à plus de compréhension de la valeur relative des concepts sous certaines conditions.
 - *Sonder*- Il s’agit d’évaluer l’opinion du groupe par rapport aux concepts.
 - *Ranger*- Il s’agit d’identifier un ordre de préférence parmi les concepts.
 - *Évaluer*- Il s’agit de spécifier et élaborer la valeur des concepts.
 6. **Faire le consensus** : Ce patron permet de passer de peu à plus de membres de groupe qui sont volontaires pour s’engager à une proposition.
 - *Mesurer*- Il s’agit d’évaluer le degré auquel les parties prenantes sont volontaires pour s’engager à une proposition.
 - *Diagnostiquer*- Il s’agit de chercher une compréhension des raisons sous-jacentes des dissensions.
 - *Préconiser*- Il s’agit de chercher à persuader d’autres à adopter et accepter une position.
 - *Résoudre*- Il s’agit de chercher les façons de surmonter les causes sous-jacentes des dissensions.

Comme précédemment annoncé, les patrons ci-dessus sont de patrons généraux de collaboration tandis que les thinkLets sont des patrons de collaboration qui proviennent de variations spécifiques des patrons généraux. Le choix des thinkLets est fait par les ingénieurs de collaboration sur la base des variations produites. Etant donné un thinkLet, il peut être à la fois la variation de plusieurs patrons généraux de collaboration. Par exemple, un thinkLet peut amener un groupe à générer des commentaires qui évaluent les mérites d'un ensemble de concepts, ou à organiser leurs commentaires en les plaçant dans des catégories données. Pour plus de détails sur les thinkLets, voir la référence [BdV01].

Dans la section suivante, nous présentons la manière de concevoir un processus de collaboration en s'appuyant sur les concepts que nous avons énoncés.

2.3.5 Concevoir des processus de collaboration

Concevoir et déployer des processus de collaboration pour les transférer aux praticiens constitue l'une des premières missions de l'Ingénierie de la collaboration. Les termes *conception (de l'Ingénierie de la Collaboration)*, *concevoir* et *déployer* sont aussi des concepts clés dans l'Ingénierie de la collaboration. Ils conduisent en pratique à (1) élaborer un document définissant un ensemble d'étapes structurées pour atteindre des objectifs et les conditions sous lesquelles ces étapes doivent être exécutées ; (2) créer, documenter et valider une conception ; (3) implémenter le processus de collaboration et le supporter d'une manière qu'il devienne une pratique autonome dans une entreprise.

Le travail de l'ingénieur de la collaboration consiste donc à concevoir des processus de collaboration qui sont beaucoup plus riches (structuration, organisation) que les processus traditionnels classiques faits par des facilitateurs. Etant donné le manque de qualification et d'expérience dans la facilitation, la conception faite par l'ingénieur de collaboration doit être de haute qualité et se montrer plus robuste. L'objectif est donc d'accroître la compréhension de la conception et la transmission des processus de collaboration [KdV07].

Comme précédemment annoncé, l'ingénierie de la collaboration s'inspire de l'approche des patrons de conception pour concevoir les processus de collaboration. Ces patrons, aussi appelés thinkLets, constituent aujourd'hui une librairie assez importante, soixante dix environ [KBdV⁺06]. Ce faisant, l'activité clé dans la conception de processus de collaboration est *le choix des patrons*. Cependant, comme dans tout processus de conception, nous avons les phases d'analyse du problème, d'étude des différentes solutions et du choix, de la solution et de la conception des processus de collaboration. En effet, entre l'analyse et le choix du thinkLet, il est nécessaire de décomposer la tâche en différentes étapes d'activités qui peuvent être exécutées avec les thinkLets. Après ces étapes itératives, l'agenda peut être construit et la conception peut être validée. La documentation du processus à des fins de transmission se fait en parallèle à chaque étape. La figure II.5 présente une vue de l'approche de conception de l'Ingénierie de la Collaboration.

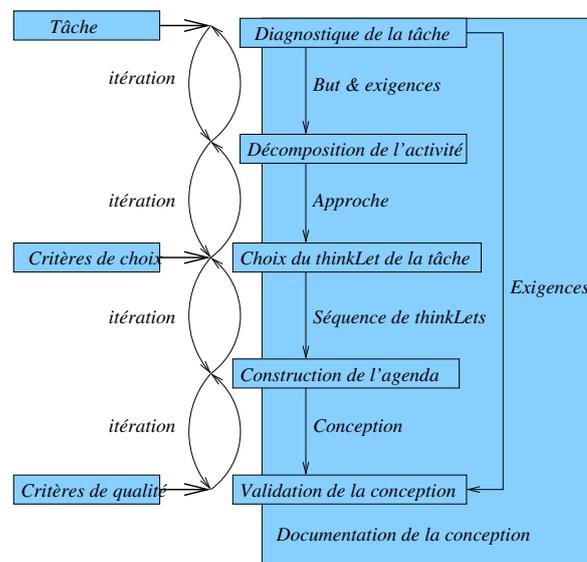


FIGURE II.5 – Approche de conception dans l’Ingénierie de la Collaboration. [KdV07]

Nous présentons ci-après les différentes étapes de la conception d’un processus de collaboration.

Diagnostic de la tâche- cette étape consiste essentiellement en la présentation des différentes parties prenantes dans le processus de collaboration ; elle définit aussi des exigences et contraintes auxquelles sont soumis le processus de collaboration et les ressources qui y sont engagées. Autrement dit, à l’issue de cette étape, les analyses suivantes sont faites : analyse de la tâche (but, livrables et objectifs), analyse des parties prenantes (groupe, parties, rôles et besoins), analyse de ressources (temps, connaissance, effort et ressources physiques), analyse des praticiens (qualifications, expérience, personnalité et domaine d’expertise). Les points précédents servent de repère pour l’analyse et la négociation des exigences et des contraintes relatives au processus de collaboration. Il est, par ailleurs, important que l’ingénieur de collaboration fasse savoir tous les détails concernant ces exigences et contraintes pour les besoins d’une instantiation future par les praticiens.

Décomposition de la tâche- une fois que les contraintes et les exigences du processus de collaboration sont connues, il reste à esquisser le processus en question. Ce travail commence par la décomposition de la tâche en différentes activités. Cela peut être inspiré des processus traditionnels de l’organisation s’il en existait avant. Dans le cas contraire, il faut d’autres références ou partir de rien. Dans ce dernier cas, les livrables doivent être déterminés afin de définir les activités qui permettent de les obtenir. Les activités sont ensuite nommées et disposées en séquences avec une description. Les activités sont au mieux décomposées en étapes et cela se fait de deux manières possibles : la décomposition de processus et la décomposition des résultats. Dans la première, on se réfère aux différents patrons de collaboration

vus dans la section 2.3.4, et dans la deuxième nous avons, entre autres, les types de résultats suivants : Entrée (créative, informative, réflexion, ...), engagement (décision, accord, ...) et l'orientation (choix, direction, ...).

Choix du thinkLet de la tâche- le travail de décomposition a pour but de déterminer les thinkLets qui vont avec les différentes activités résultant de cette décomposition. Cette correspondance entre activités et thinkLets se fait suivant une directive précise. En effet, le choix du thinkLet doit prendre en compte, étant donnés les résultats attendus, le processus et les ressources : le but du client, l'acceptation des parties prenantes, la transférabilité aux praticiens et la réutilisabilité des ressources allouées. Selon Kolfshoten et de Vreede, une classification des thinkLets basée sur les résultats et les patrons est présentée [KdV07]. Cette classification représente également une carte aidant à faire le choix des thinkLets.

Réalisation de l'agenda- cette étape est la plus importante dans la conception car elle consiste à (voir le tableau II.1) : décrire le démarrage du processus (contacts sociaux), présenter la technologie utilisée et la visée de l'effort de collaboration, définir les arrêts, présenter les résultats des étapes précédentes s'il y en a eu, déterminer les décisions que les participants vont devoir considérer compte tenu de certains critères, récapituler la session de travail et évaluer le processus. A partir de ces informations de l'agenda, le flot de contrôle du processus est défini et représenté graphiquement à l'aide du modèle du processus de facilitation présenté par de Vreede et Briggs [dVB05] (exemple II.6, infra). Il est à noter que certains des points précités, tels que les arrêts et les présentations, sont plutôt consignés dans les prescriptions du processus.

Activité	Description	Question/ affectation	Livrable	ThinkLet et Patron	Temps
1					
2					
etc.					

TABLE II.1 – Modèle d'agenda [KdV07]

Validation de la conception- la validation de la conception du processus de collaboration se fait sur la base des informations contenues dans l'agenda réalisé auparavant. Il existe plusieurs méthodes pour faire cette validation. Il y a l'*Essai* qui consiste à faire une implémentation à une petite échelle du processus de collaboration pour permettre aux membres d'une équipe d'évaluer la qualité du processus. Une autre méthode, l'*Exploration* consiste en une évaluation finale du processus de collaboration en parcourant les activités du processus avec les praticiens et le client ou avec quelques participants. Comme méthode, citons également la *Simulation* à travers laquelle l'ingénieur de collaboration essaie de répondre aux questions qu'il se pose lui-même sur la conception et de voir si ces réponses peuvent être utilisées par la suite. Enfin, l'*Evaluation de l'expert*, est encore une méthode qui dépend des critères de qualité de ce dernier eu égard à son expérience.

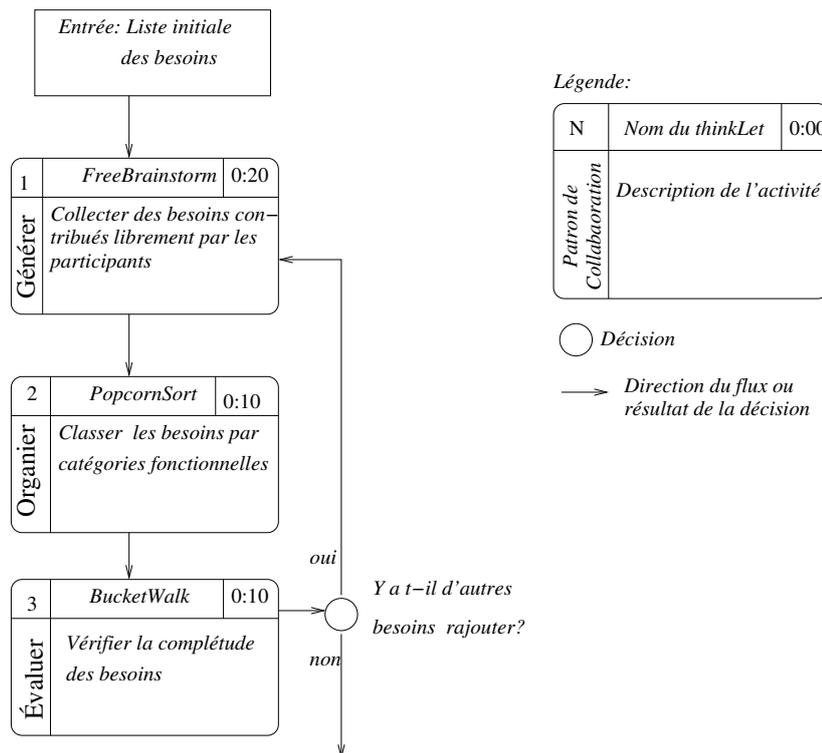


FIGURE II.6 – Une représentation graphique d’un processus utilisant des thinkLets

Documentation- un modèle de documentation de la conception du processus de collaboration a été élaboré pour faciliter l’apprentissage autonome des praticiens. Ce modèle s’appuie sur les théories cognitives [Swe88] pour proposer des moyens appropriés visant à optimiser l’apprentissage. Ce modèle propose (infra, la figure II.7 [KvdH06, KPdv06]) : (1) une présentation du processus de collaboration, la séquence des thinkLets et d’autres activités comme les arrêts et autres points pertinents ; (2) les détails nécessaires sur chaque thinkLet ; (3) la description des hypothèses pertinentes faites au moment de la conception du processus et enfin (4) les résumés des thinkLets sur des cartes servant d’aide mémoire pour les praticiens.

Une étude au cours de laquelle de nouveaux praticiens sont formés à l’utilisation du document ci-dessus a été faite par Kolfshoten et al. [KPdv06]. Le but était de détecter les difficultés liées au transfert d’un tel document en vue d’améliorer la qualité de l’approche d’apprentissage. Il en est ressorti qu’il y a un besoin de flexibilité dans l’application de la méthode car les praticiens n’ont pas souvent le même recul ou la même expérience selon les domaines. Ce dernier point constitue donc une direction pour de futurs travaux de recherche.

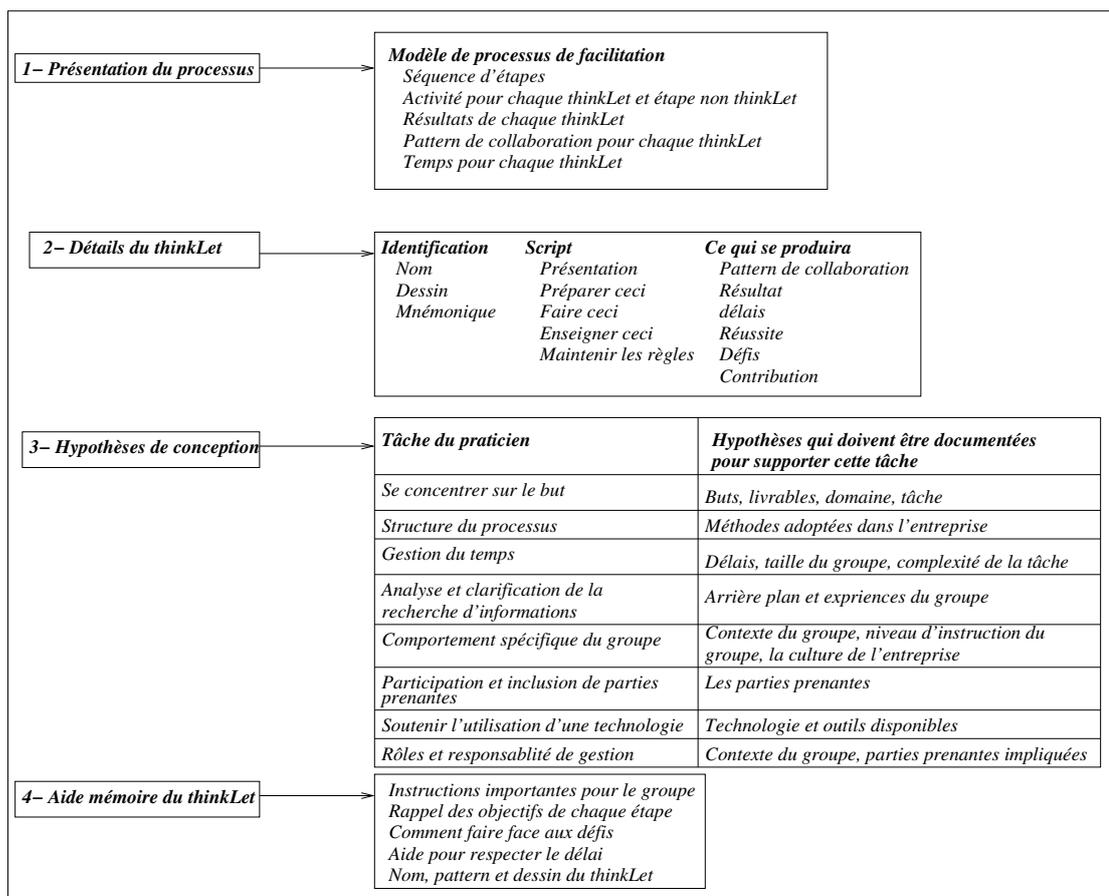


FIGURE II.7 – Modèle de documentation d'un processus de collaboration [KvdH06]

2.4 Conclusions

En explorant l'état de l'art sur la collaboration, ce chapitre nous a permis de réexaminer des notions cadres comme la coopération, la coordination et la communication qui sont souvent évoquées quand on parle de la collaboration.

Après un exposé général de ce qu'est la collaboration et les motivations qui la soutiennent, nous avons présenté quelques travaux majeurs. Ces travaux sont issus des disciplines comme l'Intelligence Collective qui s'intéresse à la manière d'accroître la qualité du travail réalisé par un groupe de travail de manière cohérente et harmonieuse comme si les membres de ce groupe ne représentaient qu'un seul individu. Pour cela, nous avons vu les différents types d'intelligences collectives (originelle, pyramidale et en essaim) avec leurs forces et leurs faiblesses. Il en est ressorti que seule l'intelligence collective originelle se prête mieux à l'émergence d'un ensemble qui englobe chaque partie et qui en même temps permet à chaque partie de s'affirmer dans sa particularité en raison des conditions dans lesquelles elle se réalise. Aussi, elle est celle qui se prête bien à notre contexte d'étude (Elicitation des Exigences) compte

tenu du nombre des participants (quelques dizaines) et aussi de l'environnement et de l'esprit de travail qui ne seront pas soumis à une hiérarchie quelconque, mais plutôt d'une complémentarité entre participants (développeur, acquéreur, client, autres parties prenantes). Nous avons également donné quelques indications sur les chemins ouverts par cette discipline (voir section 2.2).

Un autre domaine important de recherche que nous avons abordé dans cette partie est l'Ingénierie de la Collaboration, qui, elle, vise à concevoir et à déployer des processus de collaboration pour les transférer aux praticiens. En effet, L'Ingénierie de la Collaboration considère que l'étude de la collaboration peut être assimilée à une science et ce faisant, elle mérite d'être une discipline à part entière, d'où l'appellation "Ingénierie de la Collaboration". Nous expliquons en quoi cette ingénierie peut être faite, quels sont ses fondements théoriques et conceptuels, et enfin comment on réalise une conception dans ce domaine. Quelques travaux importants réalisés sur l'Ingénierie de la Collaboration ont été présentés ainsi que les chantiers ouverts à ce jour.

Compte tenu d'une proximité au niveau conceptuel sur la collaboration, notre approche qui est présentée dans le chapitre III s'inspire de l'Ingénierie de la Collaboration.

Proposition d'un modèle pour la collaboration

Puisque le contexte de nos travaux est l'Ingénierie des Exigences et plus particulièrement l'Elicitation des Exigences, nous replaçons l'étude de la collaboration dans ce chapitre. Nous présentons notre approche de recherche sur l'Elicitation Collaborative des Exigences, puis une analyse expérimentale.

3.1 Introduction

Nous avons commencé notre réflexion à partir des processus définis par la norme EIA-632 comme montré sur la figure III.1, où nous avons remarqué une forte collaboration dans la réalisation d'un système. Notre attention a surtout été attirée sur les processus techniques et plus particulière au niveau de la conception et de la réalisation en raison de notre contexte d'étude.

Sur la figure III.1, les processus d'ingénierie sont distincts tandis que les processus de collaboration sont implicites. De ce fait, la distinction entre les activités d'ingénierie et les activités de collaboration n'est pas visible directement. En d'autres termes, nous considérons initialement que les aspects d'ingénierie et de collaboration ne sont pas distincts comme illustré en (a) sur la figure III.2 proposée dans des travaux que nous avons présentés dans un article [KS07]. Cependant, puisque l'une de nos questions de recherche (Question de Recherche 3, en I à la page 5) porte sur l'amélioration de l'efficacité du processus d'Elicitation des Exigences par la collaboration, il convient de traiter conjointement les deux facettes du problème, à savoir : le défi de la recherche de l'efficacité dans l'élicitation des exigences et en quoi la collaboration peut y contribuer. De ce fait, ayant séparé les deux aspects, nous avons estimé qu'il est nécessaire d'organiser et de structurer le travail collaboratif qui, d'une part, est dirigé par des processus dits *processus de collaboration*, et d'autre part nous conduit à distinguer les tâches d'ingénierie qui sont elles aussi dirigées par des processus appelés *processus d'ingénierie* comme montré en (b) sur la figure III.2. Cependant, puisque la collaboration se produit autour de tâches d'ingénierie, la collaboration dépend donc de celles-ci, c'est-à-dire qu'elle dépend des processus d'ingénierie. Cela conduit à l'étape d'intégration des deux types de processus en (c) où les processus de collaboration sont conçus pour les processus d'ingénierie. Cela veut dire que la collaboration n'a de signification que lorsqu'elle est prise dans un contexte particulier qui est celui du travail à réaliser. En effet, on parle

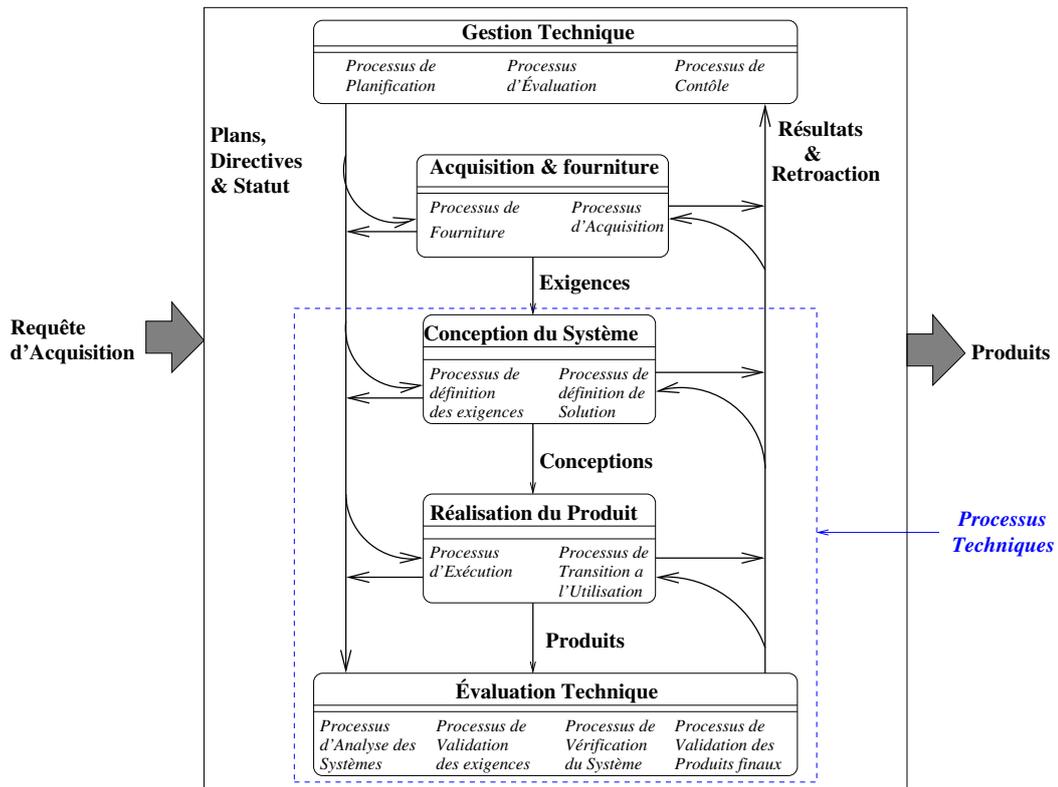


FIGURE III.1 – Processus de l’EIA-632.

souvent de la collaboration mais sans fixer les limites des deux champs (collaboration et travail à réaliser) dans le but de les exprimer clairement.

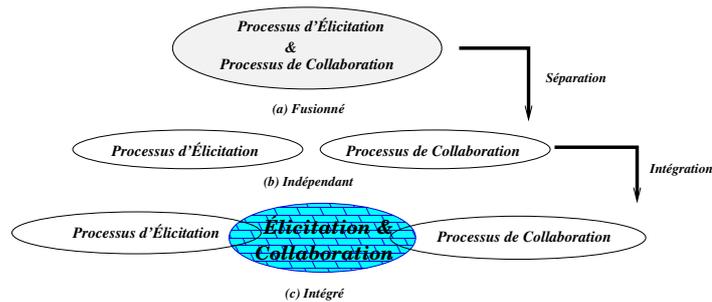


FIGURE III.2 – Indication du problème [KS07]

Sur la base de cette remarque, notre domaine d’étude est donc le domaine à l’intersection du domaine des *Processus d’Élicitation* et de celui des *Processus de Collaboration*.

Ainsi, nous réalisons la séparation des problèmes en considérant cette fois-ci qu’ils ne sont pas complètement disjoints. Pour ce faire, nous choisissons de réaliser les tâches d’ingénierie à travers des processus d’ingénierie qui sont déjà connus et standardisés tels que ceux de de

la norme EIA-632. Ainsi, nous ne les définissons pas parce que nous considérons qu'ils sont déjà connus et acceptés par tous [All99]. Cependant, les processus de collaboration qui leur sont associés doivent être identifiés et définis. Nous nous référons à l'approche de l'Ingénierie de la Collaboration présentée dans le chapitre 2.3 pour définir les processus de collaboration. Cela nous conduit à nous focaliser sur le nouveau domaine que nous avons défini, à savoir la partie de la collaboration qui est couplée avec l'ingénierie. Les processus de collaboration attendus doivent donc être applicables à n'importe quel standard de l'ingénierie système car l'ingénierie de la collaboration offre des processus de collaboration réutilisables, prédictibles et transférables grâce au thinkLet. Cela a pour effet d'améliorer l'efficacité de la collaboration selon Briggs et al. [BdVJ03]. Ainsi, nous visons à tirer partie des avantages de l'Ingénierie de la Collaboration pour accroître l'efficacité de l'Elicitation des Exigences. Dans la sous-section suivante, nous proposons un modèle pour la collaboration basé sur la séparation de la logique d'Ingénierie des Exigences et la logique de la collaboration.

3.2 Modèle de Collaboration

Notre vision de la collaboration peut être résumée par la notation formelle suivante. La collaboration est une fonction définie ainsi :

$$\textit{Collaboration} : (\textit{PROC}_I \times \textit{PROC}_C, \textit{CON}) \longrightarrow \textit{RESULT}$$

Où :

\textit{PROC}_I représente un sous ensemble de processus d'ingénierie qui sont essentiellement inclus dans $\textit{PROC}_{EIA-632}$,

$\textit{PROC}_{EIA-632}$ représente les processus de la norme EIA-632,

\textit{PROC}_C est un sous ensemble de processus de collaboration à définir,

\textit{CON} est un ensemble de contraintes pour un processus de collaboration,

$\textit{RESULTAT}$ est un ensemble de résultats de la collaboration.

Cette définition est la base de notre approche. En effet, nous voyons la collaboration dans le domaine de l'ingénierie comme une fonction utilisant les processus d'ingénierie avec des règles à respecter et les processus de collaboration qui décrivent les différentes interactions entre les participants et le flot de données. Ici, \textit{PROC}_I est un processus de l'EIA-632. Les conditions (préconditions et post-conditions) permettant une collaboration efficace sont définies en utilisant des contraintes auxquelles les acteurs, les compagnies et les activités sont soumis. \textit{CON} et \textit{PROC}_C sont des inconnues de la définition ci-dessus. Elles sont définies dans les sous-sections suivantes.

3.2.1 Contraintes de la collaboration : *CON*

Comme tout processus, les processus de collaboration sont soumis à des contraintes dont nous distinguons trois types : les contraintes relatives aux acteurs, les contraintes relatives aux tâches et les contraintes relatives aux ressources utilisées. Ces contraintes sont prises en compte dans la définition des règles de collaboration. Par exemple, les contraintes liées aux acteurs sont la disponibilité, les compétences, la confidentialité, etc. Les activités de collaboration dépendent de la structure de contrôle des processus d'ingénierie. Des règles relatives aux ressources de collaboration sont soumises à la disponibilité, le droit de propriété, etc. Le nombre de contraintes peut s'accroître exponentiellement si nous ne faisons pas attention aux contraintes que nous souhaitons prendre en compte.

En résumé, nous pouvons identifier les trois types de contraintes : contraintes acteurs, contraintes de tâches et contraintes de ressources. Une bonne définition des règles de collaboration est essentielle parce qu'elle garantit une bonne définition des processus de collaboration tout en n'oubliant pas la spécification des hypothèses sur ces processus.

3.2.2 Conception de Processus de Collaboration : *PROC_C*

Dans notre approche, nous définissons les processus de collaboration en deux parties. D'abord, nous définissons les interactions dans les processus de collaboration et deuxièmement nous établissons leur structure de contrôle. Nous détaillons ci-après ces deux points.

Interactions

A partir de la description précédente, nous pouvons déduire le modèle suivant qui est un modèle pour la collaboration inspiré de la norme EIA-632. Comme décrit dans la section 1.2.1 à la page 7, cette norme distingue trois types d'acteurs : le développeur, l'acquéreur, et les autres parties prenantes du système. Ces acteurs vont constituer ce qu'il convient d'appeler les *participants* aux processus de collaboration dans l'approche de l'Ingénierie de la Collaboration. Ces acteurs peuvent indiquer des individus ou des équipes de travail. La figure III.3 présente l'interaction dans notre modèle de collaboration. Nous pouvons remarquer que l'interaction se fait à travers et autour d'une liste de tâches.

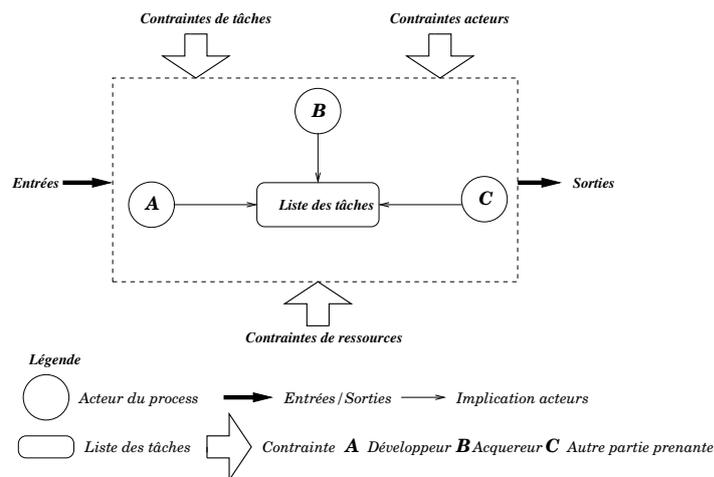


FIGURE III.3 – Modèle de collaboration pour l’EIA-632

Le rectangle en pointillés représente le processus de collaboration en charge des entrées et des sorties ; le processus est aussi soumis aux trois types de contraintes de collaboration définies. Nous appliquons ce modèle au processus d’Ingénierie des Exigences dans la section 3.4.2.

L’autre partie du processus de collaboration qui est la structure de contrôle est définie dans la section suivante.

Comportement des processus collaboratifs

Cette partie est focalisée sur les contraintes de tâches. Les interactions et les séquences de tâches sont au centre du modèle de collaboration. Dans notre approche, la structure de contrôle du processus de collaboration n’est pas identique à celle des processus d’ingénierie. Cependant, il y a une interdépendance entre les deux. La structure de contrôle définit l’ordre d’exécution des tâches. Le langage de thinkLet permet d’exprimer cet ordre une fois qu’il est fixé comme montré sur la figure III.4. Cet exemple concerne une session d’élucidation des exigences commençant par un ensemble d’exigences initiales.

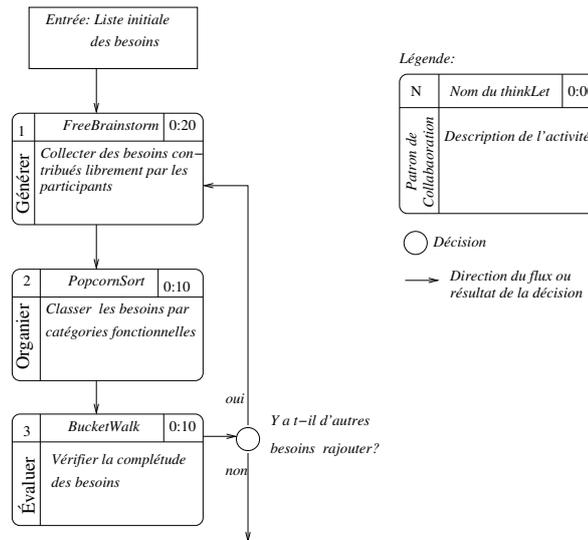


FIGURE III.4 – Structure de contrôle dans les processus de collaboration

Cette session est constituée d'activités supportées par les thinkLets appelés 'FreeBrainstorm', 'PopcornSort' et 'BucketWalk' [BdV03]. Ces thinkLets sont des variations particulières des patrons de base 'Générer', 'Organiser' et 'évaluer'.

Dans le but de formaliser ce qui précède, nous proposons un modèle conceptuel qui est décrit dans la section 3.3.

3.3 Un modèle conceptuel

Comme annoncé précédemment, nous distinguons deux types de processus : les Processus d'Ingénierie et les Processus de Collaboration. Les premiers sont déjà offerts par la norme EIA-632, ils n'ont donc pas besoin d'être redéfinis tandis que les derniers doivent être conçus et définis au moyen de l'Ingénierie de la Collaboration. Dans cet objectif, le thinkLet devient un concept fondamental dans notre modèle. La figure III.5 (infra) montre un diagramme de classes avec les différents concepts que nous utilisons :

Besoin- il est l'expression du souhait ou du problème d'un utilisateur ou de toute autre partie prenante que nous appelons ici *participant*. Le *besoin* s'exprime de manière non formelle, de façon orale ou écrite de telle sorte que celui qui l'exprime se sente à l'aise. Le besoin peut être exprimé directement par le *participant* ou extrait par une autre personne, généralement l'ingénieur des exigences qui participe à la session en tant que facilitateur pour poser des questions ou utiliser d'autres méthodes qui incitent à produire plus de besoins.

Contrainte- il s'agit de toute limitation ou obligation réglementaire à laquelle le système doit être soumis et qui peut donner lieu à des exigences système. Les exigences résultant des contraintes de ce type sont souvent des exigences non fonctionnelles liées par exemple à

la sécurité, tandis que les *besoins* des utilisateurs donnent en général lieu à des exigences fonctionnelles.

Exigence- elle est une expression formelle et technique du *Besoin* ou de la *Contrainte*. Toutefois, il faut noter que tous les besoins et toutes les contraintes ne sont pas techniquement réalisables et ne sauraient être traduits en exigences techniques. En général, on utilise les modèles (gabarit) de spécification pour formaliser les besoins ce qui donne lieu à des exigences qui sont techniquement satisfaisables. La transformation des besoins en exigences est nécessaire car seules les exigences sont exploitables par l'équipe de réalisation. Les exigences ont un identifiant, un libellé et un niveau de priorité défini en fonction du type de système à concevoir.

Cahier de charges- il représente un document dans lequel sont rédigées les spécifications des *exigences* selon les normes du domaine. En effet, selon que l'on soit dans l'aéronautique ou dans les systèmes d'information, les exigences ne sont pas spécifiées de la même manière. Outre les différences de spécification, tout *cahier des charges* doit être organisé en *Catégories* et *sous Catégories* si nécessaires. Cela permet d'accroître la facilité d'utilisation du document. Le *cahier des charges* est le document qui est transmis à l'équipe de développement pour la conception et la réalisation du système final. La rigueur doit être de mise pour la rédaction du *cahier des charges* car la réussite du projet dépend de sa qualité en terme de structuration et de clarté.

Catégorie- elle représente une sous partie du document de spécification, le *cahier des charges* qui correspond généralement à un type d'exigences donné. En effet, les exigences sont de différents types ; Elles peuvent être classifiées en exigences de type *fonctionnel* et de type *non fonctionnel*. Ces deux catégories peuvent aussi avoir des sous catégories. Cette organisation en *catégories* et *sous catégories* a pour avantage d'accroître la lisibilité, la clarté et la compréhension du document.

Processus d'Ingénierie- les *exigences* sont définies suivant des processus qui sont établis par des normes, nous avons retenu ici la norme EIA-632. Ces normes décrivent en détail les différentes étapes à suivre pour définir les exigences.

Tâche- les étapes définies par le *processus d'ingénierie* correspondent à l'exécution d'une *tâche* particulière. Puisque nous sommes dans un contexte collaboratif, il s'agit ici de tâches collaboratives, celles qui sont menées conjointement par les participants.

Activité- chacune des *tâches* est composée à son tour d'*activités*. Les tâches sont souvent complexes et nécessitent d'être décomposées en plusieurs activités.

ThinkLet- il est le support des *activités* des *tâches* collaboratives. Le *thinkLet* encapsule les informations qui décrivent comment exécuter une tâche de manière collaborative. Cette

description représente un des composants du ThinkLet qui est appelé Script dans la terminologie de l'Ingénierie de la Collaboration (infra, l'annexe 1). Puisqu'un thinkLet est un patron de conception, il peut être adapté selon la situation en changeant le script, tout en gardant les principes fondamentaux de la technique collaborative. Il est également possible de proposer un nouveau ThinkLet lorsqu'il n'existe pas encore un thinkLet adapté à la situation en cours. Pour plus de détails, se reporter à la section 2.3.4 à la page 46.

Processus de Collaboration- il s'agit du processus élaboré par un ingénieur de la collaboration pour les participants à une session d'Elicitation des Exigences par exemple. Ces processus sont construits à partir de *thinkLets* et ils impliquent au moins trois *Participants*. Les processus de collaboration sont documentés par l'ingénieur de la collaboration et contiennent toutes les informations nécessaires permettant à une équipe de réaliser un travail collaboratif de manière autonome, voir la section 2.3.5 à la page 48.

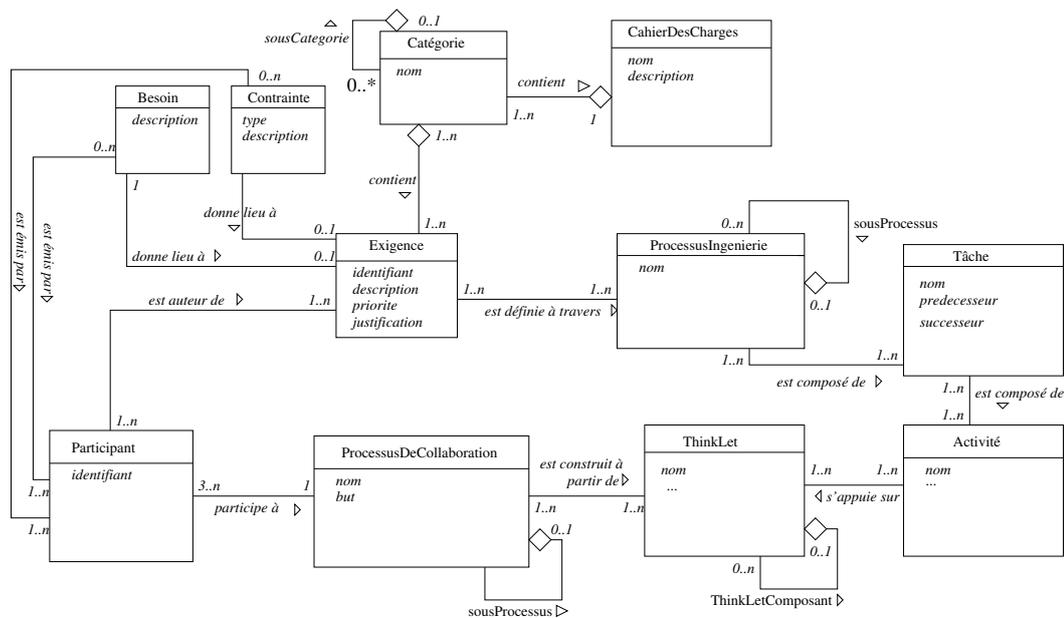


FIGURE III.5 – Un modèle de classes des processus d'ingénierie et de collaboration [Kon08]

Le Modèle ci-dessus, que nous avons proposé dans un de nos contributions [Kon08], formalise la phase d'intégration entre les deux types de processus en faisant le lien entre les tâches et les thinkLets suivant le modèle proposé dans la section 3.1.

Comme précédemment annoncé, nous avons choisi l'Ingénierie des Exigences comme contexte d'étude et plus spécifiquement la phase d'Elicitation des exigences. La section 3.4 présente la situation et décrit les différents processus qui interviennent dans cette phase et aussi un diagramme d'interaction UML [RV04], un des objectifs étant l'implémentation d'un outil permettant de supporter le processus d'élicitation collaborative des exigences.

3.4 Environnement de Travail Collaboratif pour l'Elicitation

3.4.1 Modélisation des Acteurs Collaboratifs

Ici, nous illustrons notre approche avec les processus d'Ingénierie des Exigences (voir la section 1.2.3 à la page 11). La figure III.6 met en évidence le "Processus de collaboration pour l'ingénierie des exigences" P , avec ses quatre sous processus P_1 , P_2 , P_3 , P_4 . Ces sous-processus représentent respectivement le "Processus de collaboration pour l'identification et la collecte des besoins", par exemple le *Brainstorming* [Os48] et le *Storytelling* (construction d'une histoire collective) [AG06]; le "Processus de transformation des besoins en exigences techniques", par exemple le *template* de spécification Volere [RR07]; le "Processus de collaboration pour la validation des exigences" et le "Processus de collaboration pour l'enregistrement des exigences" [All99].

Dans ce contexte, le processus de validation vise à assurer la conformité de la définition des exigences par rapport aux besoins initiaux des utilisateurs. Le processus d'enregistrement consiste à mettre par écrit les exigences spécifiées soit sur papier, soit en utilisant un outil automatisé telsque TrueReq [Tru08]. Les acteurs impliqués dans les processus sont indiqués par les flèches partant des acteurs et dirigées vers les tâches du processus. Les flèches dirigées vers le bas montrent le sens du flot de données (entrées et sorties). En particulier, les entrées sont des besoins de l'équipe des acquéreurs (EA) et de l'équipe des autres parties prenantes (APP).

Ces besoins sont identifiés et collectés en collaboration avec l'équipe des ingénieurs des exigences (EIE).

Les besoins résultant du premier processus (P_1) sont utilisés comme entrée pour le deuxième processus (P_2) pour être transformés en exigences techniques. A la troisième étape (P_3), les exigences techniques sont validées par l'ensemble des acteurs pour ensuite être enregistrées par l'équipe d'ingénieurs des exigences (P_4). Le résultat du processus P est un document contenant l'ensemble des exigences spécifiées et validées par tous les acteurs. Ce document s'appelle le *Cahier des charges*.

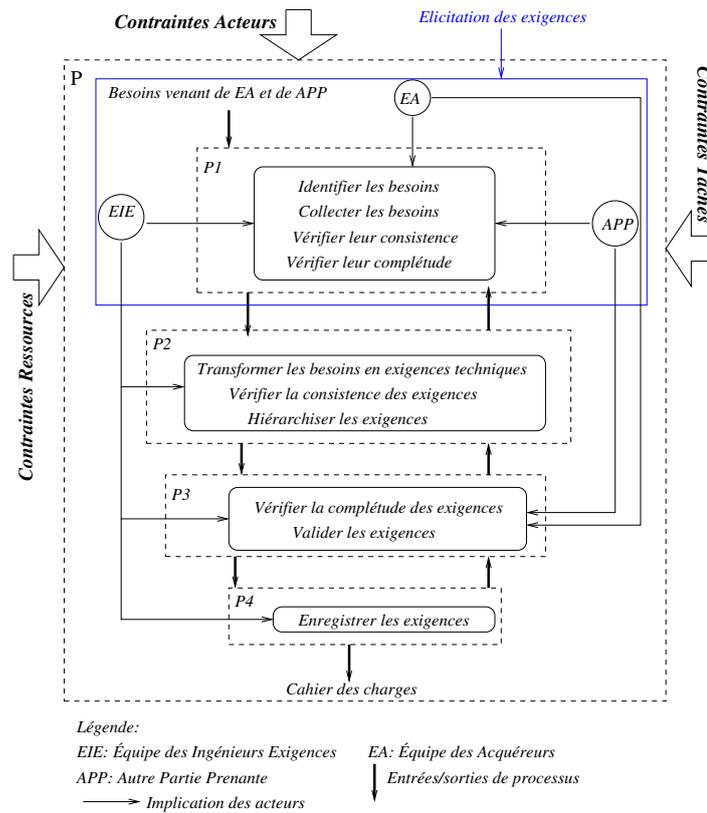


FIGURE III.6 – Processus de Collaboration pour l'Ingénierie des Exigences

3.4.2 Modèle d'Interaction avec UML

La figure III.7 (a) montre les acteurs et la figure III.7 (b) les activités à accomplir dans le processus d'Ingénierie des Exigences (IE). Nous utilisons le diagramme d'interactions pour décrire comment les acteurs interagissent dans les processus collaboratifs de l'IE. Comme le diagramme des cas d'utilisation, le diagramme d'interaction est un diagramme de haut niveau car ils ne montrent pas tous les détails des activités de l'Ingénierie des Exigences.

Dans la terminologie de l'Ingénierie de la collaboration, *EA*, *APP* et *EIE* sont des participants d'une session collaborative. Le rôle joué par *EA* et *APP* est celui du praticien et *EIE* peut être à la fois praticien et facilitateur. Dans le cas où *EIE* est uniquement praticien, les processus de collaboration sont conçus et transférés par un ingénieur de collaboration aux participants de la session. Ici, nous considérons le cas où *EIE* est à la fois praticien et facilitateur pour le processus d'Elicitation des Exigences (voir le processus P1 à la figure III.6).

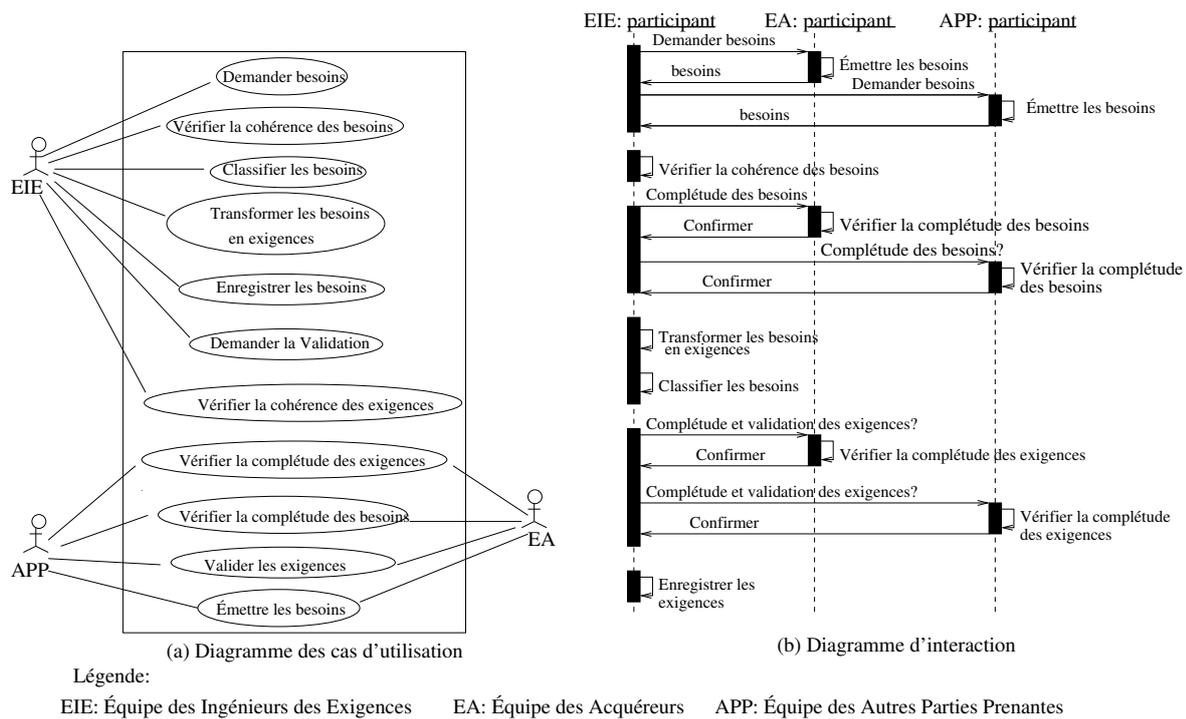


FIGURE III.7 – Diagrammes de Cas d'Utilisation et d'Interaction de haut niveau pour le processus d'Ingénierie des Exigences

Sur la base du modèle présenté dans le présent chapitre, nous proposons dans le chapitre IV une méthodologie pour l'exécution de l'Elicitation Collaborative des Exigences visant à répondre à nos questions de recherche. Nous proposons ensuite des expérimentations de l'approche que nous avons réalisées à travers des études de cas.

Vers une méthodologie pour l'Elicitation Collaborative des Exigences

4.1 Présentation générale

Dans cette partie, nous présentons notre méthodologie en plusieurs étapes avec une illustration qui porte sur la conception d'une nouvelle bibliothèque en ligne. La méthodologie que nous proposons comporte six (6) étapes comme le montre la figure IV.1. Chacune de ces étapes est présentée en détail dans une sous-section appropriée. Les étapes 2, 3 et 4 correspondent à l'exécution des activités prescrites pour l'exécution des processus de conception du système de la norme EIA-632 : *Processus de Définition des Exigences* et *Processus de Définition de la Solution*, sixième et septième processus de la norme. L'étape 5 est également inspirée de la norme EIA-632 qui décompose un système en une structure hiérarchique (système, sous-système, sous-système de sous-système, etc.). Notre étape de hiérarchisation des exigences s'appuie sur le même principe en considérant les différents modules du système et les catégories d'exigences qui leur sont associés. L'étape finale, la sixième, correspond à l'un des processus d'évaluation technique : *Processus de validation des exigences*, onzième processus de la norme.

Il est à noter que les compétences de facilitation sont encapsulées dans le thinkLet dans le but de rendre tout participant capable de faciliter une session sans être un professionnel de la facilitation. L'autonomie dont il est question ne vise donc pas la suppression du rôle du facilitateur dans une session. En revanche, cette autonomie se réfère à la faculté qu'a tout participant d'être un facilitateur. La méthodologie est ainsi présentée dans ce chapitre. L'illustration de chaque phase est également présentée.

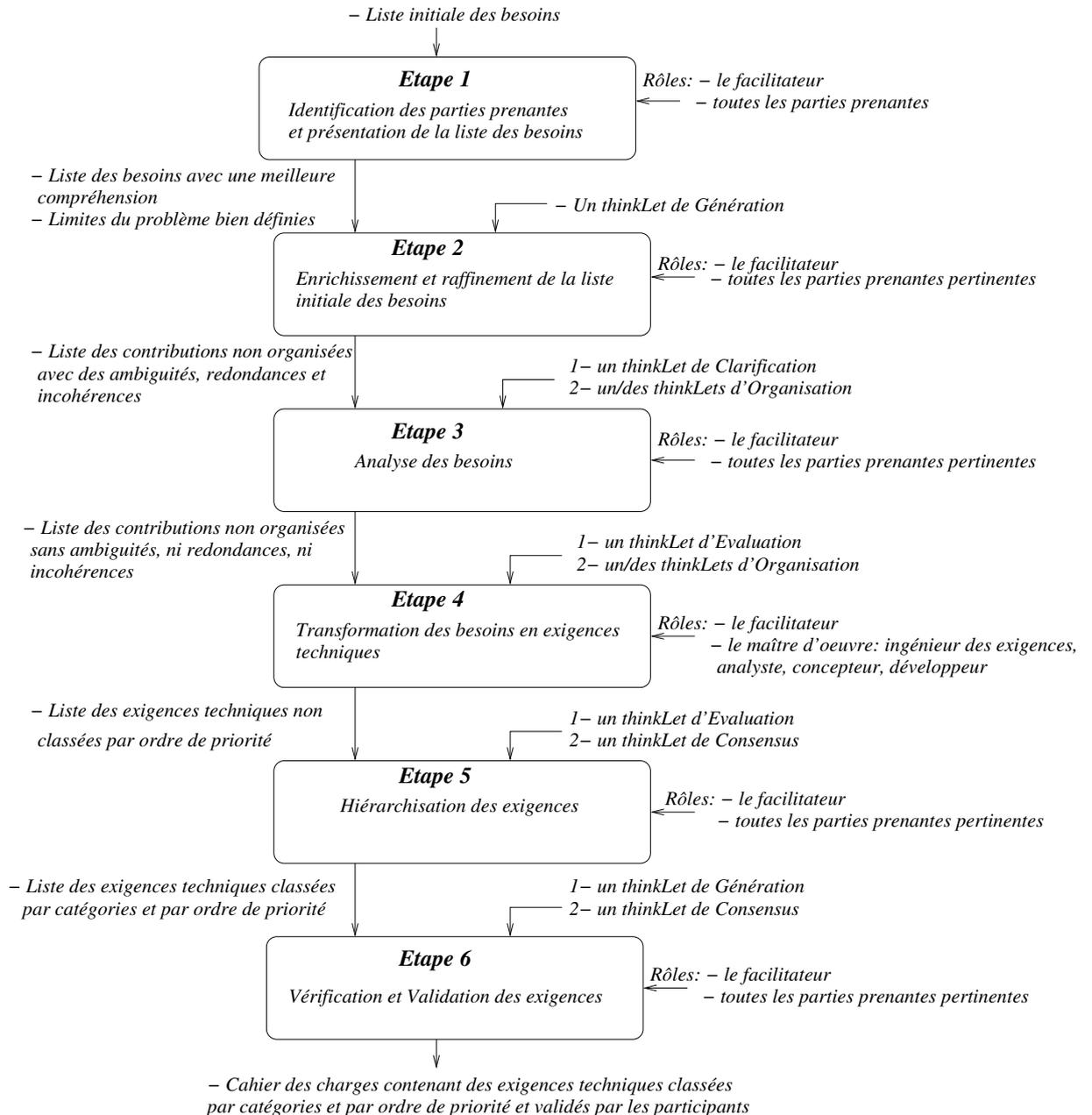


FIGURE IV.1 – Les étapes de la méthodologie.

4.2 Identification des parties prenantes pertinentes et présentation de la liste initiale des besoins

Entrée :

1. Des participants
2. Une liste d'exigences initiales

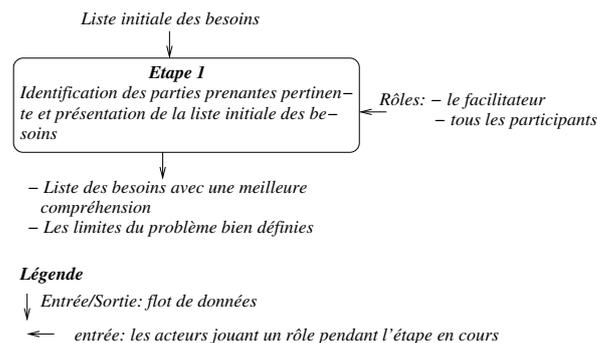


FIGURE IV.2 – Etape d'identification des parties prenantes.

Au cours de cette étape, le facilitateur identifie les parties prenantes à la conception et à la réalisation du système, leur affecte des rôles qu'elles doivent jouer au cours de la session. Il s'agit des utilisateurs finaux, des clients, de l'équipe technique composée d'analystes, de concepteurs et de développeurs, puis de toute autre personne concernée de près ou de loin par le système. En d'autres termes, n'est partie prenante que celui/celle qui a des compétences requises dans le processus d'Elicitation Collaborative des Exigences.

Après l'identification des parties prenantes, le facilitateur anime une session de discussion d'environ trente (30) minutes pour présenter les différentes facettes du problème en question. Cela se fait en distribuant une liste des points à traiter qui correspond à la liste initiale des besoins du système qui est en général établie par l'ingénieur des exigences. Cette liste permet aux participants d'avoir une définition identique du problème pour qu'ils commencent sur la même base. Au cours de cette phase, le facilitateur veille à ce que les limites du problème soient définies et connues de toutes les parties prenantes afin de mieux cibler le problème. Tout le monde peut poser des questions ou donner des réponses pendant cette phase en vue d'une meilleure compréhension du contexte et des contours du problème.

Le facilitateur est ici un spécialiste de la collaboration qui n'est pas nécessairement un ingénieur des exigences. Son rôle consiste à faire collaborer efficacement les différentes parties prenantes pertinentes. Par contre, il est tenu de connaître un tant soit peu le problème à la résolution duquel il participe. Par exemple, s'il s'agit de définir des exigences pour une bibliothèque en ligne, le facilitateur doit savoir ce qu'un tel système est censé offrir à ses utilisateurs comme service de base. Pour ce qui est des autres exigences plus pointues, la

collaboration avec les ingénieurs des exigences permet de mieux couvrir ces niveaux.

Sortie :

1. Toutes les parties prenantes identifiées
2. Liste initiale des besoins avec une meilleure compréhension
3. Les limites du problème sont définies

Illustration

Le facilitateur anime la session en commençant par une phase de présentation des différents participants. Cette phase a pour objectif de permettre aux membres du groupe de se connaître tout en créant une atmosphère plus détendue et plus conviviale et cela avant d'identifier les différentes parties prenantes. Puis, le facilitateur présente aux différents participants le contexte du problème, la vision ou les perspectives et le positionnement actuel du projet. Dans l'exemple de la bibliothèque il s'agit des différents points suivants :

Etape 1
<p>Contexte et expression des besoins</p> <p><i>La bibliothèque de l'université de Toulouse BibToulouse a décidé récemment de doter sa bibliothèque d'un système de réservation d'ouvrages et de gestion des prêts en ligne pour ses usagers. Les rayons sur le site sont diverses : Informatique, Biologie, Physique, Médecine, etc. La bibliothèque BibToulouse assure aussi la distribution en diverses langues d'une large sélection d'ouvrages.</i></p> <p><i>L'objectif fondamental du futur site www.BibToulouse.com est de permettre aux utilisateurs (étudiants, enseignants, personnel administratif et technique) de rechercher des ouvrages par thème, auteur, mot-clé, etc., de se constituer un panier virtuel, de pouvoir les réserver en ligne sur le Web en vue d'un prochain emprunt d'une part, et d'autres parts de pour faire l'administration et la maintenance du site.</i></p>
<p>Vision du projet : élicitation des exigences</p> <p><i>L'objectif de ce travail est de collecter, analyser et définir les besoins de haut niveau et les caractéristiques du futur site Web www.BibToulouse.com. Il se concentre sur les fonctionnalités requises par les utilisateurs, et sur la justification des exigences.</i></p>
<p>Positionnement</p> <p><i>Le site www.BibToulouse.com se veut être un site à la hauteur des attentes de ses utilisateurs tout en leur facilitant la procédure des prêts.</i></p> <p><i>Le but du projet consiste à :</i></p> <ul style="list-style-type: none"> - Réduire au maximum les difficultés liées à la recherche d'ouvrages dans les bibliothèques - Automatiser au plus la procédure de réservation des ouvrages - Réduire au plus les pertes d'ouvrages dans les bibliothèques - Réduire au maximum la charge de travail qui revient au personnel de la bibliothèque

TABLE IV.1 – Phase de préparation

Une fois que le problème est situé, le facilitateur procède à l'identification des parties prenantes et des rôles qui leur sont affectés. Pour le cas de la bibliothèque, les différentes parties prenantes sont :

- Le maître d'ouvrage qui est la bibliothèque de l'université de Toulouse Biblio-UT : représenté par son personnel administratif et technique qui seront une classe particulière d'utilisateurs du système (administration et maintenance du système),

- Les usagers de la bibliothèque qui sont une autre classe d'utilisateurs du système : étudiants, enseignants et chercheurs,
- Le maître d'œuvre représenté par l'équipe de conception et de réalisation du système.

Le maître d'ouvrage qui est l'acquéreur du système fait partie de la classe des utilisateurs en ce sens qu'il utilise le système pour l'administration des ressources et pour faire, par exemple, des mises à jour portant sur les ouvrages de la bibliothèque.

Les usagers sont également une autre classe d'utilisateurs essentiellement intéressés par la recherche, la réservation et le prêt des ouvrages.

Le maître d'œuvre est l'ensemble des parties prenantes qui s'occupent de l'établissement d'un cahier de charges pour le futur système, de sa conception et de sa réalisation. L'ingénieur des exigences, l'analyste, le concepteur et le développeur font partie de cette classe de parties prenantes.

Besoins fonctionnels	Besoins non-fonctionnels
Recherche	Qualité
<ul style="list-style-type: none"> - L'utilisateur doit trouver l'ouvrage recherché le plus rapidement possible - Le système offrira plusieurs méthodes de recherche : par titre, auteur, etc. 	<ul style="list-style-type: none"> - L'ergonomie doit être sobre - Il faut un formulaire de demande simple - Une aide en ligne efficace est nécessaire
Découverte	Performance
<ul style="list-style-type: none"> - Tout ouvrage est présenté sur sa propre page - Toute page pour un ouvrage comporte une image de cet ouvrage - Toute page pour un ouvrage contient la table des matières détaillée de l'ouvrage 	<ul style="list-style-type: none"> - Le système supportera jusqu'à 100000 usagers au moins - Une recherche d'ouvrage ne dépassera pas 30 secondes
Sélection	
<ul style="list-style-type: none"> - L'utilisateur peut sélectionner les ouvrages les uns après les autres - Les étudiants sont limités à trois ouvrages 	
Prêt	
<ul style="list-style-type: none"> - L'utilisateur peut accéder au formulaire de prêt à tout moment - Le numéro d'identifiant doit être enregistré au moment du prêt 	

TABLE IV.2 – Liste initiale des besoins du système Bib-UT

Une fois que ces différentes parties prenantes sont identifiées ainsi que leurs rôles, le facilitateur présente alors la liste initiale des besoins. Un exemple de liste est celle présentée dans le tableau IV.2. Cette liste est organisée en catégories et sous catégories pour faciliter la présentation et la compréhension du problème et du travail demandé. Cela a pour but de faciliter ultérieurement l'organisation et la classification des exigences finales bien qu'il sera possible de créer de nouvelles catégories et sous catégories. La liste peut être vide selon les cas ou ne contenir qu'une liste de catégories. Une fois cette liste présentée et après quelques discussions qui visent à mieux cerner le problème, le facilitateur conduit le groupe à l'étape suivante qui correspond à l'enrichissement et au raffinement de la liste initiale des besoins.

4.3 Evolution et affinement de la liste des besoins

Entrée :

1. Le facilitateur et toutes les parties prenantes identifiées
2. Liste initiale des besoins avec une meilleure compréhension
3. Les limites du problème sont définies
4. Un thinkLet de génération

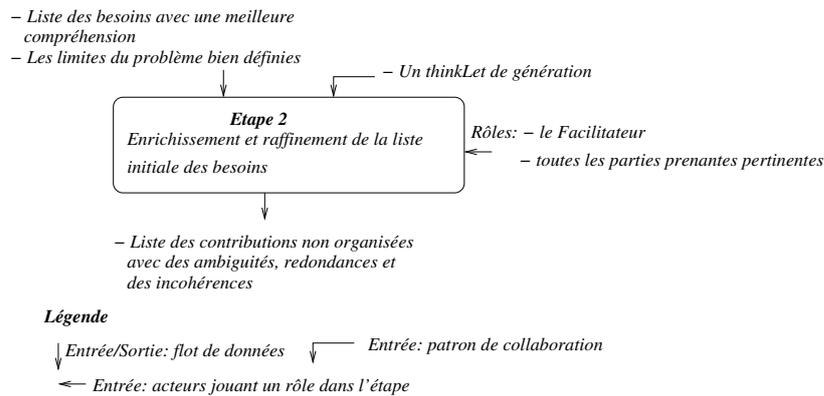


FIGURE IV.3 – Etape d'évolution de la liste initiale des besoins.

Une fois que les parties prenantes sont identifiées, que la liste initiale des exigences est connue et que les contours du problème sont également cernés par tous, les participants peuvent donc commencer à enrichir la liste initiale des besoins. Cet enrichissement se fait en fournissant de nouveaux besoins et en élaborant les besoins déjà proposés. Cette phase est très libre car tout participant peut contribuer comme il le souhaite et de manière anonyme. L'anonymat a pour avantage d'éviter que les contributions des uns et des autres ne soient considérées ou rejetées compte tenu de la personnalité de leurs auteurs [Bri06]. Cette phase est très importante car elle permet aussi l'émergence de nouvelles idées qui peuvent être créatrices ou innovatrices. Les utilisateurs et les parties prenantes contribuent librement en faisant connaître leurs "besoins". Pour un besoin, il n'existe pas un format imposé, il peut s'exprimer de plusieurs manières pourvu qu'il exprime quelque chose de la part de l'émetteur. Cette étape commence à prendre fin quand les participants sont à court d'idées. Cela s'exprime de différentes manières. A ce moment, le facilitateur intervient pour terminer la session.

Puisqu'on utilise un thinkLet de génération pour soutenir les tâches de cette activité, il faut un système support de groupe (GSS) ayant les fonctionnalités suivantes :

- F1) un éditeur de texte offrant un espace pour écrire des contributions avec des numéros d'identifiant pour chaque idée contribué,
- F2) une session collaborative permettant à tous les participants de se connecter anonymement de sorte que les contributions de tous soient visibles et accessibles par tous,

F3) un droit spécial est offert au facilitateur de la session qui peut non seulement contribuer comme les autres participants, mais il peut aussi supprimer des contributions lorsqu'elles ne contribuent pas à faire avancer le travail,

F4) le facilitateur doit avoir aussi le droit d'inviter un nouveau participant à la session en lui envoyant un email et le droit d'exclure un participant pendant la session si cela s'avère nécessaire.

Sortie :

1. Une liste de besoins non organisés avec des redondances, des ambiguïtés et des incohérences.

Illustration

Une fois que la liste initiale des besoins est présentée et comprise par tous les participants qui connaissent désormais leurs rôles, ils sont prêts à enrichir cette liste soit en contribuant à de nouveaux besoins, soit en élaborant des besoins déjà existants ou proposés par d'autres participants.

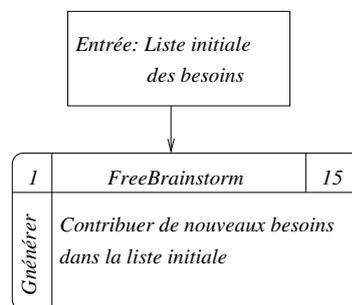


FIGURE IV.4 – étape de l'enrichissement de la liste initiale des besoins

Le premier thinkLet que nous utilisons est un thinkLet de génération qui permet aux participants de produire de nombreuses idées qui sont des besoins. Le thinkLet de génération choisi est le "FreeBrainstorm". Le facilitateur conduit le groupe dans une activité de génération et de création de nouveaux besoins en suivant les instructions inscrites dans le script du thinkLet selon les fonctionnalités de l'outil automatisé dont le groupe dispose avec une configuration particulière. Concrètement, dans notre cas présent, il faut un outil offrant un éditeur coopératif permettant à plusieurs participants d'écrire leurs contributions sur une page que tout le monde voit en même temps. Les contributions des uns et des autres sont anonymes et libres. Le tableau IV.3 montre un exemple du résultat d'une telle activité. Les points en italique dans le tableau sont de nouvelles contributions à la fin du "FreeBrainstorm". Nous pouvons remarquer l'ajout de contraintes liées au système qui sont une sous-catégories de la

catégorie des exigences non fonctionnelles.

L'outil doit permettre au participant de naviguer à travers les différentes catégories et sous catégories et de contribuer à de nouveaux besoins en sa convenance de façon anonyme et libre.

Etape 2

Besoins fonctionnels	Besoins non-fonctionnels
Recherche	Qualité
<ul style="list-style-type: none"> - L'utilisateur doit trouver l'ouvrage recherché le plus rapidement possible - Le système offrira plusieurs méthodes de recherche : par titre, auteur, etc. - <i>Des précisions sont nécessaires pour les références</i> - <i>Les résultats de la recherche seront disponibles dans une page particulière</i> 	<ul style="list-style-type: none"> - L'ergonomie doit être sobre - <i>La mise en page du site doit faciliter la démarche à l'aide d'une présentation claire et intuitive</i> - <i>La présentation du site doit être simple pour éviter trop d'efforts</i> - Il faut un formulaire de demande simple - <i>L'effort pour remplir un formulaire doit être minimal</i> - Une aide en ligne efficace est nécessaire - <i>L'utilisateur peut accéder à l'aide en ligne à tout moment</i> - <i>Une visite guidée fera partie de l'aide</i>
Découverte	Performance
<ul style="list-style-type: none"> - Tout ouvrage est présenté sur sa propre page - Toute page pour un ouvrage comporte une image de cet ouvrage - Toute page pour un ouvrage contient la table des matières détaillée de l'ouvrage - <i>Des extraits de chapitres d'un ouvrage seront accessibles sur la page de cet ouvrage</i> - <i>La page d'un ouvrage présente aussi la quantité disponible pour le prêt</i> 	<ul style="list-style-type: none"> - Le système supportera jusqu'à 100000 usagers au moins - Une recherche d'ouvrage ne dépassera pas 30 secondes - <i>Le système supportera au moins 5000 connexions simultanées</i> - <i>Le catalogue contiendra au moins 600000 ouvrages</i>
Sélection	Contraintes
<ul style="list-style-type: none"> - L'utilisateur peut sélectionner les ouvrages les uns après les autres - Les étudiants sont limités à trois ouvrages - <i>Le nombre d'ouvrages à emprunter est limité à six pour les enseignants et les chercheurs</i> - <i>L'utilisateur a la liberté de mettre un ouvrage dans le panier ou de l'enlever</i> 	<ul style="list-style-type: none"> - <i>Le système permet à l'utilisateur de faire des mises à jour</i> - <i>Le système permet aux usagers de mettre à jour leurs données personnelles</i> - <i>Les enseignants peuvent faire la mise à jour de leurs données personnelles</i> - <i>Les étudiants pourront accéder à leurs données personnelles pour les modifier</i>
Prêt	
<ul style="list-style-type: none"> - L'utilisateur peut accéder au formulaire de prêt à tout moment - Le numéro d'identifiant doit être enregistré au moment du prêt - <i>Les coordonnées de l'utilisateur sont disponibles dans le formulaire de prêt</i> - <i>un étudiant peut renouveler un prêt deux fois de suite au maximum</i> - <i>un enseignant peut renouveler son prêt au moins une fois de suite</i> - <i>tout usager peut renouveler son prêt jusqu'à trois fois de suite</i> 	

Légende	
Symboles	Significations
- Besoin	désigne un besoin préexistant
- <i>Besoin : description</i>	désigne un nouveau besoin ou une reformulation

TABLE IV.3 – Résultat du Brainstorming libre

4.4 Analyse des besoins

Entrée :

1. Une liste de besoins non organisés avec des redondances, des ambiguïtés et des incohérences.
2. Un thinkLet de clarification
3. Un/des thinkLet(s) d'organisation

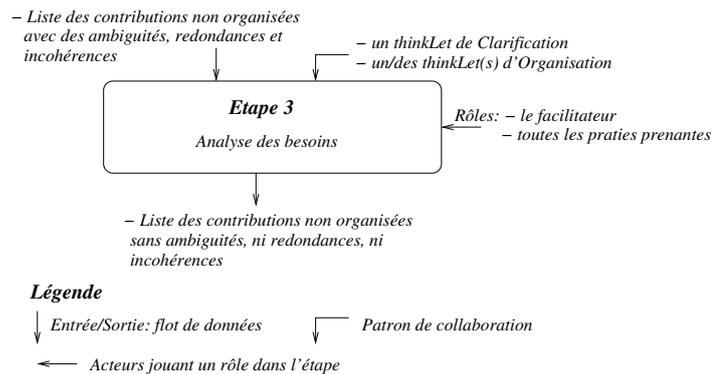


FIGURE IV.5 – Etape d'analyse des besoins.

4.4.1 Suppression des ambiguïtés

Un besoin ambigu est un besoin ayant plusieurs interprétations possibles. Ce besoin peut être clair pour l'utilisateur, mais ambigu pour d'autres parties prenantes [Buj05]. Par conséquent, à partir de la liste des besoins provenant de l'étape précédente, on identifie les besoins qui ne sont pas clairs pour tous dans le but de les clarifier. Pour chaque besoin, chaque participant fait savoir s'il a compris ou non. On les exprime sous des formes plus compréhensibles en tenant compte des justifications et des perspectives derrière chaque besoin. Pour plus de détails, merci de vous reporter à l'illustration (infa, page 78)

¶item le système permet aux personnels de la bibliothèque de mettre à jour les données relatives aux ouvrages.

4.4.2 Suppression des redondances

On dit qu'il y a redondance de besoins lorsqu'au moins deux besoins utilisent des mots différents pour traiter les mêmes sujets ou pour exprimer la même chose.

Pour supprimer les redondances, on identifie les contributions semblables ou identiques dans le but de les fusionner. Pour ce faire, il est important que chaque contribution ait une "justification" et une "perspective" clairement identifiées. La ressemblance des contributions se mesurera soit par l'analyse sémantique des termes qu'elles utilisent, soit par la comparaison de leurs justifications et perspectives. Par exemple considérons les contributions suivantes :

- Le système permet aux usagers de mettre à jour leurs données personnelles.
- Les enseignants peuvent faire la mise à jour de leurs données personnelles.
- Les étudiants pourront accéder à leurs données personnelles pour les modifier.

Les trois contributions ci-dessous peuvent être fusionnées pour donner une seule contribution qui est :

- le système permet aux usagers de mettre à jour leurs données personnelles

En effet, les enseignants et les étudiants font partie de la classe usager des utilisateurs du système. De plus, la modification de données personnelles fait partie de la mise à jour des données.

4.4.3 Suppression des incohérences

L'incohérence des contributions se manifeste par des contradictions entre elles [Buj05]. Par exemple on considère les exigences suivantes :

- Un étudiant peut renouveler un prêt deux fois de suite au maximum.
- Un enseignant peut renouveler son prêt au moins une fois.
- Tout usager peut renouveler son prêt jusqu'à trois fois de suite.

Les trois contributions ci-dessus portent sur les modalités de renouvellement d'un prêt. Cependant, il y a contradiction entre la première et la troisième et entre la deuxième et la troisième. Selon la première, un étudiant peut renouveler son prêt deux fois successives alors que la troisième dit qu'il peut le renouveler trois fois. La deuxième contribution dit que les enseignants chercheurs n'ont pas un nombre limite de renouvellement des ouvrages alors que la troisième dit qu'ils sont limités à trois renouvellements. Ce type d'incohérence peut être résolu en n'utilisant pas le mot "usager" pour exprimer des exigences qui dépendent du type d'usager. En d'autres termes, il faut éviter les factorisations. On obtient donc ce qui suit :

- Un étudiant a droit à deux renouvellements de son prêt.
- Un enseignant chercheur peut renouveler un prêt trois fois de suite.

L'incohérence est ainsi supprimée.

Comme pour l'étape 2, on utilise des thinkLets de clarification et d'organisation pour soutenir les tâches de cette activité. Les fonctionnalités de l'outil qui sont nécessaires à l'utilisation de ces thinkLets de clarification et d'organisation sont :

F1) L'outil doit offrir la possibilité à chaque participant d'attribuer des valeurs comme "ambiguë" et "claire" à des contributions pour mesurer leur niveau d'ambiguïté ou de clarté.

F2) L'outil doit permettre à chaque participant de marquer les contributions similaires de la même manière (couleur, cocher, etc.) pour déterminer les redondances.

F3) L'outil doit permettre à chaque participant de marquer ou de mettre en évidence les contributions contradictoires.

Le facilitateur doit diriger la session et veiller à ce que les faits et dires de chaque participant soient pris en compte à toutes les étapes. L'utilisation de l'outil a pour avantage de permettre à tous de contribuer au même moment, ce qui n'est pas possible sans outil. Le facilitateur peut tout de même introduire à chaque fois des temps de discussion si cela s'avère nécessaire.

Sortie :

- Une liste de contributions sans ambiguïtés, ni redondances, ni incohérences.

Illustration

Ayant la liste de besoins du tableau IV.3, nous passons maintenant à l'analyse des besoins suivant le processus décrit par la figure IV.6.

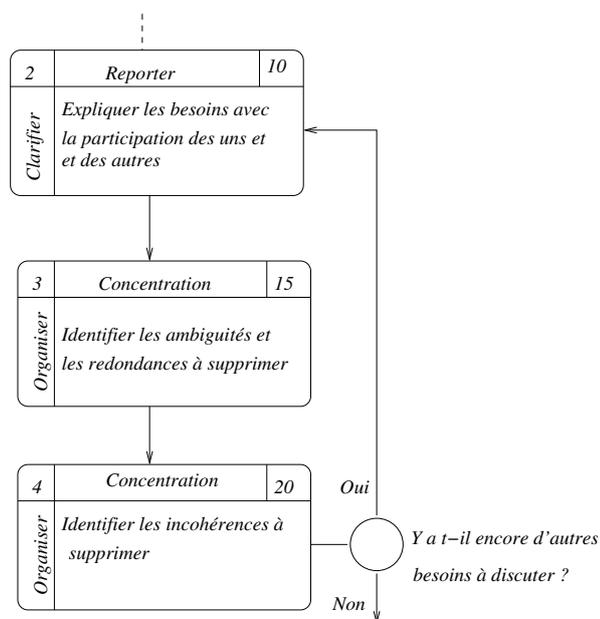


FIGURE IV.6 – Analyse des besoins

Suppression des ambiguïtés

On utilise le thinkLet de clarification "Reporter" qui permet d'expliquer les contributions avec la participation de tous. Dans l'exemple présent, on considère le tableau IV.3 et chaque point de chaque catégorie. Le besoin : "le système permet à l'utilisateur de faire des mises à jour" est ambigu en un sens qu'il ne précise pas de quel type d'utilisateur il s'agit et de quel type de mise à jour il s'agit puisqu'il y en a plusieurs. Les utilisateurs peuvent être le personnel de la bibliothèque qui font des mises à jour du type : enregistrer les retours d'ouvrages et modifier la quantité disponible en rayon, enregistrer de nouveaux ouvrages arrivés dans la

bibliothèque, etc. L'autre type d'utilisateurs est composé d'usagers de la bibliothèque qui sont les étudiants, les enseignants et les chercheurs. Ces derniers sont concernés par des mises à jour portant sur leurs données personnelles telles leurs adresses postales et électroniques. Pour pallier ce type d'ambiguïté, on décompose le besoin en plusieurs besoins pour être précis. Cela donnerait :

- Le système permet au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages.
- Le système permet aux usagers de mettre à jour leurs données personnelles.

On procède ainsi pour tous les besoins qui ne paraissent pas clairs aux participants.

Suppression des redondances

Le tableau IV.3 contient des besoins qui se répètent sous différentes formes. De plus, la phase de suppression des ambiguïtés est très susceptible d'engendrer de nouvelles redondances. Pour supprimer ces redondances, on utilise un thinkLet d'organisation appelé "Concentration". Ce thinkLet prend une liste de contributions contenant des redondances des ambiguïtés et des chevauchements et renvoie une nouvelle liste dans laquelle les contributions ambiguës sont reformulées et les contributions portant sur les redondantes sont fusionnées.

Suppression des incohérences

Cette étape consiste à identifier les contributions incohérentes et à supprimer l'incohérence entre elles.

Le même thinkLet d'organisation "Concentration" peut être utilisé pour la suppression des incohérences.

A la fin de ces différentes étapes, nous avons le tableau IV.4 avec les nouvelles formulations en italique.

Etape 3

Besoins fonctionnels	Besoins non-fonctionnels
Recherche	Qualité
<ul style="list-style-type: none"> - L'utilisateur doit trouver l'ouvrage recherché le plus rapidement possible - Le système offrira plusieurs méthodes de recherche : par titre, auteur, etc. - Des précisions sont nécessaires pour les références - Les résultats de la recherche seront disponibles dans une page particulière 	<ul style="list-style-type: none"> - L'ergonomie doit être sobre - La mise en page du site doit faciliter la démarche à l'aide d'une présentation claire et intuitive - La présentation du site doit être simple pour éviter trop d'efforts - Il faut un formulaire de demande simple - L'effort pour remplir un formulaire doit être minimale - Une aide en ligne efficace est nécessaire - L'utilisateur peut accéder à l'aide en ligne à tout moment - Une visite guidée fera partie de l'aide
Découverte	Performance
<ul style="list-style-type: none"> - Tout ouvrage est présenté sur sa propre page - Toute page pour un ouvrage comporte une image de cet ouvrage - Toute page pour un ouvrage contient la table des matières détaillée de l'ouvrage - Des extraits de chapitres d'un ouvrage seront accessibles sur la page de cet ouvrage - La page d'un ouvrage présente aussi la quantité disponible pour le prêt 	<ul style="list-style-type: none"> - Le système supportera jusqu'à 100000 usagers au moins - Une recherche d'ouvrage ne dépassera pas 30 secondes - Le système supportera au moins 5000 connexions simultanées Le catalogue contiendra au moins 600000 ouvrages
Sélection	Contraintes
<ul style="list-style-type: none"> - L'utilisateur peut sélectionner les ouvrages les uns après les autres - Les étudiants sont limités à trois ouvrages - Le nombre d'ouvrages à emprunter est limité à six pour les enseignants et chercheurs - L'utilisateur a la liberté de mettre un ouvrage dans le panier ou de l'enlever 	Mises à jour
Prêt	
<ul style="list-style-type: none"> - L'utilisateur peut accéder au formulaire de prêt à tout moment - Le numéro d'identifiant doit être enregistré au moment du prêt - Les coordonnées de l'utilisateur sont disponibles dans le formulaire de prêt - un étudiant a droit à deux renouvellements de son prêt - un enseignant chercheur peut renouveler un prêt jusqu'à trois fois de suite 	<ul style="list-style-type: none"> - le système permet aux usagers de mettre à jour leurs données personnelles - le système permet au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages

Légende	
Symboles	Significations
- Besoin	désigne un besoin préexistant
- Besoin : description	désigne un nouveau besoin ou une reformulation

TABLE IV.4 – Résultat de l'Analyse des besoins

4.5 Transformation des besoins en exigences techniques

Entrée :

1. Une liste de contributions sans ambiguïtés, ni redondances, ni incohérences.
2. Un thinkLet d'évaluation.
3. Un thinkLet d'organisation.

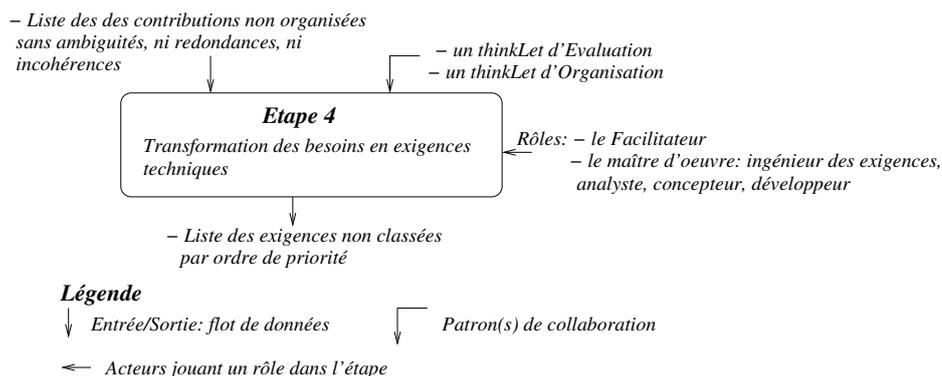


FIGURE IV.7 – Etape de transformation des besoins en exigences techniques.

Les besoins à l'issue de la phase d'analyse ne sont pas formels, ils sont exprimés en langage naturel et manquent souvent de précisions qui font qu'ils ne sont pas directement exploitables par l'équipe de développement du système. De ce fait, il est important de les transcrire en un langage techniquement compréhensible afin qu'ils puissent être implémentés. On parle de transformation des besoins en exigences techniques. Cette étape a pour but la structuration des exigences. Les exigences mal structurées sont des exigences regroupées de manière arbitraire, qui ne sont pas présentées de manière organisée, qui sont cachées dans des phrases sans être identifiées par un numéro et qui n'utilisent pas le verbe d'exigence "devoir". Ce type d'exigences est facilement oublié et non pris en compte dans la réalisation du système. Cette structuration se fait suivant plusieurs règles qui sont, par exemple, l'utilisation du verbe d'exigence "devoir", l'attribution d'un numéro d'identifiant unique, d'une description, d'un justificatif, d'un auteur et bien d'autres choses à chaque exigence [Buj05].

Comme déjà dit, les contributions en entrée de cette étape sont des besoins qui n'ont pas de format spécifique imposé. Au cours des étapes précédentes on ne mettait pas un accent particulier sur la forme des contributions. La raison est que ces contributions étaient informelles et n'exprimaient que les besoins des différentes parties prenantes. Ce qui importait était la clarté, la non redondance et la cohérence de ces contributions. Nous ne nous soucions pas du fait que ces besoins soient techniquement réalisables ou pas. Le besoin doit donc avoir du sens d'abord pour celui qui l'émet et ensuite pour ceux qui le reçoivent puisque nous sommes dans un contexte de collaboration. Tout besoin n'ayant pas une réponse technique

ne saurait être traduit en exigence technique. Dans un souci de formalisation, les exigences ont ici le format spécifique qui est inspiré du modèle de spécification Volere [RR07] présentée à la figure IV.8.

Exigence #: <i>identifiant unique</i>
Type d'exigence: <i>catégorie et éventuellement sous catégorie de l'exigence</i>
Description: <i>une phrase pour définir l'intention de l'exigence</i>
Justification: <i>la raison d'être de l'exigence</i>
Auteur: <i>le rôle de la personne qui a fourni l'exigence</i>
Priorité: <i>une valeur signifiant l'importance de l'exigence pour son auteur</i>
Moyen de validation: <i>le ou les moyens utilisés pour valider l'exigence</i>
Date: <i>date de dernières modifications</i>

FIGURE IV.8 – Les composants d'une exigence technique.

Le numéro d'une exigence est un identifiant composé de symboles alphanumériques qui est unique.

Le type d'exigence est une information sur la catégorie et/ou sous catégorie à laquelle appartient une exigence. Par exemple nous avons les catégories "exigences fonctionnelles" et "exigences non fonctionnelles" qui peuvent elles aussi avoir des sous catégories telles que "exigences de qualité" et "exigences de sécurité".

La description de l'exigence doit être une combinaison de sujet, de verbe et de complément :

Sujet + Verbe(s) + Complément

Le verbe d'exigence "devoir" est en général le premier verbe du groupe verbal. Un exemple de description d'une exigence serait donc :

"Le système doit permettre aux usagers de faire la mise à jour de leurs données personnelles".
Tout besoin doit être transformé pour donner la forme ci-dessus.

La justification montre la raison ou la motivation derrière la proposition de chaque exigence. Une justification de l'exigence ci-dessus peut être : "les données personnelles des usagers du système peuvent changer à tout moment".

Une perspective peut être : "pour permettre aux usagers de mettre à jour eux-mêmes et de façon libre leurs données personnelles".

Les besoins dont cette exigence provient sont par exemple :

- le système permet aux usagers de mettre à jour leurs données personnelles,
- les enseignants peuvent faire la mise à jour de leurs données personnelles,
- les étudiants pourront accéder aux informations qui les concernent pour les modifier.

Nous constatons que cette étape prolonge la phase précédente car elle permet, dans une certaine mesure, de réduire la redondance.

Au cours de cette phase, on (re)organise les exigences en catégories et sous catégories existantes ou nouvelles qui ont été créées. On ouvre chaque catégorie et les exigences qu'elle contient sont passées en revue. On peut en rajouter ou les modifier.

On utilise un thinkLet d'évaluation et un thinkLet d'organisation pour cette étape. Les fonctionnalités que l'outil sous-jacent doit offrir sont :

F1) L'outil doit permettre aux spécialistes des exigences et à la partie technique d'attribuer des valeurs aux exigences symbolisant le niveau de faisabilité technique, par exemple 0, 2, 3 et 4. Le chiffre 0 voulant dire qu'il n'y a pas de solution technique pour ce besoin et 4 signifie que le besoin est techniquement satisfaisable.

F2) Le système doit permettre de créer/supprimer des répertoires et sous répertoires symbolisant les catégories et sous catégories pour les exigences.

F3) Le système doit permettre de déplacer une exigence d'une catégorie à une autre. C'est le facilitateur de la session qui crée les catégories avec l'approbation des participants. C'est aussi lui qui peut faire changer de catégories à une exigence. Comme toujours, une conversation ou une discussion peut suivre si le facilitateur le juge nécessaire.

Sortie

1. Une liste de catégories contenant des exigences techniques mais non classées par ordre de priorité.

Illustration

Une fois que nous avons la liste des besoins du tableau IV.4 qui sont sans ambiguïté, ni redondances, ni incohérences, nous pouvons procéder à la transformation de ces besoins en exigences techniques. Il s'agit d'abord de mettre chaque besoin sous la forme d'une description d'exigence comme montré sur la figure IV.8, puis d'attribuer les autres propriétés telles que le numéro d'identifiant, la justification, l'auteur et la date de création.

La figure IV.9 montre le processus de collaboration suivant lequel cette étape est exécutée.

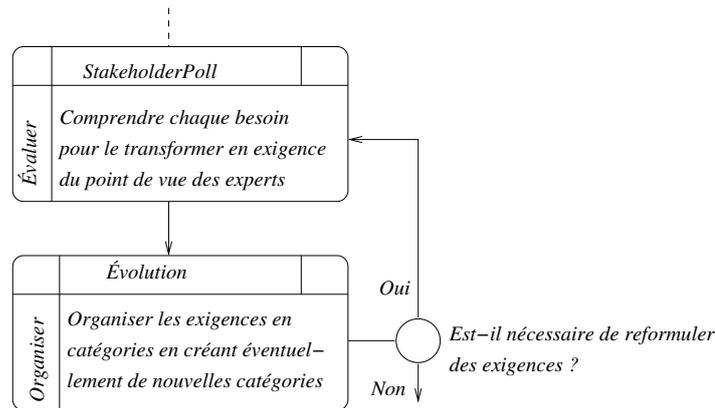


FIGURE IV.9 – Transformation des besoins en exigences techniques

On choisit le thinkLet d'évaluation "StakeholderPoll" pour évaluer chaque besoin en vue de le transformer en exigence. Ce thinkLet est surtout utilisé par des experts tels que les ingénieurs des exigences, des analystes et des concepteurs. Nous utilisons ensuite un thinkLet d'organisation en l'occurrence le thinkLet "Evolution" pour réorganiser ces exigences en catégories et sous catégories car de nouveaux types d'exigences peuvent être découverts pendant le passage des besoins aux exigences. Ce thinkLet prend en entrée une liste de contributions non catégorisées, renvoie une liste de catégories pour l'organisation des contributions et affecte à chaque contribution une catégorie. Dans l'exemple de la bibliothèque on aurait le tableau IV.5 après le thinkLet "StakeholderPoll" et le tableau IV.6 montre le résultat après le thinkLet "Evolution". Nous remarquons l'apparition de nouvelles exigences dans le tableau IV.5 en italique et la création de deux nouvelles catégories d'exigences dans le tableau IV.6.

Etape 4.1

Besoins fonctionnels	Besoins non-fonctionnels
<p>Recherche</p> <ul style="list-style-type: none"> - EFR1 : Le système doit trouver l'ouvrage recherché le plus rapidement possible - EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, etc. - EFR3 : Le système doit donner des références pour chaque ouvrage - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche 	<p>Qualité</p> <ul style="list-style-type: none"> - ENFQ1 : Le système doit avoir une présentation simple claire et intuitive - ENFQ2 : Le système doit avoir une mise en page simple -> justification : pour faciliter l'utilisation et éviter trop d'efforts - ENFQ3 : Le système doit offrir un formulaire simple -> justification : pour minimiser l'effort pour remplir un formulaire - ENFQ4 : Le système doit avoir une aide en ligne accessible à l'utilisateur à tout moment - ENFQ5 : Le système doit permettre une visite guidée avec son aide en ligne - ENFQ6 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe - ENFQ7 : Le système doit permettre de stocker du numéro d'identifiant et mot de passe jusqu'à la fin de la procédure de prêt - ENFQ8 : Le système doit supprimer le mot de passe à la fin de la procédure de prêt de l'utilisateur
<p>Découverte</p> <ul style="list-style-type: none"> - EFD1 : Le système doit présenter chaque ouvrage sur sa propre page - EFD2 : Le système doit offrir une image de chaque ouvrage sur sa page - EFD3 : Le système doit permettre de visualiser la table des matières détaillée de chaque ouvrage sur sa page - EFD4 : Le système doit rendre disponibles des extraits de chapitres d'un ouvrage sur sa page - EFD5 : Le système doit fournir des informations sur la quantité disponible pour le prêt d'un ouvrage 	<p>Performance</p> <ul style="list-style-type: none"> - ENFP1 : Le système doit gérer au moins 100000 usagers - ENFP2 : Le système doit retrouver un ouvrage au bout de 30 secondes - ENFP3 : Le système doit supporter au moins 5000 connexions simultanées - ENFP4 : Le système doit avoir un catalogue d'au moins 600000 ouvrages
<p>Sélection</p> <ul style="list-style-type: none"> - EFS1 : Le système doit permettre à l'utilisateur de sélectionner les ouvrages les uns après les autres - EFS2 : Le système doit limiter le prêt des étudiants à trois ouvrages - EFS3 : Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages - EFS4 : Le système doit permettre à l'utilisateur de mettre un ouvrage dans le panier ou de l'enlever librement 	<p>Contraintes</p> <p>Mises à jour</p> <ul style="list-style-type: none"> - CMJ1 : le système doit permettre aux usagers de mettre à jour leurs données personnelles - CMJ2 : le système doit permettre au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages

Prêt	
<ul style="list-style-type: none"> - EFP1 : Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt - EFP3 : Le système doit montrer les coordonnées de l'utilisateur dans le formulaire de prêt - EFP4 : Le système doit permettre à un étudiant de renouveler un prêt deux fois de suite au maximum - EFP5 : Le système doit permettre à un enseignant de renouveler son prêt au moins une fois - <i>EFP6 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur</i> 	-----
Légende	
Symboles	Significations
- NUM : description	désigne une exigence préexistante
- <i>NUM : description</i>	désigne une nouvelle exigence ou une reformulation
->	désigne une justification pour une exigence

TABLE IV.5 – Résultat du "StakeholderPoll"

Les points en italique représentent la reformulation des besoins pour en faire des descriptions d'une exigence technique avec, parfois, une justification à côté. Il faut rappeler qu'une justification n'est pas obligatoire. Dans la deuxième phase de la transformation, nous faisons une réorganisation de toutes les exigences dont le résultat est présenté dans le tableau IV.6.

Etape 4.2

Besoins fonctionnels	Besoins non-fonctionnels
Recherche	Qualité
<ul style="list-style-type: none"> - EFR1 : Le système doit trouver l'ouvrage recherché le plus rapidement possible - EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, etc. - EFR3 : Le système doit donner des références pour chaque l'ouvrage - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche 	<ul style="list-style-type: none"> - ENFQ1 : Le système doit avoir une présentation simple claire et intuitive - ENFQ2 : Le système doit avoir une mise en page simple -> justification : pour faciliter l'utilisation et éviter trop d'efforts pour éviter - ENFQ3 : Le système doit offrir un formulaire simple -> justification : pour minimiser l'effort pour remplir un formulaire - ENFQ4 : Le système doit avoir une aide en ligne accessible à l'utilisateur à tout moment - ENFQ5 : Le système doit permettre une visite guidée avec son aide en ligne - <i>ENFQ6 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe</i> - <i>ENFQ7 : Le système doit permettre de stocker le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt</i> - <i>ENFQ8 : Le système doit supprimer le mot de passe la fin de la procédure de prêt de l'utilisateur</i>
Découverte	Performance
<ul style="list-style-type: none"> - EFD1 : Le système doit présenter chaque ouvrage sur sa propre page - EFD2 : Le système doit offrir une image de chaque ouvrage sur sa page - EFD3 : Le système doit permettre de visualiser la table des matières détaillée de chaque ouvrage sur sa page - EFD4 : Le système doit rendre disponibles des extraits de chapitres d'un ouvrage sur sa page - EFD5 : Le système doit fournir des informations sur la quantité disponible pour le prêt d'un ouvrage 	<ul style="list-style-type: none"> - ENFP1 : Le système doit gérer au moins 100000 usagers - ENFP2 : Le système doit retrouver un ouvrage au bout de 30 secondes - ENFP3 : Le système doit supporter au moins 5000 connexions simultanées - ENFP4 : Le système doit avoir un catalogue d'au moins 600000 ouvrages

Sélection	Contraintes
<ul style="list-style-type: none"> - EFS1 : Le système doit permettre à l'utilisateur de sélectionner les ouvrages les uns après les autres - EFS2 : Le système doit limiter le prêt des étudiants à trois ouvrages - EFS3 : Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages - FS4 : Le système doit permettre à l'utilisateur de mettre un ouvrage dans le panier ou de l'enlever librement 	<p>Mises à jour</p> <ul style="list-style-type: none"> - CMJ1 : le système doit permettre aux usagers de mettre à jour leurs données personnelles - CMJ2 : le système doit permettre au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages
Prêt	Panier
<ul style="list-style-type: none"> - EFP1 : Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt - EFP3 : Le système doit montrer les coordonnées de l'utilisateur dans le formulaire de prêt - EFP4 : Le système doit permettre à un étudiant de renouveler un prêt deux fois de suite au maximum - EFP5 : Le système doit permettre à un enseignant de renouveler son prêt au moins une fois - EFP6 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur 	<ul style="list-style-type: none"> - CP1 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur -> justification : sa durée de vie n'excède pas la durée de la connexion
	Sécurité
	<ul style="list-style-type: none"> - ENFS1 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe - ENFS2 : Le système doit permettre de stocker le prêt, le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt - NFS3 : Le système doit supprimer le mot de passe après la fin de la procédure de prêt de l'utilisateur
Légende	
Symboles	Significations
- NUM : description	désigne une exigence préexistante
- NUM : <i>description</i>	désigne une nouvelle exigence ou une reformulation
->	désigne une justification pour une exigence

TABLE IV.6 – Résultat du thinkLet "Evolution"

Deux nouvelles sous catégories, "Panier" et "Sécurité", ont été créées pour mieux organiser les exigences. L'exigence de la sous catégorie "Panier" était dans la sous catégorie "Prêt" et celles de la sous catégorie "Sécurité" était dans la sous catégorie "Qualité".

4.6 Hiérarchisation des exigences techniques

Entrée :

1. Une liste de catégories contenant des exigences techniques non hiérarchisées.
2. Un thinkLet d'évaluation.
3. Un thinkLet de consensus.

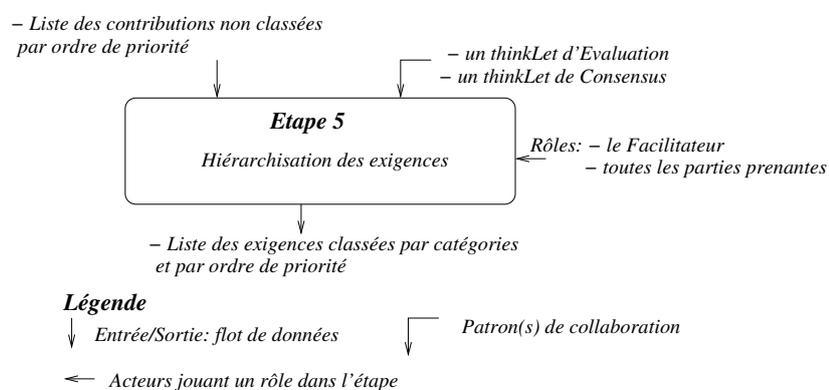


FIGURE IV.10 – Etape de la hiérarchisation des exigences techniques.

Cette étape consiste à mettre les exigences par ordre de priorité selon un certain nombre de critères identifiés. Ces critères sont passés en revue et les participants évaluent chaque exigence selon les critères. Dans la norme EIA-632 [All99], cette étape porte le nom de hiérarchisation des exigences.

Les thinkLets utiles à cette étape sont un thinkLet d'évaluation et un thinkLet de consensus. Les fonctionnalités de l'outil supportant ces activités sont :

F1) L'outil doit permettre d'attribuer un niveau de priorité à chaque exigence sur une échelle de 1 à 10 par exemple.

F2) L'outil doit permettre ensuite d'afficher le score pour chaque exigence en marquant la différence avec la couleur. Par exemple, le vert est affecté aux exigences qui ont un haut score et le rouge à celles ayant un score bas.

F3) L'outil doit permettre de voter pour s'accorder sur la priorité des exigences qui présentent de nombreux contrastes. Une exigence E_i présente un contraste lorsque sa priorité est 8 pour un participant et 2 ou 3 pour d'autres.

Le facilitateur peut introduire une discussion entre participants pour aboutir à des accords.

Sortie :

1. Une liste de catégories contenant des exigences hiérarchisées.

Illustration

Cette étape a pour but de disposer les exigences par ordre de priorité. Cet ordre de priorité dépend des critères définis par les différentes parties prenantes. L'ordre hiérarchique peut porter sur le coût d'une solution, sa faisabilité technique, les retombées sociales et politiques de la solution, etc.

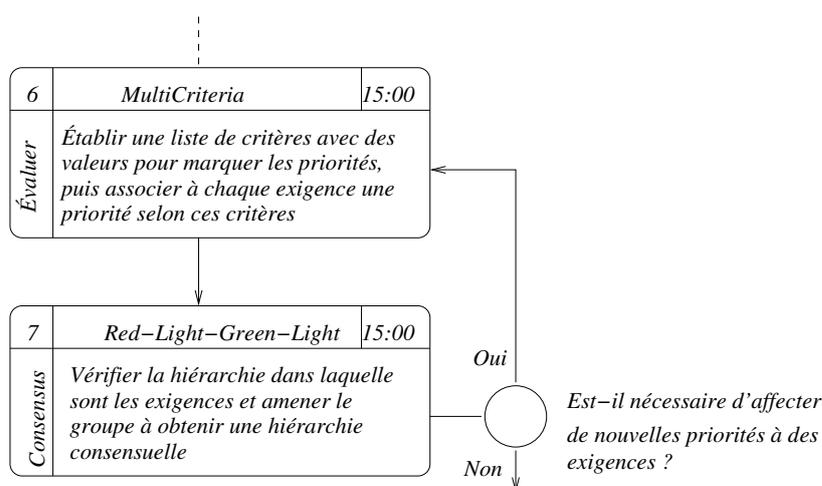


FIGURE IV.11 – Hiérarchisation des exigences

Dans notre cas, on utilise le thinkLet d'évaluation "MultiCriteria" qui prend en entrée un liste des points à évaluer, une liste de critères d'évaluation pour chaque point et une liste de pondérations des critères pour la régulation de l'influence de chaque critère sur l'évaluation complète. Le thinkLet renvoie un tableau montrant comment le groupe a évalué les exigences et une liste des exigences par ordre de priorité.

Soit la liste de critères donnée dans le tableau IV.7 :

Avec chaque critère et son poids, on revoie l'ordre des exigences. Le résultat est semblable à celui présenté dans le tableau IV.8 (supra) dans un autre ordre. Pour prendre une décision finale on utilise le thinkLet de consensus "Red-Light-Green-Light" qui prend en entrée les résultats d'un vote multicritère et qui renvoie un consensus dans le groupe et un ensemble de points hiérarchisés.

Pour notre exemple de la bibliothèque, le facilitateur conduit l'étape de pondération des critères en demandant à chaque partie d'attribuer des poids aux critères qu'elle propose. On trouve le tableau IV.8 (supra) avec l'exemple de la bibliothèque.

Une fois que les critères et leurs poids selon les rôles des participants sont connus, le

Etape 5.1

- la solution ne doit pas être coûteuse car le budget de la bibliothèque est limité
- la solution doit promouvoir l'image de l'université
- facilité de réalisation technique
- importance métier de l'exigence
- utilisabilité
- facilité de recherche d'ouvrages
- facilité de mise à jour
- utilité
- ...

La pondération des critères peut être :

1 : priorité très faible et 7 : priorité très faible

TABLE IV.7 – Critères et échelle pour l'évaluation

Etape 5.2

Critères	Maître d'ouvrage	Maître d'œuvre	Usager	Personnel bibliothèque
Coût	7	3	1	4
Promotion de l'image de l'université	6	2	3	6
Facilité d'implémentation	1	7	1	1
Importance métier	2	6	6	7
Facilité de recherche d'ouvrages	1	3	7	7
Facilité de mise à jour	1	4	4	7
Utilisabilité	3	4	5	5
Utilité	5	4	6	6

TABLE IV.8 – Pondération des critères par les différents rôles

facilitateur conduit le groupe à évaluer chaque exigence selon ces critères. Cela revient à ce que chaque participant considère les exigences du tableau IV.6 (supra) et qu'il attribue un poids à chacune d'elles selon les critères définis (supra, le tableau IV.8). C'est l'évaluation multicritères pour laquelle on utilise le thinkLet "MultiCriteria". Nous obtenons enfin des résultats selon un ordre de priorité qui n'est pas l'ordre consensuel (infra, tableau IV.9).

Etape 5.3

Exigences	Priorités
<ul style="list-style-type: none"> - EFR1 : Le système doit retrouver l'ouvrage le plus rapidement possible - EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, mot clé, etc. - ENFS1 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe - ENFS2 : Le système doit permettre de stocker le prêt, le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt - ENFS3 : Le système doit supprimer le mot de passe après la fin de la procédure de prêt de l'utilisateur 	7
<ul style="list-style-type: none"> - EFR3 : Le système doit donner des références pour chaque ouvrage. 	5
<ul style="list-style-type: none"> - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche - EFP1 : <i>Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment</i> - EFP2 : <i>Le système doit enregistrer le numéro d'identifiant au moment du prêt</i> 	4
<ul style="list-style-type: none"> - EFP1 : Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt 	3
<ul style="list-style-type: none"> - ENFQ1 : Le système doit avoir une présentation simple claire et intuitive - ENFQ2 : Le système doit avoir une mise en page simple -> justification : pour faciliter l'utilisation et éviter trop d'efforts - EFS1 : Le système doit permettre à l'utilisateur de sélectionner les ouvrages les uns après les autres - EFS2 : Le système doit limiter le prêt des étudiants à trois ouvrages - EFS3 : Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages - EFS4 : Le système doit permettre à l'utilisateur de mettre un ouvrage dans le panier ou de l'enlever librement 	2
<ul style="list-style-type: none"> - CMJ1 : le système doit permettre aux usagers de mettre à jour leurs données personnelles - CMJ2 : le système doit permettre au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages - CP1 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur -> justification : sa durée de vie n'excède pas la durée de la connexion 	1

Légende	
Symboles	Significations
- NUM : description	désigne une exigence préexistante
- NUM : <i>description</i>	désigne une nouvelle exigence ou une reformulation
->	désigne une justification pour une exigence

TABLE IV.9 – Résultat du "MultiCriteria"

Pour amener le groupe à un consensus, on utilise le thinkLet de consensus appelé "Red-Light-Green-Light" qui prend les résultats de "MultiCriteria" et renvoie une liste ordonnée selon un ordre consensuel (voir tableau IV.10). Les points en italique sont ceux qui ont changé de priorité.

Etape 5.4	
Exigences	Priorités
- EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, mot clé, etc. - ENFS1 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe - ENFS2 : Le système doit permettre de stocker le prêt, le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt - ENFS3 : Le système doit supprimer le mot de passe après la fin de la procédure de prêt de l'utilisateur	7
- EFS2 : <i>Le système doit limiter le prêt des étudiants à trois ouvrages</i> - EFS3 : <i>Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages</i> - EFS4 : <i>Le système doit permettre à l'usager de mettre un ouvrage dans le panier ou de l'enlever librement</i>	6
- EFR3 : Le système doit donner des références pour chaque ouvrage. - EFR1 : <i>Le système doit retrouver l'ouvrage le plus rapidement possible</i>	5
- EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche - EFP1 : Le système doit rendre accessible à l'usager le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt	4
- CMJ1 : <i>le système doit permettre aux usagers de mettre à jour leurs données personnelles</i> - CMJ2 : <i>le système doit permettre au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages</i> - EFR4 : <i>Le système doit donner les résultats de la recherche dans une page particulière.</i> - CP1 : <i>Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur</i> -> <i>justification : sa durée de vie n'excède pas la durée de la connexion</i>	3
- ENFQ1 : Le système doit avoir une présentation simple claire et intuitive - ENFQ2 : Le système doit avoir une mise en page simple -> justification : pour faciliter l'utilisation et éviter trop d'efforts - EFS1 : Le système doit permettre à l'usager de sélectionner les ouvrages les uns après les autres	2
	1

Légende	
Symboles	Significations
- NUM : description	désigne une exigence préexistante
- NUM : <i>description</i>	désigne une nouvelle exigence ou une reformulation
->	désigne une justification pour une exigence

TABLE IV.10 – Résultat du "Red-Light-Green-Light"

4.7 Vérification et Validation des exigences

Entrée :

1. Une liste de catégories contenant des exigences hiérarchisées.
2. Un thinkLet de génération.
3. Un thinkLet de consensus.

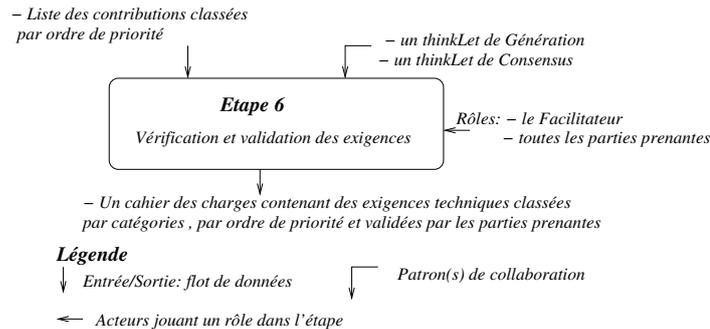


FIGURE IV.12 – Etape de vérification et de validation des exigences.

Cette étape consiste à faire une dernière vérification des exigences produites pendant les phases précédentes et à les faire valider par les différentes parties prenantes pertinentes. Il s'agit d'une vérification et d'une validation par rapport à la définition des exigences et non par rapport à un système déjà construit. Puisque l'étape de validation est souvent sujette à des conflits entre différentes parties, une négociation est toujours à envisager. Pour cette phase de négociation, nous nous sommes inspirés de l'approche de négociation des exigences connues sous le nom de win-win [Boe88].

4.7.1 Identification du problème

Chaque exigence d'une catégorie est considérée comme une *win condition*. Pour chaque *win condition*, chaque participant observe si elle lui convient. Si oui, le participant la valide en passant dessus. Si non, le participant met une objection en bas de cette win condition pour signifier son désaccord avec ce point tout en explicitant le problème.

4.7.2 Proposition d'options aux problèmes

Entrée :

1. Liste des *win conditions* non validées avec objection.

Les participants passent en revue chaque *win condition* ayant reçu une objection dans le but de proposer des solutions alternatives.

Sortie :

1. Liste de *win conditions* non validées avec objections et suggestions de nouvelles *win conditions*.

4.7.3 Validation des options

Pour chaque exigence ayant reçu une ou plusieurs objections avec des propositions de solutions alternatives appelées *options* dans la terminologie de l'approche *Win Win*, ces options sont passées en revue. Les participants émettent leur approbation ou désapprobation pour chaque option avec des arguments à l'appui. A la fin de cette étape, on choisit l'option sur laquelle tout le monde s'accorde pour la valider en remplacement de la *win condition* objectée.

Sortie :

- liste d'options validées

Cette étape est supportée par un thinkLet de génération pour permettre aux participants de contribuer à de nouvelles idées par des propositions, des contre-propositions, et un thinkLet de consensus pour leur permettre de s'accorder sur les différentes options qui s'offrent à eux. Les fonctionnalités requises d'un outil qui supporte ces thinkLets sont :

F1) l'outil doit permettre d'éditer des propositions et des contre-propositions l'une en dessous de l'autre,

F2) l'outil doit permettre de voter pour faire un choix consensuel et de le valider.

Sortie :

1. Une liste de catégories contenant des exigences hiérarchisées et validées par tous les participants.

Illustration

Cette étape consiste à vérifier pour une dernière fois les exigences techniques et à les faire valider par les participants. Nous utilisons l'approche de négociation *Win Win* pour vérifier et valider les exigences.

Pour notre exemple, on considère chacune des catégories et sous catégories définies dans le tableau provenant de l'étape de la hiérarchisation des exigences (section 4.6) et on fixe chaque exigence comme *win condition*. Chaque participant lit sur son écran ces *win conditions*. Lorsque le participant estime qu'un *win condition* n'est pas valide, il le marque comme un *win condition* problématique en spécifiant la nature du problème. Dans le cas contraire, il passe au *win condition* suivant.

Au deuxième passage, les participants font la même chose en proposant cette fois-ci des solutions appelées options. Pour ce faire, on utilise le thinkLet de génération "Comparative Brainstorming" pour déterminer un ensemble des solutions potentielles aux problèmes précédemment identifiés.

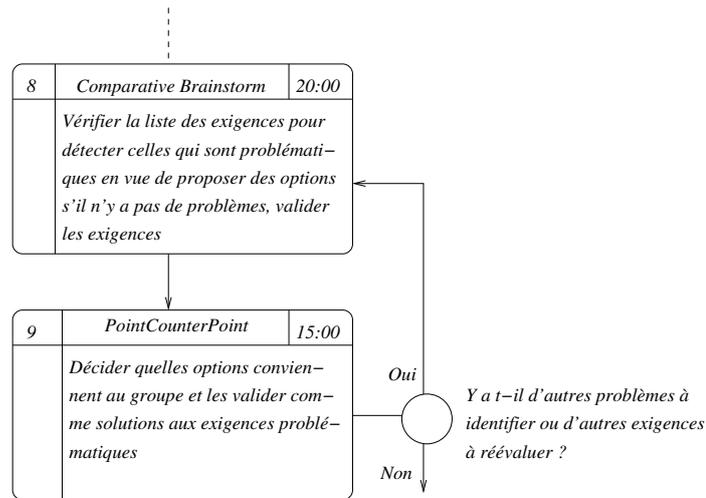


FIGURE IV.13 – Vérification et Validation des exigences

Le thinkLet "Comparative Brainstorming" prend en entrée un ensemble de critères pour juger quelles solutions sont bonnes et lesquelles ne le sont pas et il renvoie en sortie un ensemble de solutions potentielles. Ici, les solutions potentielles portent le nom d'options dans la terminologie de l'approche *Win Win*. Une fois que les options pour chaque problème sont identifiées, on cherche un accord pour obtenir une nouvelle *win condition* en remplacement de celle qui avait été trouvée problématique au début du processus de vérification et de validation. Pour ce faire, on utilise le thinkLet de consensus "PointCounterPoint" pour obtenir un consensus par rapport à ces options. Ce thinkLet prend en entrée une proposition discutable et mène à une position consensuelle. Le facilitateur peut, de temps en temps introduire des discussions si cela est nécessaire. A la fin de cette étape, on a un cahier de charges dont chaque entrée est sous la forme :

Exigence #: <i>CPI</i>
Type d'exigence: <i>Contrainte de conception du Panier</i>
Description: <i>Le système ne doit pas sauvegarder dans la base le panier</i>
Justification: <i>la durée de vie du panier n'excède pas la durée de la connexion</i>
Auteur: <i>l'équipe de conception</i>
Priorité: <i>5 (Moyenne)</i>
Moyen de validation: <i>Simulation</i>
Date: <i>14/11/2008</i>

FIGURE IV.14 – Exemple de spécification d'une exigence

Pour notre exemple, on considère le résultat du tableau IV.10 avec chaque point comme *win condition*. On retrouve un résultat comme celui présenté dans le tableau IV.11 après le premier passage. Les points désignés par une étoile (*) représentent les problèmes identifiés

par les participants. Au second passage, le tableau IV.12 est engendré et les participants proposent des options pour résoudre les problèmes identifiés. Ces options sont des points désignés par deux étoiles (**). Au troisième passage le tableau IV.13 où les points désignés par plus (+) représentent les options retenues pour remplacer les *win conditions* identifiés comme problématiques au premier passage. Il peut y avoir plusieurs options à condition qu'elles ne soient ni contradictoires, ni redondantes.

Etape 6.1

Exigences	Priorités
<ul style="list-style-type: none"> - EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, mot clé, etc. - ENFS1 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe - ENFS2 : Le système doit permettre de stocker le prêt, le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt - ENFS3 : Le système doit supprimer le mot de passe après la fin de la procédure de prêt de l'utilisateur 	7
<ul style="list-style-type: none"> - EFS2 : Le système doit limiter le prêt des étudiants à trois ouvrages * <i>Cela n'est pas suffisant pour des étudiants qui commencent la recherche</i> - EFS3 : Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages - EFS4 : Le système doit permettre à l'utilisateur de mettre un ouvrage dans le panier ou de l'enlever librement 	6
<ul style="list-style-type: none"> - EFR3 : Le système doit donner des références pour chaque ouvrage. - EFR1 : Le système doit retrouver l'ouvrage le plus rapidement possible * <i>Au cas où le système ne retrouve pas l'ouvrage quelles sont les possibilités pour l'utilisateur ?</i> * <i>Y a-t-il d'autres alternatives pour une recherche vaine ?</i> * <i>Est-ce possible de faire des suggestions de commande d'un ouvrage n'existant pas dans la bibliothèque ?</i> 	5
<ul style="list-style-type: none"> - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche - EFP1 : Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt 	4
<ul style="list-style-type: none"> - CMJ1 : le système doit permettre aux usagers de mettre à jour leurs données personnelles * <i>Il sera difficile d'assurer vérifier l'authenticité de ces informations</i> * <i>Comment le système vérifiera-t-il la véracité de ces données ?</i> - CMJ2 : le système doit permettre au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - CP1 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur -> justification : sa durée de vie n'excède pas la durée de la connexion 	3
<ul style="list-style-type: none"> - ENFQ1 : Le système doit avoir une présentation simple claire et intuitive - ENFQ2 : Le système doit avoir une mise en page simple -> justification : pour faciliter l'utilisation et éviter trop d'efforts - EFS1 : Le système doit permettre à l'utilisateur de sélectionner les ouvrages les uns après les autres 	2
	1

Légende	
Symboles	Significations
-	désigne un point préexistant
*	désigne un point pour un problème
->	désigne une justification pour une exigence

TABLE IV.11 – Résultat du premier passage de l'étape de la vérification et la validation

Etape 6.2

Exigences	Priorités
<ul style="list-style-type: none"> - EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, mot clé, etc. - ENFS1 : Le système doit permettre la saisie sécurisée des numéro d'identifiant et le mot de passe - ENFS2 : Le système doit permettre de stocker le prêt, le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt - ENFS3 : Le système doit supprimer le mot de passe après la fin de la procédure de prêt de l'utilisateur 	7
<ul style="list-style-type: none"> - EFS2 : Le système doit limiter le prêt des étudiants à trois ouvrages <i>* Cela n'est pas suffisant pour des étudiants qui commencent la recherche</i> <i>** Le système doit permettre aux étudiants au niveau de la recherche d'emprunter deux ouvrages de plus</i> <i>** Le système doit permettre aux étudiants en recherches d'emprunter le même nombre d'ouvrages que les enseignants et chercheurs</i> <i>** Le système doit autoriser les étudiants en recherche de prendre plus que trois ouvrages</i> - EFS3 : Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages - EFS4 : Le système doit permettre à l'utilisateur de mettre un ouvrage dans le panier ou de l'enlever librement 	6
<ul style="list-style-type: none"> - EFR3 : Le système doit donner des références pour chaque ouvrage. - EFR1 : Le système doit retrouver l'ouvrage le plus rapidement possible <i>* Au cas où le système ne retrouve pas l'ouvrage quelles sont les possibilités pour l'utilisateur ?</i> <i>* Y a-t-il d'autres alternatives pour une recherche vaine ?</i> <i>* Est-ce possible de faire des suggestions de commande d'un ouvrage n'existant pas dans la bibliothèque ?</i> <i>** Le système doit permettre à l'utilisateur de notifier quelque part qu'il n'a pas pu trouver un ouvrage qu'il cherchait</i> <i>** Le système doit permettre de permettre aux usagers de demander l'achat d'un ouvrage par la bibliothèque</i> <i>** Le système doit proposer à l'utilisateur une liste ouvrages susceptibles de remplacer celui qu'il n'a pas pu trouver</i> 	5
<ul style="list-style-type: none"> - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche - EFP1 : Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt 	4
<ul style="list-style-type: none"> - CMJ1 : le système doit permettre aux usagers de mettre à jour leurs données personnelles <i>* Il sera difficile d'assurer vérifier l'authenticité de ces informations</i> <i>* Comment le système vérifiera-t-il la véracité de ces données ?</i> <i>** Le système doit tenir compte seulement des informations enregistrées au moment de l'inscription à l'université</i> <i>** Le système doit enregistrer la demande de modification qui sera validée par le personnel de la bibliothèque sur présentation de justificatifs</i> <i>** Le système doit permettre toutes modifications seulement après que l'utilisateur ait lu et accepté une déclaration sur l'honneur</i> - CMJ2 : le système doit permettre au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - CP1 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur -> justification : sa durée de vie n'excède pas la durée de la connexion 	3

- ENFQ1 : Le système doit avoir une présentation simple claire et intuitive	2
- ENFQ2 : Le système doit avoir une mise en page simple	
-> justification : pour faciliter l'utilisation et éviter trop d'efforts	
- EFS1 : Le système doit permettre à l'utilisateur de sélectionner les ouvrages les uns après les autres	
1	

<i>Légende</i>	
Symboles	Significations
-	désigne un point préexistant
*	désigne un problème
**	désigne une option proposée
->	désigne une justification pour une exigence

TABLE IV.12 – Résultat du deuxième passage pendant la vérification et la validation

Etape 6.3	
Exigences	Priorités
<ul style="list-style-type: none"> - EFR2 : Le système doit offrir plusieurs méthodes de recherche : par titre, auteur, mot clé, etc. - ENFS1 : Le système doit permettre la saisie sécurisée du numéro d'identifiant et du mot de passe - ENFS2 : Le système doit permettre de stocker le prêt, le numéro d'identifiant et le mot de passe jusqu'à la fin de la procédure de prêt - ENFS3 : Le système doit supprimer le mot de passe après la fin de la procédure de prêt de l'utilisateur 	7
<ul style="list-style-type: none"> - EFS2 : Le système doit limiter le prêt des étudiants à trois ouvrages * <i>Cela n'est pas suffisant pour des étudiants qui commencent la recherche</i> ** <i>Le système doit permettre aux étudiants au niveau de la recherche d'emprunter deux ouvrages de plus</i> ++ <i>Le système doit permettre aux étudiants en recherche d'emprunter le même nombre d'ouvrages que les enseignants et chercheurs</i> ** <i>Le système doit autoriser les étudiants en recherche de prendre plus que trois ouvrages</i> - EFS3 : Le système doit limiter le prêt des enseignants et chercheurs à six ouvrages - EFS4 : Le système doit permettre à l'utilisateur de mettre un ouvrage dans le panier ou de l'enlever librement 	6
<ul style="list-style-type: none"> - EFR3 : Le système doit donner des références pour chaque ouvrage. - EFR1 : Le système doit retrouver l'ouvrage le plus rapidement possible * <i>Au cas où le système ne retrouve pas l'ouvrage quelles sont les possibilités pour l'utilisateur ?</i> * <i>Y a-t-il d'autres alternatives pour une recherche vaine ?</i> * <i>Est-ce possible de faire des suggestions de commande d'un ouvrage n'existant pas dans la bibliothèque ?</i> ** <i>Le système doit permettre à l'utilisateur de notifier quelque part qu'il n'a pas pu trouver un ouvrage qu'il cherchait</i> ** <i>Le système doit permettre aux usagers de demander l'achat d'un ouvrage par la bibliothèque</i> ++ <i>Le système doit proposer à l'utilisateur une liste d'ouvrages susceptibles de remplacer celui qu'il n'a pas pu trouver</i> 	5
<ul style="list-style-type: none"> - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - EFR5 : Le système doit permettre le parcours et le reclassement facile des résultats de la recherche - EFP1 : Le système doit rendre accessible à l'utilisateur le formulaire de prêt à tout moment - EFP2 : Le système doit enregistrer le numéro d'identifiant au moment du prêt 	4

<ul style="list-style-type: none"> - CMJ1 : le système doit permettre aux usagers de mettre à jour leurs données personnelles <i>* Il sera difficile d'assurer vérifier l'authenticité de ces informations</i> <i>* Comment le système vérifiera t-il la véracité de ces données ?</i> <i>** Le système doit tenir compte seulement des informations enregistrées au moment de l'inscription à l'université</i> <i>** Le système doit enregistrer la demande de modification qui sera validée par le personnel de la bibliothèque sur présentation de justificatifs</i> <i>++ Le système doit permettre toutes modifications seulement après que l'utilisateur a lu et accepté une déclaration sur l'honneur</i> - CMJ2 : le système doit permet au personnel de la bibliothèque de mettre à jour des données relatives aux ouvrages - EFR4 : Le système doit donner les résultats de la recherche dans une page particulière. - CP1 : Le système ne doit pas sauvegarder dans la base le panier de l'utilisateur -> justification : sa durée de vie n'excède pas la durée de la connexion 	3
<ul style="list-style-type: none"> - ENFQ1 : Le système doit avoir une présentation simple claire et intuitive - ENFQ2 : Le système doit avoir une mise en page simple -> justification : pour faciliter l'utilisation et éviter trop d'efforts - EFS1 : Le système doit permettre à l'utilisateur de sélectionner les ouvrages les uns après les autres 	2
	1

<i>Légende</i>	
Symboles	Significations
-	désigne un item préexistant
*	désigne un problème
**	désigne une option proposée
++	désigne une option retenue
->	désigne une justification pour une exigence

TABLE IV.13 – Résultat du troisième passage pendant la vérification et la validation

Dans ce chapitre, nous avons présenté une méthodologie illustrée pour l'élicitation collaborative des exigences. Cela visait deux points importants qui sont : l'identification des étapes dans l'Elicitation Collaborative des Exigences et l'identification des fonctionnalités qu'un outil doit disposer pour supporter ces étapes. Ayant ainsi identifié les étapes de la méthodologie et les fonctionnalités nécessaires des outils pour supporter chacune de ces étapes, nous abordons dans un ultime chapitre l'outillage de la méthodologie et les travaux expérimentaux entrepris pour son évaluation.

Outillage et travaux expérimentaux de la méthodologie

5.1 Introduction

Puisque nous avons vu dans chapitres précédents la nécessité des outils automatisés pour supporter le processus d'Elicitation des Exigences, nous présentons en l'occurrence deux types d'outils pour supporter la méthodologie proposée dans le chapitre précédent. Ces outils devraient donc avoir les différentes fonctionnalités requises que nous avons identifiées pendant la présentation des étapes de la méthodologie.

Le premier outil est un outil de collaboration appelé ThinkTank qui nous permet d'éliciter les besoins des différentes parties prenantes. Le second est un outil de spécification des exigences qui permet d'exprimer les besoins sous forme d'exigences techniques.

5.2 Les Systèmes Supports de Groupe

Un système support de groupe, communément appelé GSS (*Group Support System*), est un système qui supporte la réflexion collective. Il est une sorte de collection d'outils de messagerie instantanée et de vote pour relever des mesures. Tout utilisateur peut faire une contribution à une liste partagée à tout moment, et toute contribution d'un participant apparaît instantanément sur les écrans des autres participants. Un GSS permet le partage de fenêtres où les utilisateurs peuvent contribuer en même temps, sans que l'un soit bloqué par l'autre.

Les GSS incluent une variété d'outils pour le vote tels que les échelles, la répartition des voix et le vote multicritère. Les utilisateurs peuvent entrer leurs idées dans l'outil de vote, l'évaluer, et voir instantanément les résultats en ligne.

Selon Briggs, ce qui est intéressant avec ce type d'outil n'est pas le fait de voir se produire à l'écran ce que vous pouvez faire, mais plutôt le fait de voir se produire dans le groupe ce que vous pouvez faire [BGB01]. En effet, l'utilisation de GSS permet de créer des patrons d'interactions humaines et de raisonnement qui sont prévisibles et réutilisables par les gens travaillant ensemble vers un but commun [BdVJT01]. Par exemple, le *brainstorming* qui est un processus supporté par les GSS permet à un groupe de rompre avec les façons de penser habituelles en encourageant la réflexion libre. Les GSS permettent également d'organiser les

idées du groupe, de faire converger rapidement le groupe vers des idées plus claires et qui méritent plus d'attention.

Les GSS donnent des supports pour définir une séquence d'étapes (processus de collaboration) qu'une équipe peut suivre pour accomplir ses tâches. Ces étapes sont définies et consignées dans l'agenda du groupe. A chaque étape, l'outil est configuré d'une manière particulière pour promouvoir l'implication de tous les participants en créant de nouveaux patrons de collaboration. Les patrons de collaboration en question sont ceux présentés dans le chapitre 2 sur l'Ingénierie de la Collaboration. Pour rappel, il s'agit de : générer, réduire, organiser, élaborer, résumer, évaluer, faire le consensus. Puisque les GSS sont le plus souvent basés sur un réseau (Web), les membres de l'équipe peuvent interagir indépendamment de toute localisation géographique. Cependant, ils ne sont pas appropriés pour toutes les interactions de groupes. En effet, une équipe peut avoir besoin d'interagir d'une manière traditionnelle (c'est-à-dire les réunions plénières) pour résoudre certains problèmes [BGB01]. Cependant les recherches ont montré que sous certaines conditions, les équipes utilisant un GSS peuvent réduire leur temps de travail de 50 % et réduire le cycle de leur projet de 60 % à 90 % [FSDH07], [DHN90]. De telles équipes obtiennent de meilleurs résultats qu'elles n'auraient pu obtenir sans l'utilisation d'un GSS [FH01].

5.3 A propos de ThinkTank

ThinkTank est un type de GSS basé sur le Web. Il est un produit de GroupSystems et il offre de nombreuses fonctionnalités permettant à un groupe de collaborer pour des prises de décisions, des réunions ordinaires, des ateliers pour la collecte des exigences ou tout autre type d'idées. Le choix de ThinkTank est dû à sa maturité en tant que GSS car il permet l'implémentation des patrons de collaboration que nous avons utilisés dans la méthodologie. De ce fait, l'accent est mis sur les fonctionnalités de ThinkTank dont nous avons besoin dans l'exécution de la méthodologie.

Pour une présentation de l'outil ThinkTank, se reporter à l'annexe 2 (infra, la page 139). Le but de la présentation de l'outil Thinktank est de montrer que les fonctionnalités requises qui ont été identifiées au cours des différentes étapes de notre méthodologie sont offertes par Thinktank.

Cependant, bien que ThinkTank dispose de nombreuses fonctions intéressantes en plus de celles que nous avons exploitées, il n'est pas approprié pour toutes les étapes de notre méthodologie . Ainsi, la section suivante est consacrée à la présentation du deuxième outil utilisé dans nos travaux.

5.4 SpecJ

Nous présentons ici le prototype de spécification des exigences que nous avons développé pour soutenir une partie de notre méthodologie. En effet, si ThinkTank permet la génération

d'idées qui représentent dans notre contexte particulier les besoins d'un utilisateur, SpecJ quant à lui nous permettra de formaliser ces besoins en terme d'exigences. Cela se fera également de façon collaborative car il s'agit là d'une autre application web accessible par tous les participants. Cependant, le travail de spécification relève plus de l'équipe des ingénieurs des exigences qui sont assez compétents pour élaborer ce document. Les utilisateurs pourront tout de même y accéder pour la consultation en vue de vérifier et de valider les exigences. Nous présentons ci-après l'architecture fonctionnelle de SpecJ.

5.4.1 Architecture fonctionnelle

Diagramme de cas d'utilisation

Nous nous intéressons ici aux cas d'utilisation les plus importants pour notre méthodologie comme le montre la figure V.1.

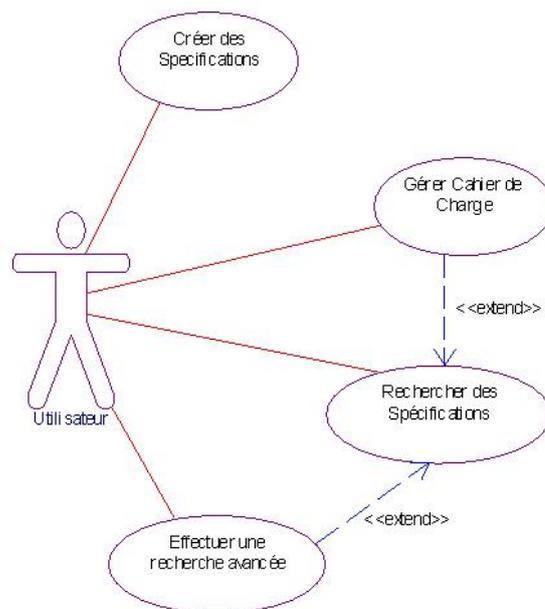


FIGURE V.1 – Diagramme des cas d'utilisation de SPECJ.

L'utilisateur représente ici tout utilisateur du système SpecJ.

SpecJ peut être utilisé de façon générale pour :

- **Créer** de nouvelles spécifications : l'utilisateur doit avoir la possibilité de créer de nouvelles exigences.
- **Gérer** le cahier des charges : l'utilisateur doit avoir la possibilité d'ajouter les spécifications qu'il a créées dans un cahier des charges. Il doit également pouvoir supprimer, modifier le contenu. Le cahier des charges contient essentiellement les spécifications organisées en catégories. L'utilisateur pourra donc faire des opérations sur les catégories et les spécifications. Par exemple, changer la catégorie d'une spécification.

- Faire de la **recherche** sur les spécifications déjà définies. Cette recherche peut être une recherche rapide ou avancée. Quand il s’agit d’une recherche rapide, l’utilisateur saisit des mots clés qu’il désire ou consulte simplement le cahier de charge sans idées préconçues. Dans le cas d’une recherche avancée, l’utilisateur fait des recherches en fonction, soit de la catégorie des spécifications recherchées, soit de leur priorité, soit d’un sujet particulier, soit en fonction de leur date de création.

Nous présentons dans la section suivante le diagramme des classes de conception qui participent aux différents cas d’utilisation que nous venons de citer.

Diagramme des classes de conception

Nous n’avons pas représenté ici toutes les classes et les méthodes de l’application. Seules celles qui contribuent de façon significative à la compréhension du fonctionnement général y sont représentées. Nous donnons les détails sur la spécification de ces méthodes dans l’annexe 5 à la page 145.

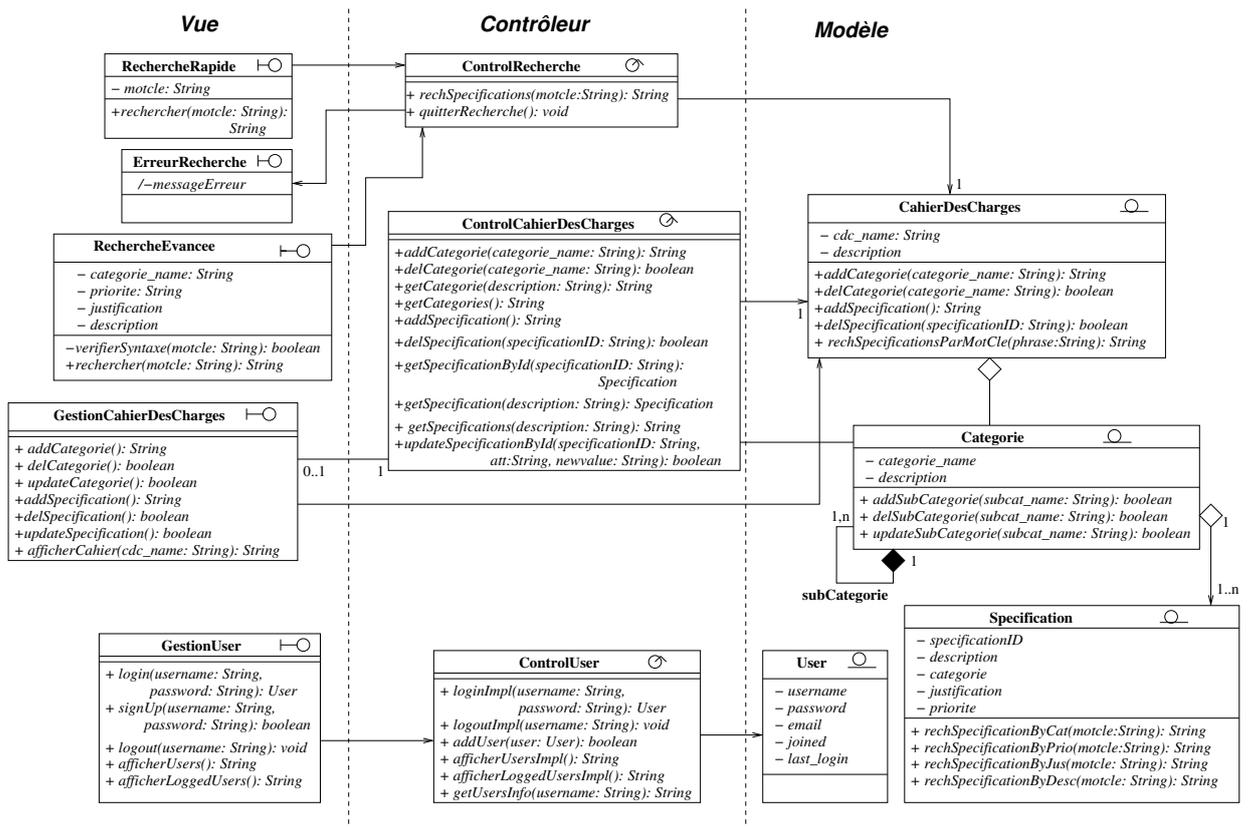


FIGURE V.2 – Diagramme des classes de conception de SpecJ

Notre application SPECJ est une application Web dont l’architecture est du type MVC2 (Modèle-Vue-Contrôleur) comme montré sur la figure V.2. Selon cette architecture, une application Web peut être découpée en trois parties : (a) le *Modèle* représente les données de la

logique métier qui sont dans une base de données, (b) la *Vue* représente l'IHM ou l'interface de l'application et (c) le *Contrôleur* correspond au composant de l'application qui répond à la saisie de l'utilisateur [Fél06]. Le contrôleur traduit les événements de l'IHM en modifications du modèle, puis définit la manière dont l'IHM réagit à ces événements. Nous présentons dans l'annexe 3 à la page 143 les supports techniques que nous avons utilisés pour implémenter chacun de ces composants dans SPECJ.

5.4.2 Modèle conceptuel de données de l'application

La figure V.3 montre le modèle entités-relation de la base de données sur laquelle s'appuie l'application SPECJ. Il existe une table *USERS* qui contient l'ensemble des utilisateurs ayant un compte créé pour utiliser SPECJ. La table *LOGGEDUSERS* contient des utilisateurs qui sont connectés au cours d'une session donnée. Lorsqu'un utilisateur se déconnecte, cette table est mise à jour par la suppression de l'utilisateur déconnecté de la liste des utilisateurs connectés. La table *CATEGORIE* contient l'ensemble des catégories auxquelles peuvent appartenir une spécification donnée. Une catégorie peut avoir des sous catégories contenues dans la table *SUBCATEGORIE* qui contiennent des spécifications. La table *SPECIFICATION* contient l'ensemble des spécifications. *CAHIERDESCHARGES* est la table contenant l'ensemble des catégories avec les spécifications. Pour plus de détails sur ces tables, se référer à l'annexe 4 à la page 144 qui les décrit en MySQL.

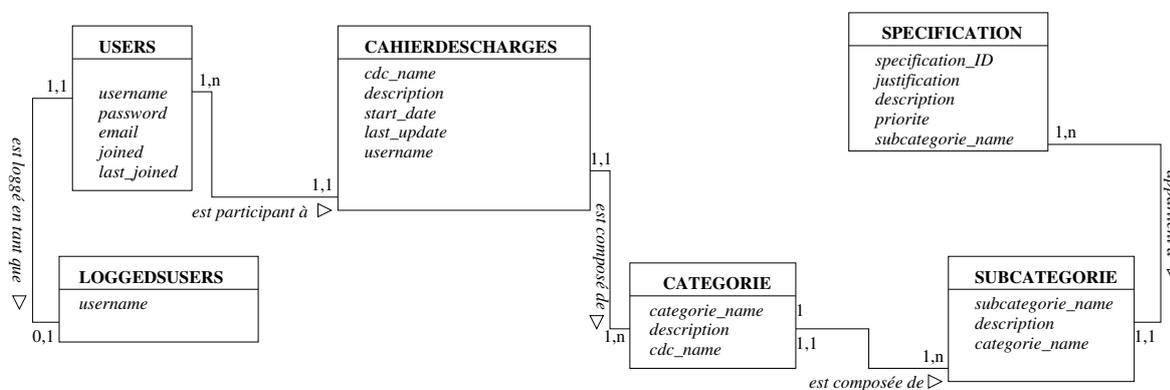


FIGURE V.3 – Modèle conceptuel de données de SPECJ.

5.4.3 Interface de SPECJ avec scénario

Les fonctions perceptibles à travers l'utilisation de l'interface actuelle de SPECJ sont :

- L'ajout de spécification.
- La suppression d'une spécification.
- La mise à jour d'une spécification.
- La recherche de spécifications par critères (catégorie, description, priorité et justification).

- La consultation de la liste des spécifications et des catégories auxquelles elles appartiennent.
- La consultation de la liste de tous les utilisateurs inscrits (qui ont ouvert un compte).
- La consultation de la liste des utilisateurs connectés actuellement.
- L’affichage du cahier des charges.
- La déconnexion automatique de l’utilisateur au bout d’une durée maximum qui peut être modifiée (un timer).

La figure V.4 présente la page d’accueil de SPECJ. Lorsque l’utilisateur est déjà inscrit et possède donc un compte, il peut directement se connecter et accéder à la session de spécification des exigences. Dans le cas contraire, il devra d’abord ouvrir un compte et se connecter ensuite pour pouvoir participer à une session de spécification. Lorsque la création de compte réussit, l’application renvoie une page à l’utilisateur avec les détails sur son compte ainsi qu’un formulaire pour la connexion, voir la figure V.5.

Welcome in SpecJ specification management tool

Log in if you are already a member.edit

Username

Password

Sign up if you are not a member yet.edit

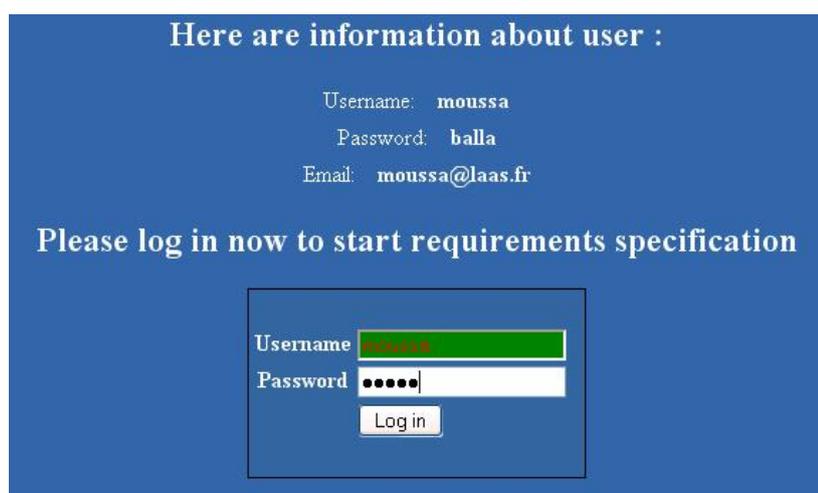
Username:

Password:

Confirm Password:

Email:

FIGURE V.4 – page d’accueil de l’application SpecJ



Here are information about user :

Username: **moussa**
Password: **balla**
Email: **moussa@laas.fr**

Please log in now to start requirements specification

Username

Password

FIGURE V.5 – Confirmation de l'inscription du nouvel utilisateur

Lorsque la connexion réussit, une page de confirmation correspondant à la page de gestion des spécifications est renvoyée à l'utilisateur. Il peut alors accéder à la liste des spécifications ou le cahier des charges pour faire des opérations qu'il souhaite (infra, la figure V.6). Plusieurs participants peuvent se connecter en même temps et accéder au cahier des charges. La figure V.7 montre trois utilisateurs (jacquie, paci et moussa) tous connectés à la même session de spécification. Ils font de la spécification collaborative.

Tout utilisateur peut accéder à la base de données de tous les utilisateurs ayant un compte sur SPECJ et obtenir des informations sur eux. Par exemples, les dates d'adhésion et de dernière connexion des utilisateurs (infra, la figure V.8).

Les utilisateurs peuvent également consulter la liste des utilisateurs connectés à la session en cours comme montré à la figure V.9.

You are logged as: **moussa**.
[Log out](#)

To view all specifications table, click [here](#)
 To view current logged users, click [here](#)
 To make research about specifications, click [here](#)
 To view specifications by categories, click [here](#)

Specification Manager

[Add A Specification](#)

Suppression	Categorie	Description	Justification	Priorite
<input checked="" type="checkbox"/>	<input type="text" value="null"/>	<input type="text" value="New Specification"/>	<input type="text" value="null"/>	<input type="text" value="null"/>
<input checked="" type="checkbox"/>	<input type="text" value="Performance"/>	<input type="text" value="Le système doit avoir un catalogue pour au moins 600000 ouvrages"/>	<input type="text" value="Des ouvrages en très grand nombre sont à la disposition des usagers"/>	<input type="text" value="HAUTE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Qualité"/>	<input type="text" value="Le système doit supprimer le mot de passe après la fin de la procédure de prêt"/>	<input type="text" value="Pour des raisons de sécurité"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Performance"/>	<input type="text" value="Le système doit gérer au moins 100000 utilisateurs"/>	<input type="text" value="La bibliothèque a de nombreux usagers"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Recherche"/>	<input type="text" value="Le système doit offrir des références pour chaque ouvrage"/>	<input type="text" value="Pour optimiser la recherche"/>	<input type="text" value="HAUTE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Recherche"/>	<input type="text" value="Le système doit donner les résultats de la recherche dans une page particulière"/>	<input type="text" value="Pour une bonne présentation du résultat"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Découverte"/>	<input type="text" value="Le système doit offrir une image de chaque ouvrage sur sa page"/>	<input type="text" value="Pour une bonne présentation du résultat"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Découverte"/>	<input type="text" value="Le système doit fournir des informations sur la quantité disponible pour le prêt"/>	<input type="text" value="Cette information peut être utile pour celui qui veut faire un prêt"/>	<input type="text" value="MOYENNE"/>

FIGURE V.6 – Confirmation de la connexion de l'utilisateur

You are logged as: **moussa**.
[Log out](#)

You are logged as: **jacou**.
[Log out](#)

You are logged as: **jacque**.
[Log out](#)

To view all specifications table, click [here](#)
 To view current logged users, click [here](#)
 To make research about specifications, click [here](#)
 To view specifications by categories, click [here](#)

Specification Manager

[Add A Specification](#)

Suppression	Categorie	Description	Justification	Priorite
<input checked="" type="checkbox"/>	<input type="text" value="Performance"/>	<input type="text" value="Le système doit avoir un catalogue pour au moins 600000 ouvrages"/>	<input type="text" value="Des ouvrages en très grand nombre sont à la disposition des usagers"/>	<input type="text" value="HAUTE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Qualité"/>	<input type="text" value="Le système doit supprimer le mot de passe après la fin de la procédure de prêt"/>	<input type="text" value="Pour des raisons de sécurité"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Performance"/>	<input type="text" value="Le système doit gérer au moins 100000 utilisateurs"/>	<input type="text" value="La bibliothèque a de nombreux usagers"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Recherche"/>	<input type="text" value="Le système doit offrir des références pour chaque ouvrage"/>	<input type="text" value="Pour optimiser la recherche"/>	<input type="text" value="HAUTE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Recherche"/>	<input type="text" value="Le système doit donner les résultats de la recherche dans une page particulière"/>	<input type="text" value="Pour une bonne présentation du résultat"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Découverte"/>	<input type="text" value="Le système doit offrir une image de chaque ouvrage sur sa page"/>	<input type="text" value="Pour une bonne présentation du résultat"/>	<input type="text" value="MOYENNE"/>
<input checked="" type="checkbox"/>	<input type="text" value="Découverte"/>	<input type="text" value="Le système doit fournir des informations sur la quantité disponible pour le prêt"/>	<input type="text" value="Cette information peut être utile pour celui qui veut faire un prêt"/>	<input type="text" value="MOYENNE"/>

FIGURE V.7 – Spécification collaborative

You are logged as: **jacque**.
[Log out](#)

Overview of signed up users, current logged users and specifications

Users | **Current Logged Users** | **Specifications**

User Name	Email	Joined Date	Last Login
moussa	moussa@laas.fr	07-08-2009	07-08-2009
nabil	nabil@laas.fr	06-29-2009	07-01-2009
bomfourou	bomfourou@bomfourouland.fr	06-25-2009	06-25-2009
paci	pasca@aol.com	06-20-2009	07-08-2009
eve	eve@yahoo.fr	06-20-2009	06-24-2009
jona	jonathan@msn.com	06-20-2009	07-08-2009
evek	eve@yahoo.fr	06-20-2009	07-07-2009
pamela	pasca@aol.com	06-20-2009	06-24-2009
pascaline	pasca@aol.com	06-20-2009	06-24-2009
esther	esther@yahoo.fr	06-20-2009	07-01-2009
ema	emma@bamagnet.net	06-20-2009	07-02-2009
sara	sara@katimarche.org	06-20-2009	07-02-2009
rom	ruben@caramail.com	06-20-2009	06-25-2009
ruben	ruben@caramail.com	06-20-2009	06-24-2009
fallou	fallou@yahoo.fr	06-23-2009	07-01-2009
emilie	emilie@msn.com	06-23-2009	06-23-2009
romaric	rguiller@yahoo.fr	06-24-2009	N/A
abram	abraham@israel.is	06-24-2009	07-01-2009
isaac	isaac@israel.is	06-24-2009	06-24-2009
jacque	sire@msn.com	06-23-2009	07-09-2009

[Go back to the Specifications Management page](#)

FIGURE V.8 – Liste des utilisateurs inscrits

You are logged as: **paci**.
[Log out](#)

Users | **Current Logged Users** | **Specifications**

User Name	Email	Last Login
moussa	moussa@laas.fr	07-08-2009
jacque	sire@msn.com	07-08-2009
paci	pasca@aol.com	07-08-2009

[Go back to the Specifications Management page](#)

FIGURE V.9 – Liste des utilisateurs connectés

La figure V.10 montre l’affichage du contenu d’un cahier des charges sous forme de tableau. Chaque exigence apparait avec les informations la concernant.

Categorie	Description	Justification	Priorite
Sélection	Le système doit limiter le prêt des étudiants à trois ouvrages	Il est toujours possible de renouveler le prêt	HAUTE
Découverte	Le système doit fournir des informations sur la quantité disponible pour le prêt	Cette information peut être utiles pour celui qui veut faire un prêt	MOYENNE
Découverte	Le système doit offrir une image de chaque ouvrage sur sa page	Pour une bonne présentation du résultat	MOYENNE
Contrainte	Le système doit permettre aux usagers de mettre à jour les données personnelles	Il est important que les usagers soient joignables à leurs adresses de domicile	HAUTE
Performance	Le système doit gérer au moins 100000 utilisateurs	La bibliothèque a de nombreux usagers	MOYENNE
Qualité	Le système doit supprimer le mot de passe après la fin de la procédure de prêt	Pour des raisons de sécurité	MOYENNE
Recherche	Le système doit donner les résultats de la recherche dans une page particulière	Pour une bonne présentation du résultat	MOYENNE
Recherche	Le système doit offrir des références pour chaque ouvrage	Pour optimiser la recherche	HAUTE
Performance	Le système doit avoir un catalogue pour au moins 600000 ouvrages	Des ouvrages en très grand nombre sont à la disposition des usagers	HAUTE
Recherche	Le système doit permettre le parcours et le reclassement facile des ouvrages	Pour faciliter la recherche	MOYENNE
Recherche	Le système doit enregistrer le numéro identifiant au moment du prêt	Seuls les usagers enregistrés peuvent prêter des ouvrages	HAUTE
Qualité	Le système doit avoir une mise en page simple	Pour faciliter l'utilisation et éviter trop d'efforts	BASSE

FIGURE V.10 – Le cahier des charges sous forme de tableau

SPECJ permet également de faire des recherches sur les spécifications avec des critères portant sur leur description, catégorie, justification ou priorité, se référer aux figures V.11, V.12, V.13. Lorsque le mot clé fourni par l'utilisateur ne correspond à aucune des spécifica-

FIGURE V.11 – Page de recherche des spécifications

tions du cahier des charges, un message d’erreur est renvoyé comme montré à la figure V.14.

Dans SPECJ, il est possible de faire l’affichage de la table des catégories avec les sous-catégories et leurs descriptions (infra, la figure V.15).

You are logged as: [paul](#).
[Log out](#)

Here is the response to your request

Affichage resultat

Click [here](#) to go back precedent page

Categorie	Description	Justification	Priorite
Recherche	Le système doit enregistrer le numéro identifiant au moment du prêt	Seuls les usagers enregistrés peuvent prêter des ouvrages	HAUTE
Recherche	Le système doit permettre le parcours et le reclassement facile des ouvrages	Pour faciliter la recherche	MOYENNE
Recherche	Le système doit offrir des références pour chaque ouvrage	Pour optimiser la recherche	HAUTE
Recherche	Le système doit donner les résultats de la recherche dans une page particulière	Pour une bonne présentation du résultat	MOYENNE

FIGURE V.12 – Résultat de la recherche avec mot clé "Recherche" pour catégorie

You are logged as: [paul](#).
[Log out](#)

Here is the response to your request

Affichage resultat

Click [here](#) to go back precedent page

Categorie	Description	Justification	Priorite
Recherche	Le système doit permettre le parcours et le reclassement facile des ouvrages	Pour faciliter la recherche	MOYENNE
Qualité	Le système doit supprimer le mot de passe après la fin de la procédure de prêt	Pour des raisons de sécurité	MOYENNE
Performance	Le système doit gérer au moins 100000 utilisateurs	La bibliothèque a de nombreux usagers	MOYENNE
Recherche	Le système doit donner les résultats de la recherche dans une page particulière	Pour une bonne présentation du résultat	MOYENNE
Découverte	Le système doit offrir une image de chaque ouvrage sur sa page	Pour une bonne présentation du résultat	MOYENNE
Découverte	Le système doit fournir des informations sur la quantité disponible pour le prêt	Cette information peut être utiles pour celui qui veut faire un prêt	MOYENNE

FIGURE V.13 – Résultat de la recherche avec le mot clé "MOYENNE" pour priorité

You are logged as: [jacque](#).
[Log out](#)

Here is the response to your request

This specification seems to be not exist !

Click [here](#) to go back precedent page

FIGURE V.14 – Résultat d'une recherche portant sur une spécification qui n'existe pas.

You are logged as: [jacque](#).
[Log out](#)

Categories and sub categories of specifications

Click on a category. The sub category select box will display all of the sub categories in that categorie.
Then click on one of the descriptions to see sub categories descriptions.

Category	Sub category	Description of categories and sub categories
<input type="text" value="Fonctionnelle"/> <input type="text" value="Non Fonctionnelle"/>	<input type="text" value="CONTRAINTE"/> <input type="text" value="PERFORMANCE"/> <input type="text" value="QUALITE"/> <input type="text" value="SECURITE"/>	<input type="text" value="ENSEMBLE DES EXIGENCES EXPRIMANT CERTAINES LIMITATIONS SUR LE SYSTEME"/> <input type="text" value="ENSEMBLE DES EXIGENCES POUR ASSURER LA SECURITE SUR LE SYSTEME"/> <input type="text" value="ENSEMBLE DES EXIGENCES POUR ASSURER QUE LE SYSTEME EST EFFICACE"/> <input type="text" value="ENSEMBLE DES EXIGENCES POUR LA QUALITE DU SYSTEME"/>

FIGURE V.15 – Catégories, sous-catégories et descriptions.

Après une présentation de l’outillage dont nous disposons pour soutenir notre méthodologie, nous pouvons maintenant présenter les expérimentations et les résultats que nous avons obtenus.

5.5 Travaux Expérimentaux et études de cas

Dans cette section, nous présentons des résultats d’expériences que nous avons menées pour évaluer quelques méthodes d’Elicitation des Exigences présentées dans la première partie de l’état de l’art en s’appuyant sur l’approche que nous avons proposée dans le chapitre III.

A cet effet, nous avons mené deux études de cas avec deux groupes d’étudiants. Le premier groupe était un groupe de quarante quatre (44) étudiants et le second était un groupe de quatorze (14) étudiants. Ces études de cas ont également fait l’objet des travaux que nous avons présentés dans un article [KSK09].

5.5.1 Démarche

Selon le modèle conceptuel que nous avons présenté dans la section 3.3, nous considérons d’abord le processus d’ingénierie qui est le processus d’Elicitation des Exigences dans notre cas. Ensuite, nous avons choisi certaines méthodes collaboratives existant pour le processus d’Elicitation des Exigences. Dans nos études de cas, nous avons choisi le *Brainstorming* [Osb48], *LeafHopper* [BdV03], *Storytelling* [AG06] et le *Dialogue* [Hoa07]. Parmi ces quatre méthodes d’éllicitation, il existait déjà pour le *Brainstorming* et le *LeafHopper* les *thinkLets* respectifs *FreeBrainstorm* et *LeafHopper* dont les descriptions sont accessibles dans la librairie des *thinkLets* proposée par Briggs et de Vreede [BdV03].

Puisqu’il n’existait pas encore de *thinkLets* pour les méthodes d’éllicitation que sont le *Storytelling* et le *Dialogue*, nous les avons conceptualisées en *thinkLets*. Nous avons donc proposé deux nouveaux *thinkLets* dont les noms sont inspirés des méthodes qu’ils supportent. Ces nouveaux *thinkLets* sont présentés dans le menu détail dans l’annexe 1 à partir de la page 137. Bien que toutes ces méthodes soient utilisées pour l’Elicitation des Exigences, les tâches exécutées varient d’une méthode à une autre. Une fois que les *thinkLets* sont identifiés, nous concevons un processus collaboratif pour supporter l’Elicitation Collaborative des Exigences. La figure V.16 montre les processus de collaboration conçus par l’ingénieur de collaboration qui fut aussi le facilitateur des deux sessions.

Nous avons utilisé *ThinkTank* [Gro08] développé par GroupSystems comme outil de collaboration pour supporter la génération d’idées et la collaboration. *ThinkTank* est un outil basé sur le Web supportant les principales activités d’un processus de collaboration. Ainsi, il permet aux utilisateurs de réfléchir, de voter, d’organiser et de réviser leurs idées en mode collaboratif. *ThinkTank* permet aussi l’enregistrement des données d’une réunion électronique.

Nous présentons ci-après, les différentes étapes suivies lors de nos deux sessions au cours desquelles nous avons employé les processus de collaboration présentés dans la figure V.16.

Début : le facilitateur a commencé la session en se présentant lui-même et en expliquant le but de la session. La session était anonyme, c'est-à-dire que chaque participant peut se connecter au GSS pour contribuer sous un identifiant fictif. Ceci permettait aux participants d'être libres dans l'expression de leurs idées.

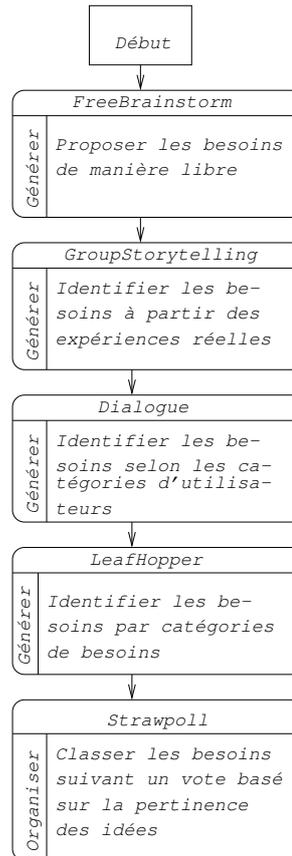


FIGURE V.16 – Processus de collaboration pour l'élicitation dans les études de cas

FreeBrainstorming : durant quinze minutes, les participants produisaient des idées relatives aux exigences du système. Le facilitateur devait souvent intervenir pour guider les participants dans l'expression de leurs idées. Le facilitateur supprimait des idées qui ne contribuaient pas à l'élicitation des exigences, comme par exemple "Cela est vrai" et "Je suis d'accord". A la fin de cette étape, nous avons eu une liste d'exigences système à laquelle tous les participants ont contribué.

Storytelling : lors de cette étape, le facilitateur demandait aux participants de réfléchir à partir du point de vue d'une partie prenante en racontant une histoire relative à ce qui lui est arrivé, en leur demandant de donner des cas d'utilisation à partir de la perspective de cette partie prenante. En d'autres termes, ils devaient donner une histoire selon différentes

perspectives, ce que le système doit faire et quels types de problèmes ou de défis sont rencontrés en faisant cela. Ensuite, le facilitateur leur a demandé de lire les histoires des autres participants et d'essayer de les élaborer. Après 15 minutes passées à raconter des histoires, le facilitateur a demandé de les résumer en termes de besoins uniquement. Finalement, les besoins ont été extraits des histoires.

Dialogue : le facilitateur demandait aux participants d'échanger des besoins à partir de différents points de vue des parties prenantes. Par exemple, il leur a été demandé de faire des dialogues entre les différentes parties prenantes impliquées. Cette étape a duré environ 10 minutes. Sur la base des dialogues resultants, le résumé des besoins a été fait.

LeafHopper : lors de cette étape, les participants fournissaient leurs besoins selon les catégories fonctionnelles liées au système. Environ huit minutes plus tard, lorsque le facilitateur a remarqué que les participants ne contribuaient plus à de nouveaux besoins, il leur a demandé d'arrêter.

Une fois que toutes les méthodes ont été exécutées, le facilitateur a demandé aux participants de résumer tous les besoins et d'essayer de trouver ceux qui manquaient. Par conséquent, les participants ont répondu aux questions qui visaient à assurer la complétude des exigences. Dans le but d'avoir des exigences plus complètes, le facilitateur essayait de motiver les participants à élaborer les exigences. Certaines incitations à l'élaboration sont : *"les exigences sont liées à la sécurité"*, *"les exigences sont aussi liées à la technologie"*. Cette discussion plénière a duré approximativement vingt minutes.

Strawpoll : les participants devaient sélectionner les dix (nombre pouvant changer selon le désir du groupe) plus importantes exigences parmi celles qui ont été générées. Pour ce faire, le thinkLet *Strawpoll* a été utilisé à l'aide de la fonctionnalité de vote du GSS. Cela a produit une liste de besoins ordonnés selon la perception de l'importance des besoins pour les participants. L'importance des besoins se mesure à selon les priorités suivantes : *Haute (fonctionnalité obligatoire)*, *Moyenne (fonctionnalité utile)* et *Basse (fonctionnalité facultative)*. Les besoins obligatoires doivent être pris en compte dans le système final, puis viennent ceux qui sont utiles et enfin ceux qui sont facultatifs. Finalement, les participants ont sélectionné un ensemble de dix besoins clés.

Les deux sessions ont été ainsi exécutées. Il y eût certaines différences dans la durée des étapes entre la première et la deuxième session. La raison réside dans la différence de la taille des groupes. Le premier groupe avait plus de participants que le second. Les étapes du premier groupe ont été plus longues car il eut beaucoup plus de contributions et de discussions.

Après les sessions, les étudiants ayant participé ont rempli nos questionnaires pour l'évaluation qualitative et quantitative des méthodes utilisées dont nous présentons les résultats ci-dessous.

5.5.2 Résultats

Le tableau V.1 présente les résultats des questionnaires remplis par les participants de la première et de la seconde session.

Dans chaque session, il y avait un facilitateur qui est aussi un spécialiste de l'Ingénierie des Exigences. Quarante et quatre (44) étudiants ont participé à la première session, mais nous n'avons reçu que 34 formulaires dûment remplis dont 23 avaient déjà participé à une session d'Elicitation des Exigences. Dans la seconde session, 14 étudiants ont participé et nous avons eu 10 formulaires correctement remplis par les participants dont un seul n'avait jamais participé à une session d'Elicitation des Exigences. Nous avons donc reçu 44 formulaires dûment remplis à l'issu des deux sessions.

1 = Fort désaccord 7 = Fort accord

Méthodes	Mesures avec $n=44$	
	Moyennes	écart-types
a- La méthode était facile à apprendre		
Brainstorming	6	1.17
Dialogue	5.32	1.36
Storytelling	4.93	1.42
LeafHopper	5.43	1.4
b- Les règles et les instructions de la méthode étaient claires		
Brainstorming	5.55	1.23
Dialogue	4.61	1.28
Storytelling	4.34	1.4
LeafHopper	4.98	1.25
c- La méthode était efficace, elle demandait moins d'effort/temps pour les maximum de résultats		
Brainstorming	5	1.6
Dialogue	4.39	1.46
Storytelling	4.11	1.48
LeafHopper	4.89	1.54
d- La méthode aidait beaucoup pour exprimer les résultats		
Brainstorming	5.11	1.35
Dialogue	4.8	1.27
Storytelling	4.61	1.48
LeafHopper	5.23	1.35
e- Les exigences générées avec cette méthode étaient utiles pour notre projet de conception		
Brainstorming	5.27	1.5
Dialogue	4.84	1.26
Storytelling	4.57	1.39
LeafHopper	5.23	1.28
f- Les exigences générées avec cette méthode étaient de haute qualité		
Brainstorming	4.16	1.36
Dialogue	4.27	1.29
Storytelling	4.14	1.22
LeafHopper	4.98	1.39
g- Cette méthode a aidé à assurer la complétude des exigences		
Brainstorming	4.27	1.6
Dialogue	4.41	1.42
Storytelling	4.27	1.34
LeafHopper	4.82	1.39
h- Cette méthode a aidé à assurer la rigueur des exigences		
Brainstorming	4.25	1.38
Dialogue	4.36	1.3
Storytelling	4.5	1.2
LeafHopper	4.98	1.32

TABLE V.1 – Résultats de la première et de la deuxième session

Les questionnaires ont été répondus par des étudiants de master en Ingénierie Système et en analyse et gestion des politiques pour la technologie. Ils avaient donc une connaissance des méthodes et techniques d'Ingénierie des Exigences. Par conséquent, leur jugement sur la qualité, la complétude, etc. des exigences n'était pas insignifiant. En outre, puisque l'évaluation visait plutôt la comparaison des méthodes, les réponses n'étaient pas absolues. Ces réponses ont cependant permis d'avoir un aperçu intéressant sur la valeur des méthodes utilisées.

En fin de compte, nous constatons que les étudiants ont trouvé que FreeBrainstorm et LeafHopper sont plus pratiques tandis qu'ils ont jugé Dialogue et Storytelling plus élaborés du point de vue de la rigueur et de la complétude. Quant à la clarté, FreeBrainstorm (5,55) et LeafHopper (4,98) ont été préférés. Il en est de même que pour l'efficacité et de l'aide qu'apportaient les méthodes (FreeBrainstorm (5,00, 5,11) et LeafHopper (4,89, 5,23)).

En ce qui concerne l'utilité et la qualité des exigences obtenues, LeafHopper a obtenu le meilleur résultat (5,23 et 4,98). Des explications possibles sont : LeafHopper fut la dernière méthode exécutée, ou les participants n'étaient plus inspirés pour contribuer à de nouvelles exigences. Le score élevé pour l'utilité et la qualité s'explique quelque part par les instructions du thinkLet LeafHopper qui demandaient des exigences plus spécifiques, et ceci en focalisant le groupe sur les aspects fonctionnels de la conception. Pour la complétude et la rigueur, les meilleurs résultats ont été attribués à LeafHopper (4,82, 4,98) bien que les méthodes, Dialogue (4,41, 4,36) et Storytelling (4,27, 4,5) aient également obtenu de très bons résultats. Ainsi, les participants ont estimé que Dialogue et Storytelling sont des méthodes plus élaborées pour assurer une exploration complète des besoins provenant de différentes parties prenantes.

Certaines limites de la présente étude sont les suivantes :

- Les exigences obtenues par les étudiants au cours de cette étude n'ont pas été utilisées dans une session ultérieure. Une étude sur le terrain est donc souhaitable pour évaluer l'utilité et la qualité des exigences.
- L'évaluation de la qualité des exigences ne faisait pas partie de nos préoccupations car l'étude était basée sur un cas fictif. De fait, une évaluation réelle de la qualité des exigences n'était pas possible.
- L'ordre des thinkLets était fixé. Un ordre différent dans la séquence aurait sans doute influencé les résultats obtenus.
- Les thinkLets ont été évalués après l'exécution de l'ensemble du processus. Il est vrai que cela permet une comparaison des méthodes, mais cela pouvait également provoquer une perte de précision dans l'évaluation des méthodes, en particulier celles qui ont été utilisées en premier.

Conclusion générale

Le travail rapporté dans ce document porte initialement sur la problématique d'acquisition des exigences. Nous avons justifié le choix du cadre de nos travaux qui porte sur l'Ingénierie Système pour ne pas limiter leur application aux systèmes informatiques, mais aussi à tout autre type de système. Afin d'adhérer aux bonnes pratiques, nous avons eu recours à une norme d'Ingénierie Système, EIA-632.

L'Elicitation des Exigences a été posée comme contexte de nos travaux. Compte tenu du caractère très critique de ce processus, et du fait qu'elle requiert l'implication de plusieurs parties prenantes, une approche collaborative a été adoptée.

Un ensemble d'approches et techniques pour l'Elicitation Collaborative des Exigences ont été passées en revue dans le but de situer nos travaux par rapport à l'existant. Il en est ressorti que ces approches, bien qu'ayant de nombreux mérites, ne traitent pas particulièrement l'aspect collaboratif et elles sont limitées lorsque le nombre de participants augmente considérablement. Sur cette base, la démarche adoptée dans nos travaux a séparé les aspects de collaboration des aspects de l'élicitation (les tâches d'ingénierie qui composent le processus d'Elicitation des Exigences). Cette séparation des problèmes a été faite dans le but de traiter les deux dimensions de l'Elicitation Collaborative des Exigences (les tâches d'élicitation et la collaboration) avec une attention particulière.

L'état de l'art s'est porté sur l'étude de la collaboration en général avec des notions qui y sont rattachées : la *Coopération*, la *Coordination*, la *Communication*. Deux domaines de recherche qui s'intéressent particulièrement à la collaboration, l'*Intelligence Collective* et l'*Ingénierie de la Collaboration* ont également été abordés. A travers l'Intelligence Collective, les vertus de la collaboration dans la résolution de problèmes complexes ont été mises en évidence. L'Ingénierie de la Collaboration, quant à elle, a fourni des moyens pour construire cette collaboration.

La méthodologie qui a été proposée comporte plusieurs étapes : "Identification des parties prenantes pertinents et présentation de la liste initiale des besoins", "Enrichissement et raffinement de la liste initiale des besoins", "Analyse des besoins", "Transformation des besoins en exigences techniques", "Hiérarchisation des exigences" et "Vérification et Validation des exigences".

La méthodologie repose sur l'approche de l'Ingénierie de la Collaboration. En effet, l'Ingénierie de la Collaboration utilise les patrons de conception appelés thinkLets qui sont des briques de base pour la conception de processus de collaboration. Ces processus de collaboration sont conçus pour que les participants à l'Elicitation Collaborative puissent les exécuter autant de fois qu'il le désirent. En outre, certains avantages des thinkLets sont qu'ils permettent de

prévoir les résultats attendus et qu'ils peuvent être utilisés par des personnes n'ayant pas de compétences dans la résolution des problèmes de collaboration.

L'une des contributions significatives de nos travaux est la proposition de deux nouveaux thinkLets qui vont être rajoutés à la librairie des thinkLets qui en compte à ce jour environ 70.

Pour valider la méthodologie proposée, elle a été mise en œuvre à l'aide d'un environnement de travail collaboratif appelé Thinktank. Thinkthank est un GSS (*Group Support System*) qui permet la génération collaborative d'idées et qui offre de nombreuses fonctionnalités pour traiter les données collectées. Parmi ces fonctionnalités, il y a le vote et d'autres calculs statistiques.

Une autre contribution majeure de nos travaux est la conception et le développement d'un prototype appelé SPECJ. Ce prototype est un GSS pour supporter la spécification collaborative des exigences. SPECJ permet donc d'établir un cahier des charges conformément au modèle que nous avons développé.

Les travaux réalisés au cours de cette thèse ont ouvert de nombreuses perspectives, dont en particulier la réalisation de nouvelles études de cas pour l'évaluation complète et l'amélioration de la méthodologie. Ces travaux s'adresseront en particulier aux limites identifiées dans l'étude de cas, à savoir : l'application des méthodes à des situations réelles et leur évaluation par des experts, l'étude des effets liés à la combinaison des thinkLets, etc. D'autres travaux sont envisagés pour ajouter de nouvelles fonctionnalités au prototype SPECJ afin d'en faire un GSS plus robuste pour l'Elicitation Collaborative des Exigences.

D'autres perspectives possibles sont : l'application de l'approche à d'autres normes d'Ingénierie Systèmes (IEEE 1220, ISO 15288), l'extension de la solution afin qu'elle couvre toutes les étapes du processus d'Ingénierie des Exigences et l'extension de la méthodologie à d'autres thématiques émergentes comme la *Maintenance Collaborative*.

Références

- [AD06] P. Atelin and J. Dordoigne. *Réseaux informatiques : Notions fondamentales, Normes, Architecture, Modèle OSI, TCP/IP, Ethernet, Wi-Fi,...* Eni, 2006.
- [AEF⁺00] E. Arias, H. Eden, G. Fisher, A. Gorman, and E. Scharff. *Transcending the individual human mind—creating shared understanding through collaborative design*. ACM Transactions on Computer-Human Interaction, 7 :84–113, March 2000.
- [AFI07a] AFIS. *Ingénierie Système, Pourquoi ? Comment ?* Technical report, Association Française d’Ingénierie Système, <http://www.afis.fr/doc/presIS/presIS.htm>, 2007.
- [AFI07b] AFIS. *Maîtriser l’ingénierie des exigences : L’expérience de déploiement chez PSA Peugeot Citroën*. Technical report, Association Française d’Ingénierie Système, <http://www.afis.fr/upload/SDD/RECHERCHE/>, accédé en 2007.
- [AG06] C.A. Acosta and L. A. Guerrero. *Supporting the Collaborative Collection of User’s Requirements*. In Proceedings of International Conference of Group Decision and Negotiation (GDN), Germany, 2006.
- [AIS⁺77] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-king, and S. Angel. *A Pattern Language*. Oxford University Press, New york, 1977.
- [Aka90] Y. Akao. *Quality Function Deployment : Integrating Customer Requirements into Product Design*. Productivity Press, Cambridge, MA, 1990.
- [Ale97] I. Alexander. *Scenario Plus : A fresh approach to Requirements*. <http://www.scenarioplus.org.uk/index.htm>, accédé en Sept. 2008, 1997.
- [All99] Electronic Industries Alliance. *EIA-632 : Processes for systems engineering*. ANSI/EIA-632-1998 Approved : January 7, 1999.
- [Ann01] Kofi Annan. Extrait du discours à l’Assemblée générale de l’ONU, 2001.
- [AV93] V. De Antonellis and L. Vandoni. *Validation of object-oriented dynamic specifications*. In Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences, volume 4, pages 399–408, Wailea, HI, USA, 5-8 Jan. 1993.
- [Bô08] D. Bô. *Intelligence Collective : quels apports pour les études marketing ?* <http://testconso.typepad.com/marketingetudes/2008/01/intelligence-co.html>, accédé en mai 2008.
- [BAM⁺99] R.O. Briggs, M. Adkins, D. Mittleman, J. Kruse, S. Miller, and J.F. Nunamaker. *A technology transition model derived from field investigation of GSS use aboard the U.S.S. Coronado*. Journal of Management Information Systems, 15(3) :151–195, Winter 1998-1999.
- [BBHL94] B. Boehm, P. Bose, E. Horowitz, and M. J. Lee. *Software Requirement as Negotiation win Conditions*. In Proceeding of International Conference of Requirements Engineering, Colorado Springs, CO, USA, 1994. IEEE.
- [BdV01] R. Briggs and G. J. de Vreede. *ThinkLets : Building Blocks for Concerted Collabaoration*. GroupSystems, 2001.
- [BdV03] R. O. Briggs and G.-J. de Vreede. *ThinkLets. Building Blocks for Concerted Collaboration*. GroupSystems.com, (in press), 2003.
- [BdVJ03] R.O. Briggs, G.J. de Vreede, and J.F. Nunamaker JR. *Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems*. Journal of Management Information Systems, 19(4) :31–64, 2003.

- [BdVJT01] R. Briggs, G. de Vreede, J. Nunamaker Jr, and D. Tobey. *ThinkLets : Achieving Predictable, Repeatable Patterns of Group Interaction with Group Support Systems (GSS)*. In Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS-34), volume 1, Hawaii, USA, 2001. HICSS.
- [BGB01] B. Boehm, P. Grünbacher, and R. O. Briggs. *Developing groupware for requirements negotiation : Lessons Learned*. IEEE software, 18(3) :46–55, 2001.
- [BKdV06] R.O. Briggs, G. L. Kolfschoten, and G.J. de Vreede. *Instrumentality theory of consensus*. In Proceedings of Hawaii International conference in System Science, Symposium of Case and Field Studies of Collaboration, Hawaii, USA, 4-2 january, 2006.
- [BKGJdV06] R.O. Briggs, G.L. Kolfschoten, and D. L. Dean G.-J. de Vreede. *Defining Key Concepts for Collaboration Engineering*. In Proceedings 12th Americas Conference on Information Systems, Acapulco, Mexico, Au. 4th-6th 2006.
- [BMA⁺96] J.-A. Bartoli, J.-L. Le Moigne, F. Adreit, J.-R. Alcaraz, S. Amabile, P. Dehaene, F. Lacroux, A.-M. Nicot, and M.-J. Avenier. *Organisation Intelligente et Système d'Information Stratégique*. ECONOMICA, Paris, 1996. Collection GESTION.
- [Boe88] Barry W. Boehm. *A Spiral Model of Software Development and Enhancement*. IEEE Computer, 21(5) :61–72, 1988.
- [BP08] J.P. Bourey and H. Pingaud. *L'interopérabilité des systèmes d'entreprise : de la reconnaissance d'un besoin au développement des premières solutions*. In Colloque sur l'interopérabilité des applications d'entreprise, Ecole des Mines d'Albi-Carnaux, France, 4-5 Juin 2008.
- [Bri06] R. O. Briggs. *On theory-driven design and development of collaboration systems*. International Journal of Human-Computer Studies, 64 :573–582, 2006.
- [Buj05] R. Bujold. *Ingénierie des Exigences, L'outil de support "GenSpec"*. Revue Canadienne IEEE, Automne 2005.
- [Cai01] D. Caillard. *Intelligence Collective*. [http : //barthes.ens.fr/scpo/Presentations00-01/Caillard_IntelligenceCollective/intcol.htm](http://barthes.ens.fr/scpo/Presentations00-01/Caillard_IntelligenceCollective/intcol.htm), ENS, 2001.
- [Cam08] Cambridge University Press. *Cambridge dictionaries online*. [http : //dictionary.cambridge.org/](http://dictionary.cambridge.org/), accédé en september 2008.
- [CD05] V. Chevrier and A. St Dizier. *L'intelligence en essaim ou comment faire complexe avec du simple ?* INRIA Interstices, 2005. [http : //interstices.info](http://interstices.info).
- [Cen84] Defence Technical Information Center. *Results of Demonstration of JRIM Prototype Data Base*. Technical Report Accession ADA139772, CACI Inc-Federal Arlington VA Systems Requirements and Development Departement, 1984.
- [Com90] E.R. Comer. *Domain analysis : a systems approach to software reuse*. In Proceedings of Digital Avionics Systems Conference, pages 224–229, Grenoble, France, 15-18 Oct. 1990. IEEE/AIAA/NASA 9th.
- [Cou07] C. R. Coulin. *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation*. Thèse de doctorat, University of technology, Sydney and Université Paul Sabatier, Toulouse, LAAS-CNRS, Toulouse, France, 2007.
- [Dav86] F.D. Davis. *Technology Acceptance Model for Empirically Testing New End-User Information Systems : Theory and Results*. Cambridge, MA : Sloan School of Management, MIT, 1986.

- [Dav93] F.D. Davis. *User acceptance of information technology : System characteristics, user perception and behavioral impacts*. International Journal of Human-Computer Studies, 38(3) :475–487, 1993.
- [Dav00] A. Davis. *Just Enough Requirements Management : Where Marketing and Development Meet*. Dorset House, New York, USA, 2000.
- [DD97] G. Doman and J. Doman. *J'apprends À Lire À Mon Bébé, La Révolution Douce*. Retz, Janvier 1997.
- [DHNV90] A.R. Denis, A.R. Heminger, J.F. Nunamaker, and D.R. Vogel. *Bringing automated support to large groups : the Burr-Brown experience*. Information and Management, 18 :111–121, 1990.
- [dHvdKA04] M. den Hengst, E. van de Kar, and J. Appelman. *Designing Mobile Information Services : User Requirements Elicitation with GSS Design and Application of a Repeatable Process*. In Proceedings of the 37th Hawaii International Conference on System Sciences, Hawaii, 2004.
- [dM96] G. Terssac (de) and B. Maggi. *Autonomie et conception*. In TERSSAC (de) G. et FRIEDBERG E. (Eds), *Coopération et conception*. Octarès, Toulouse, France, 1996.
- [DP94] G. DeSanctis and M. S. Poole. *Capturing the complexity in advanced technology use : Adaptive structuration theory*. Organisation Science, 5(2) :121–147, 1994.
- [dTIdE07] GTIE : Groupe de Travail Ingénierie des Exigences. *Ingénierie des Exigences*. Technical report, Association Française d'Ingénierie Système, <http://www.afis.fr/nav/gt/ie/ie.html>, accédé en 2007.
- [DTZ08] A. Drogoul, J.-P. Treuil, and J.-D. Zucker. *Modélisation et simulation à base d'agents : Exemples commentés, outils informatiques et questions théoriques*. Dunod, 2008.
- [dVB05] G.-J. de Vreede and R. O. Briggs. *Collaborative Engineering : Designing Repeatable Processes for High-Value Collaborative Tasks*. In Proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii, 2005.
- [dVKB06] G.J. de Vreede, G. Kolfshoten, and R. Briggs. *ThinkLets : a collaboration engineering pattern language*. International Journal of Computer Applications in Technology, 25(2/3), 2006.
- [Ess02] Didier Essame. *La méthode B et l'ingénierie système. Réponse à un appel d'offre*. Technical report, IUT-Nantes, Université de Nantes, <http://www.iut-nantes.univ-nantes.fr/habrias/dessGledn/>, 2002.
- [FD91] J. Fiksel and M. Dunkle. *Principles of requirement management automation*. In Combined Proceedings and Workshops on the Reliability and Maintainability Computer-Aided Engineering in Concurrent Engineering, 1990 and 1991, pages 231–236, Leesburg, VA ; Ellicott City, MD, USA, 9-11 Oct. 1990 and 30 Sept.-3 Oct. 1991. System Sciences.
- [Fer95] J. Ferber. *Les Systèmes multi-agents : Vers une intelligence collective*. InterEditions, 1995.
- [FH01] J. Fjermestad and R. Hiltz. *Case and Field Studies of Group Support Systems : An Empirical Assessment*. Journal of Management Information Systems, 2001.
- [Fél06] J. C. Félicité. *Développement Java sous STRUTS, version 1.2*. Editions ENI, Nantes, France, 2006.
- [Fro06] Frost and Sullivan. *Meetings Around the World : The Impact of Collaboration on Business Performance*. Technical report, Frost and Sullivan, Verizon Business and Microsoft, <http://www.frost.com>, 2006.

- [FSDH07] A. Fruhling, L. Steinhauser, G. Hoff, and C. Dunbar. *Designing and Evaluating Collaborative Processes for Requirements Elicitation and Validation*. In Proceedings of the 40th Hawaii International Conference on System Sciences, Hawaii, USA, 3-6 January, 2007.
- [fSI03] International Organization for Standardization (ISO). *System engineering - System Life Cycle Processes*. International Organization for Standardization, 2003.
- [GB01] P. Grünbacher and R. O. Briggs. *Surfacing Tacit Knowledge in requirements negociation : Experiences using EasyWinWin*. In Proceeding of the 34th Hawaii International Conference on System Sciences, pages 224–229, Hawaii, 2001.
- [GBB07] R. Grangel, J.-P. Bourey, and A.J. Berre. *Solving Problems in the Parametrisation of ERPs Using a Model-Driven Approach*. Springer, London, 2007.
- [GHB⁺04] P. Grünbacher, M. Halling, S. Biffi, H. Kitapci, and B. W. Boehm. *Integrating Collaborative Processes and Quality Assurance Techniques : Experiences from Requirements Negotiation*. Journal of Management Information Systems, 20(4) :9–29, 2004.
- [GHJV95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Pattern*. Addison-Wesley, 1995.
- [GL93] J. Goguen and C. Linde. *Techniques for Requirements Elicitation*. In Proceedings of first IEEE International Symposium on Requirements Engineering, pages 152–164, San Diego, 4-6th January 1993.
- [GO01] D.L. Goldsmith and R. E. O'Rourke. *Knowledge-based Design for Adaptive Connectivity (KDAC)*. In Military Communications Conference, MILCOM 2001. Communications for Network-Centric Operations : Creating the Information Force, volume 2, pages 910–914. IEEE, 2001.
- [Gog94] J. A. Goguen. *Requirements engineering as the reconciliation of social and technical issues*. Oxford Univ., Computing Lab., UK., 1994.
- [Gra89] B. Gray. *Collaborating : Finding common ground for multiparty problems*. CA : Josses-Bass Publishers, 1989.
- [Gre04] C.S. Greenblat. *Principles and Practice of Gaming-simulation*. SAGE Publications Ltd, 2004.
- [Gro08] GroupSystems. *Groupsystems solutions*. [http ://www.groupsystems.com](http://www.groupsystems.com), 2006, accédé en septembre 2008.
- [GS04] G. Gronier and J.-C. Sagot. *Travail Collectif à Distance en Conception de Produits : Analyse de l'Usage d'un Collecticiel*. AIPTLF-BOLOGNA, 13^{eme} Congrès de psychologie du travail et des organisations, 2004.
- [Gul99] J. Gulliksen. *Bringing the Social Perspective : User Centred Design*. In Proceedings of the 8th international Conference on Human-Computer Interaction HCI(1), volume 1, pages 1327–1337, Munich, Germany, August 22-26, 1999.
- [HDKC06] M. Den Hengst, D.L. Dean, G. Kolfschoten, and A. Chakrapani. *Assessing the Quality of Collaborative Processes*. In Proceedings of the 39th Annual Hawaii International Conference on System Sciences, volume 1. System Sciences, 04-07 Jan. 2006.
- [Hoa07] N. Hoa. *Goal Management for a Multisession Dialogue*. In Proceedings of Information Technology Convergence ISITC, pages 301–305, Jeonju, Korea, 23-24 November 2007.

- [HV97] J. Howkins and R. Valantin. *Le Développement à l'Âge de l'Information : Quatre scénarios pour l'avenir des technologies de l'information et des communications*. CRDI : Centre de Recherche pour le Développement International, 1997.
- [IBM08] IBM. *Rational RequisitePro*. <http://www-01.ibm.com/software/awdtools/reqpro/>, accédé en Sept. 2008.
- [IEE90] IEEE. *IEEE Std 610.12-1990 Standard Glossary of Software Engineering Terminology*. IEEE, 1990.
- [INC00] INCOSE. *INCOSE System Engineering Handbook, version 2*. <http://g2sebok.incose.org/>, July 2000.
- [JCLW08] H. Junming, S. Chong, T. Liang, and W. Wanshan. *Modeling of collaborative design based on Colored Petri nets*. In Proceeding of the 7th Chinese Control Conference, pages 428–432, China, 16-18 July 2008.
- [JMM08] B. Jakimovski, B. Meyer, and E. Maehle. *Swarm intelligence for self-reconfiguring walking robot*. In IEEE Swarm Intelligence Symposium, pages 1–8, Sheraton Westport at St. Louis, Missouri, USA, 21-23 september, 2008.
- [J.O80] J.O. *Arrêté du 12 janvier 1973 dans Langue française*. Journal Officiel, page 21, 1980.
- [JZBC08] M. Jankovic, P. Zaraté, J.C. Bocquet, and J. Le Cardinal. *Collaborative Decision Making : Complementary Developments of a Model and an Architecture as a Tool Support*. Journal of Decision Support Systems Technologies, 1(1) :35–45, 2008.
- [KABC93] L. Kermad, C. Ausfelder, J.-. Bourey, and E. Castelain. *Integrative approach for a functional specification of FMS control*. Elsevier, 1993.
- [KBdV+06] G. Kolfshoten, R. O. Briggs, G.-J. de Vreede, P. H. M. Jacobs, and J. H. Appelman. *A conceptual foundation of the thinkLet concept for Collaboration Engineering*. Journal of Human-Computer Studies, 64 :611–625, 2006.
- [KdV07] G.L. Kolfshoten and G.J. de Vreede. *The Collaboration Engineering Approach for Designing Collaboration Processes*. In Collaboration Researchers' International Workshop on Groupware, Bariloche, Argentine, 2007. Lecture Notes in Computer Science (LNCS).
- [KdVBS07] G.L. Kolfshoten, G.J. de Vreede, R.O. Briggs, and H.G. Sol. *Collaboration Engineerability*. In Proceedings of the International Conference on Group Decision and Negotiation, Mt. Tremblant, Canada, Concordia University, 14-17 May, 2007.
- [KG04] D. Kulak and E. Guiney. *Use Cases, Requirements In Context*. ACM Press, September 2004.
- [Kon08] J. Konaté. *Conceptual Foundations of a Collaborative Requirements Elicitation Approach Based on ThinkLet*. In Proceedings of Multi Conference on Computer Science and Information Systems, pages 374–377, Amsterdam, Netherlands, 22-27 July 2008.
- [KPdv06] G.L. Kolfshoten, L. Pietron, and G.J. de vreede. *A training approach for the transition of repeatable collaboration processes to practitioners*. In Siefert, S. Weinhardt, C. (eds) International Conference on Group Decision and Negotiation, Universitätsverlag Karlsruhe, Karlsruhe, 2006.
- [Kru94] R.A. Krueger. *Focus groups : A practical guide for applied research*. Sage Publications, 1994.

- [KS98] G. Kotonya and I. Sommerville. *Requirements Engineering : Process and Techniques*. John Wiley and Son, Great Britain, 1998.
- [KS07] J. Konaté and A.E.K. Sahraoui. *Collaboration in Requirements Engineering Process*. In Proceedings of 13th International Conference on Concurrent Enterprising, pages 107–114, Sophia-Antipolis, France, 04-06 June 2007.
- [KS08] P.G.W. Keen and H.G. Sol. *Decision Enhancement Services : Reshearsing the Future for Decisions That Matter*. IO Press BV, Amsterdam, Netherlands, 2008.
- [KSFBY03] M. Klein, H. Sayama, P. Faratin, and Y. Bar-Yam. *The Dynamics of Collaborative Design : Insights From Complex Systems and Negotiation Research*. Journal of Concurrent Engineering Research and Applications, 11 :201–209, 2003.
- [KSK09] J. Konaté, A.E.K. Sahraoui, and G.L. Kolfshoten. *Collaborative Requirements Engineering : a Process-Centred Approach*. Journal of Systems and Software, Submitted In February 2009, In process, 2009.
- [KvdH06] G.L. Kolfshoten and S. van der Hulst. *Collaboration Process Design Transition to Practitioners : Requirements from Cognitive Load Perspective*. In Siefert, S. Weinhardt, C. (eds) International Conference on Group Decision and Negotiation, Universitätsverlag Karlsruhe, Karlsruhe, 2006.
- [Lar09] Larousse. *Encyclopédie Contributive Larousse.fr*. Larousse, <http://www.larousse.fr/encyclopedie/>, accédé en janvier 2009.
- [Lau97] R. Laughlin. *Un Univers différent*. Fayard, 1997.
- [LC93] Y.I. Liou and M. Chen. *Integrating group support systems, joint application development, and computer-aided software engineering for requirements specification*. In Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences, volume 3, pages 4–12, 5-8 Jan. 1993.
- [LEV97] Pierre LEVY. *L'intelligence Collective : Pour une Anthropologie du Cyberspace*. La Découverte, 1997.
- [LG96] J.C.S. Do Prado Leite and A.P.P. Gilvaz. *Requirements elicitation driven by interviews : the use of viewpoints*. In Proceedings of the 8th International Workshop on Software Specification and Design, pages 85–94, Schloss Velen, Germany, 22-23 March 1996.
- [LJ97] B. Lawrence and B. Johnson. The project scoping gamble. *IEEE Software*, 14 :107–109, 1997.
- [LL90] E.A. Locke and G.P. Latham. *A theory of goal setting and tasks performance*. Englewood Cliffs : Prentice Hall, 1990.
- [LS92] L. Luqi and R. Steigerwald. *Rapid software prototyping ?* In Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, volume 2, pages 470–479, Hawaii, 7-10 Jan. 1992.
- [LSADG07] A. Lefebvre, Z. Simeu-Abazi, J.P. Derain, and M. Glade. *Dignostic of the Avionic Equipment Based on Dynamic Fault Tree*. In Proceedings of International Conference on cost effective Automation in Networked Product Development and Manufacturing, pages 388–392, Monterrey, Mexique, 2007.
- [Mag97] B. Maggi. *Coopération et coordination dans et pour l'ergonomie : quelques repères*. Performances Humaines et Techniques, Hors série, 1997.
- [Mat08] G. Mathieu. *Design Patterns*. <http://www.design-patterns.fr>, accédé en décembre 2008.

- [MBG08] R.G. Machado, M.R.S. Borges, and J.O. Gomes. *Supporting the System Requirements Elicitation through Collaborative Observations*. In Proceedings of the 14th Collaboration Researchers' International Workshop on Groupware, Omaha, Nebraska, September 14-18, 2008.
- [MS00] M. J. Moore and F. M. Shipman. *A Comparison of Questionnaire-Based and GUI-Based Requirements Gathering*. In Proceedings of IEEE International Conference on Automated Software Engineering, Grenoble, France, 11-15 September, 2000.
- [MS04] V. Monford and S.Goudeau. *Web Services et Interopérabilité des SI*. Dunod, Paris, 2004.
- [NE00] B. Nuseibeh and S. Easterbrook. *Requirements Engineering : A Roadmap*. In Proceedings of International Conference on Software Engineering, ACM Press, Limerick, Ireland, 4-11 juin, 2000.
- [Nei67] U. Neisser. *Cognitive psychology*. Englewood Cliffs, NJ : Prentice-Hall, 1967.
- [Ng81] P.A. Ng. Further analysis of the entity-relationship approach to database design. *IEEE Transactions on Software Engineering*, SE-7 :85–99, Jan. 1981.
- [Nou07] J. F. Noubel. *Intelligence Collective, la révolution invisible*. TheTransitioner, publié en novembre 2004, révisé en août 2007.
- [NS03] S. Nazi and S. Shastry. *Role of Requirements Engineering in Software Development Process : An Empirical Study*. In Proceedings of 7th IEEE International Multi-Topic Conference, pages 402–407, Islamabad, Pakistan, 8-9 december, 2003.
- [oEEE99] Institute of Electrical and Inc. Electronics Engineers. *IEEE Standard for Application and Management of the Systems Engineering Process, IEEE Std. 1220*. IEEE, 22 January 1999. Approved 8 December 1998.
- [Orl92] W. J. Orlikowski. *The duality of Technology : Rethinking the concept of technology in organisations*. *Organisation Science*, 3(3) :392–427, 1992.
- [Os48] A. Osborn. *Your Creative Power*. Applied Imagination, 1957, 1963, p. 151, page 265, 1948.
- [Par06] Parametric Technnology Corporation. *Conception d'un système de collaboration*. Technical report, Parametric Technnology Corporation, [htt://www.ptc.com](http://www.ptc.com), 2005, accédé en novembre 2006.
- [PK98] T. Park and K.-J. Kim. *Determination of an optimal set of design requirements using house of quality*. *Journal of Operations Management*, 16 :569–581, 1998.
- [Pot09] Y. Potin. *Travail Collaboratif : Quand la distance permet le rapprochement*. http://www.creg.ac-versailles.fr/article.php3?id_article=206, 2007, accédé en Avril 2009.
- [PRS+00] L. Préfontaine, L. Richard, H. Sicotte, D. Turcotte, and S. S. Dawes. *New models of Collaboration for public service delivery : Worldwide Tends*. Technical report, Center for Technology in Government, http://www.ctg.albany.edu/publications/reports/new_models_wp, 2000.
- [Que08] QuestionPro. *QuestionPro : requirements anagement tool*. <http://www.questionpro.com>, 2005, accédé en September 2008.
- [Reb07] C. Rebetez. *Définition des concepts de base*. <http://tecfa.unige.ch/staf/stafi/rebetez/staf11/periode4/>, accédé en 2007.

- [RLD02] K. Ronkko, O. Lindeberg, and Y. Dittrich. *'Bad practice' or 'Bad methods' are software engineering and ethnographic discourses incompatible ?* In Proceedings of International Symposium on Empirical Software Engineering, Nara, Japon, 3-4 October 2002. IEEE.
- [RMA03] A. Rashid, A. Moreira, and J. Araujo. *Modularisation and composition of aspectual requirements*. In Proceedings of the 2nd International Conference on Aspect-Oriented Software Development, pages 11–20, Boston, Massachusetts, 2003.
- [RR07] J. Robertson and S. Robertson. *Volere requirements specification template edition 13*. Technical report, Principals of the Atlantic Systems Guild London, Aachen and New York, www.systemsguild.com, 2007.
- [RV04] P. Rocques and F. Vallée. *UML 2.0 en action, De l'analyse des besoins à la conception J2EE*. Eyrolles, 2004. ISBN 2212114621.
- [Sah05] A.E.K. Sahraoui. *Requirements Traceability Issues : Generic Model, Methodology and Formal Basis*. Journal Information Technology and Decision Making, 4(1) :59–80, 2005.
- [Sai05] F. Saidi. *Modélisation du processus de décision collective dans le cadre de la robotique mobile collective d'assistance*. Technical report, Université d'Evry Val d'Essonne, <http://www.ibisc.univ-evry.fr/pub/basilic/OUT/Publications/2005/Sai05>, décembre 2005.
- [SAS01] Z. Simeu-Abazi and C. Sassine. *Maintenance Integration in Manufacturing Systems : From the Modeling Tool to Evaluation*. International Journal of Flexible Manufacturing Systems, 13(3), juin 2001.
- [SAZ06] Z. Simeu-Abazi and N. Zerhouni. *Compte rendu de la réunion du Groupe de Travail sur la "Modélisation et l'optimisation de la Maintenance Coopérative et Distribuée" (MACOD)*. <http://univ-valenciennes.fr/GDR-MACS/>, 2006.
- [SB92] K. Schmidt and L. Bannon. *Taking CSCW seriously*. In Computer Supported Cooperative Work (CSCW), Springer Netherlands, 1(1-2) :7–40, March 1992.
- [SB05] F.M. Santoro and P. Brezillon. *Group storytelling approach to collect contextualized shared knowledge*. In Proceedings of Sixteenth International Workshop on Database and Expert Systems Applications, pages 388–392, Copenhagen, Denmark, 2005.
- [SBC96] J.L. Soubie, F. Buratto, and C. Chabaud. *La conception de la coopération et la coopération dans la conception*. In G. de Terssac and E. Friedberg (Eds.), *Coopération et conception*. Octarès, Toulouse, France, 1996.
- [SBS05] R. Soliman, R. Braun, and S. Simoff. *The essential ingredients of collaboration*. In Proceedings of the International Symposium on Collaborative Technologies and Systems, pages 366–373, Sydney, New South Wales, 15-20 May 2005.
- [Sch02] K. Schmid. *A comprehensive product line scoping approach and its validation*. In Proceedings of the 24th International Conference on Software Engineering, pages 593–603, Orlando, Florida, USA, 19-25 Mai, 2002.
- [Sif09] J. Sifakis. *Embedded Systems Research Challenges and Work Directions*. <http://www.laas.fr/laas/1-6938-40eme-Anniversaire-du-LAAS-CNRS.php>, accédé en février 2009.
- [SIN08] SINTEF. *Enterprise Modeling*. <http://www.sintef.no>, 1999, accédé en septembre 2008.

- [Sou98] J. L. Soubie. *Modelling in Cooperative knowledge based systems*. In Proceedings of third International Conference on the Design of Cooperative Systems, pages 45–48, Cannes, France, 26-29 May 1998.
- [SP05] J. Sajaniemi and R.N. Prieto. *An investigation into professional programmers' mental representations of variables*. In Proceedings of 13th International Workshop on Program Comprehension, pages 204–210, St. Louis, Missouri, USA, 15-16 May 2005. IEEE.
- [Swe88] J. Sweller. Cognitive load during problem solving : Effects on learning. *Cognitive Science*, 12(2) :257–285, April-June 1988.
- [SWS89] P.S. Seligmann, G.M. Wijers, and H.G. Sol. *Analyzing the Structure of IS Methodologies*. In Proceedings of the 1st Dutch Conference on Information Systems, Amersfoort, the Netherlands, 1989.
- [TdAFdM02] D. F. Togneri, R. de Almeida Falbo, and C. S. de Menezes. *Supporting Cooperative Requirements Engineering with an Automated Tool*. In Workshop on Requirements Engineering, Valencia, Spain, November 11-12, 2002.
- [TFS07] L. Teixeira, C. Ferreira, and B.S. Santos. *Using Task Analysis to Improve the Requirements Elicitation in Health Information System*. In Proceedings of IEEE 29th International Conference on Engineering in Medicine and Biology Society, pages 3669–3672, Lyon, France, 22-26 Aug. 2007.
- [TH08] A.J.C. Trappey and D. W. Hsiao. *Applying collaborative design and modularized assembly for automotive ODM supply chain integration*. *Computers in Industry*, 59 :277–287, March 2008.
- [Tha08] O. Tharan. *Linux : choisir une distribution*. <http://www.linux-france.org/article/choix-distri/>, accédé en février 2008.
- [Tru08] TrueReq. *TrueReq : requirements anagement tool*. <http://www.truereq.com>, 2004, accédé en September 2008.
- [Uni08] Standford University. *Icebreakers*. <http://osa.stanford.edu/Resources/icebreakers.htm>, accédé en septembre 2008.
- [vML99] A. von Mayrhauser and S. Lang. *A coding scheme to support systematic analysis of software comprehension*. *IEEE Transactions on Software Engineering*, 25 :526–540, July-August 1999.
- [Wal00] K. Walker. *Collaborative Power : Collaboration Processes and the Semantic Emergence of Power*. In 3rd International Conference on Critical Management Studies Stream : Communication and Collaboration, 2000.
- [Wat97] I. Watson. *Applying Case-Based Reasoning*. Morgan Kaufmann Publishers, 1997.
- [WdGM98] S.S.A. Willaert, R. de Graaf, and Simon Minderhoud. *Collaborative engineering : A case study of Concurrent Engineering in a wider context*. *Journal of Engineering and Technology Management*, 15 :87–109, March 1998.
- [WdLL07] J. Wang, B.J. d'Auriol, Y.-K. Lee, and S. Lee. A Swarm Intelligence inspired Autonomic Routing Scenario in Ubiquitous Sensor Networks. In *International Conference on Multimedia and Ubiquitous Engineering*, pages 745–750, Seoul, Korea, 26-28 Avril 2007.
- [Wei99] G. Weiss. *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*. MIT Press, USA, 1999.
- [Woo02] M. Wooldridge. *An Introduction to Multiagent Systems by Michael Wooldridge*. John Wiley and Sons, Chichester, England, 2002.

- [YB01] Y. Yuhong and A. Bejan. *Modeling workflow within distributed systems*. In Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design, pages 433–439, 12-14 July 2001.
- [YWK⁺08] D. Yang, D. Wu, S. Koolmanojwong, A. W. Brown, and B. W. Boehm. *Wiki-WinWin : A Wiki Based System for Collaborative Requirements Negotiation*. In Proceedings of the 41st Hawaii International Conference on System Sciences, Hawaii, 2008.
- [Zar05] O. Zara. *Management de l'intelligence collective*. M2 Editions, 2005.

Liste des tableaux

II.1	Modèle d'agenda [KdV07]	50
IV.1	Phase de préparation	70
IV.2	Liste initiale des besoins du système Bib-UT	71
IV.3	Résultat du Brainstorming libre	75
IV.4	Résultat de l'Analyse des besoins	80
IV.5	Résultat du "StakeholderPoll"	86
IV.6	Résultat du thinkLet "Evolution"	88
IV.7	Critères et échelle pour l'évaluation	91
IV.8	Pondération des critères par les différents rôles	91
IV.9	Résultat du "MultiCriteria"	92
IV.10	Résultat du "Red-Light-Green-Light"	93
IV.11	Résultat du premier passage de l'étape de la vérification et la validation	98
IV.12	Résultat du deuxième passage pendant la vérification et la validation	100
IV.13	Résultat du troisième passage pendant la vérification et la validation	101
V.1	Résultats de la première et de la deuxième session	119

Table des figures

I.1	Collaboration et Elicitation des Exigences.	6
I.2	Besoins et Exigences [Ess02]	9
I.3	Les enjeux de l'ingénierie des exigences. [Ess02]	10
I.4	Les phases de l'Ingénierie des Exigences. [KS98, NE00, Cou07]	11
I.5	Vue détaillée des phases de l'Ingénierie des Exigences.	13
II.1	Organisation d'une entreprise intelligente. [Zar05]	34
II.2	La représentation d'un système selon la norme EIA-632. [All99]	40
II.3	La représentation d'un processus de collaboration. [KdVBS07]	40
II.4	Diagramme de classes de processus de collaboration. [KBdV ⁺ 06, dVKB06]	44
II.5	Approche de conception dans l'Ingénierie de la Collaboration. [KdV07]	49
II.6	Une représentation graphique d'un processus utilisant des thinkLets	51
II.7	Modèle de documentation d'un processus de collaboration	52
III.1	Processus de l'EIA-632.	56
III.2	Indication du problème	56
III.3	Modèle de collaboration pour l'EIA-632	59
III.4	Structure de contrôle dans les processus de collaboration	60
III.5	Un modèle de classes des processus d'ingénierie et de collaboration	62
III.6	Processus de Collaboration pour l'Ingénierie des Exigences	64
III.7	Diagrammes de Cas d'Utilisation et d'Interaction de haut niveau pour le processus d'Ingénierie des Exigences	65
IV.1	Les étapes de la méthodologie.	68
IV.2	Etape d'identification des parties prenantes.	69
IV.3	Etape d'évolution de la liste initiale des besoins.	72
IV.4	étape de l'enrichissement de la liste initiale des besoins	73
IV.5	Etape d'analyse des besoins.	76
IV.6	Analyse des besoins	78
IV.7	Etape de transformation des besoins en exigences techniques.	81
IV.8	Les composants d'une exigence technique.	82
IV.9	Transformation des besoins en exigences techniques	84
IV.10	Etape de la hiérarchisation des exigences techniques.	89
IV.11	Hiérarchisation des exigences	90
IV.12	Etape de vérification et de validation des exigences.	94
IV.13	Vérification et Validation des exigences	96
IV.14	Exemple de spécification d'une exigence	96
V.1	Diagramme des cas d'utilisation de SPECJ.	105
V.2	Diagramme des classes de conception de SpecJ	106
V.3	Modèle conceptuel de données de SPECJ.	107
V.4	page d'accueil de l'application SpecJ	108
V.5	Confirmation de l'inscription du nouvel utilisateur	109
V.6	Confirmation de la connexion de l'utilisateur	110
V.7	Spécification collaborative	110
V.8	Liste des utilisateurs inscrits	111
V.9	Liste des utilisateurs connectés	111

V.10 Le cahier des charges sous forme de tableau	112
V.11 Page de recherche des spécifications	112
V.12 Résultat de la recherche avec mot clé "Recherche" pour catégorie	113
V.13 Résultat de la recherche avec le mot clé "MOYENNE" pour priorité	113
V.14 Résultat d'une recherche portant sur une spécification qui n'existe pas.	113
V.15 Catégories, sous-catégories et descriptions.	114
V.16 Processus de collaboration pour l'élicitation dans les études de cas	116
V.17 Agenda de la session.	139
V.18 Exécution de l'activité "Présentation Générale" de l'agenda.	140
V.19 Exécution de l'activité "Exigences Fonctionnelles" de l'agenda.	141
V.20 Exécution de l'activité "Prioritiser les Fonctions" de l'agenda.	141
V.21 Résultats de l'activité "Prioritiser les Fonctions" dans l'agenda.	142
V.22 L'activité "Vote" dans l'agenda.	142
V.23 Résultats de l'activité "Vote" dans l'agenda.	143

Annexe 1 : Description de l'approche ThinkLet

Storytelling : generate

Choose this thinkLet

- to create collective story from experiences of all participants
- to make link between participant problems and real-world situations
- when people need to understand the context of the contribution
- to gather rich and detailed contributions based on experience

Overview

In this thinkLet the group members tell stories of their experiences.

Inputs

Clear understanding of the purpose of storytelling

Outputs

A common and collective story created from experiences of all participants. This thinkLet is based on GSS allowing people to speak and write their stories. Each team member shares a story with his/her experience to elicit one perspective of a common problem. Each member can elaborate and reflect on stories from other participants. Members can also ask questions for clarification. The stories are merged to create a collective story.

How to use Storytelling

Setup

1. Create story pages in GSS
 - (a) each participant sees a page and can add part of the story
 - (b) one or two extra pages, depending on the group size
2. Present the problem that is the focus points of the story

Steps

1. Says this :
 - Says this :
 - (a) Please click on the "start button". The system will bring you to empty electronic page. Each of you has a different electronic page. You will each start on a different electronic page.
 - (b) You may each type a story on the problem presented, up to 50 words long. Then you must click the submit button to send the page back to the group.
 - (c) The system will randomly bring you a different page. That page will have somebody else's story on it.
 - (d) When you see a page with somebody else's story on it, you may respond in following ways :
 - (i) You can enrich the story with complementary points
 - (ii) You can reflect or react on the story, give comments
 - (iii) You may argue against and add your own experience if you have counter example.

- (e) You may type new episode of a story on new page. Then you must send that page back to the group. The system will bring you a new page.
- (f) We will continue swapping pages and submitting comments, suggestions and elaborations about stories until we have a complete picture of the problem.
- (g) Any questions?

Storytelling Insight

In this technique a highly elaborative way of generation is used. People share experiences about a problem or issue. The results will be detailed, but fuzzy stories, which will offer contextual information. Extensive convergence is required to integrate stories and to elicit key aspects or characteristics of the problem. Story telling can also be used in a more visionary sense, to envision future scenario's and use cases of systems that are to be developed. A challenge is to ensure that the stories are sufficiently anonymous, and that naming and blaming is avoided. Another challenge is to ensure that stories are complete and open reports of experiences, and that vagueness such as 'and we all know what that meant' or 'the rest is history' is avoided.

Dialogue : generate

Choose this thinkLet if

- Participants need to envision solutions from different perspectives
- A shared understanding of the needs and challenges of each perspective is required

Overview

Participants will engage in a dialogue in which they will assume the role of a stakeholder in the issue or topic. In this way they will look at the challenge from a different perspective.

Input

*A clearly defined topic or issue to generate reflections or contributions on
A set of stakeholder roles*

Output

A dialogue in which issues and reflection are exchanged

How to use Dialogue

Setup

1. Assign each participant a stakeholders role, ensure that they understand their role, and that they stay in their role for the duration of the activity
2. Assign each participant to a discussion page with the title of the role he or she represents

Steps

1. Ask each participant to write a brief statement from the perspective of his/her role about the issue
2. Ask participants to stay in their role and respond to one or more statements opening a dialogue. You can ask questions for clarification of the statement, reflect on it, or make a counter proposal.
3. After each contribution, return to your own page and respond, in your role to the remarks, or issues raised, keep your dialogue updated alternating between contributions to others and responding to issues raised on your own page.

Dialogue Insights

A dialogue is a natural way to engage in discussion about issues and suggestions or reflections. In this way for instance a requirements negotiation can be done. As each participant is asked to discuss the requirements from a different perspective than his own, they will gain mutual understanding of the challenges and issues of each of the perspectives.

Annexe 2 : ThinkTank

Présentation

La figure V.17 présente l'interface de définition de l'agenda qui est réalisé à l'avance par le leader de la session, et il contient les différentes étapes de la réunion ainsi que les détails qui s'y rapportent.

The screenshot shows the ThinkTank software interface. At the top, there is a navigation bar with the ThinkTank logo and the text "by GroupSystems". Below this is a "Sessions" section with a toolbar containing buttons for "New", "Delete", "Start", "Save to Disk", "Load from Disk", "Save as Template", "Report", and "Tips". A table lists sessions with columns: Name, Description, Session Type, Status, Create Date, Created By, Last Accessed On, and Last Accessed By. The table contains two rows: "Requirements Gathering Wor" and "WOEJ". Below the table are tabs for "Details", "Session Settings", "Roster", "Agenda", "Documents", and "Activity Settings". The "Agenda" tab is selected, showing a list of activities on the left and an "Action Planner" panel on the right. The activities list includes: "Présentation Générale", "Portée du projet", "Fonctions", "Prioritiser les Fonctions", "Interfaces", "Exigences Techniques", and "Vote". The Action Planner panel includes: "Action Planner", "Alternative Analysis", "Break", "Categorizer", "Placeholder", and "Rank Order Vote". A text box provides instructions: "To build your agenda, drag activities from the list above and place them in the agenda on the left. Once they are in the agenda, you can edit them to change the name, add a description or add participant instructions."

FIGURE V.17 – Agenda de la session.

Nous avons ici un agenda qui décrit un processus de sept (7) étapes pour une session de définition des exigences. Six types d'activités peuvent être définis : planification d'actions (action planner), l'analyse (Alternative Analysis), l'arrêt (Break), l'organisation (Categorizer), Placeholder et le vote (Rank Order Vote). Les types d'activités qui nous concernent le plus sont l'analyse, l'organisation et le vote qui permettent de créer, selon les configurations et les scripts, des patrons de clarification, d'organisation, de génération de réduction et de consensus.

Il est possible de faire le paramétrage des différentes activités de l'agenda en octroyant des droits aux participants pour accomplir certaines actions. Ces actions peuvent être : ajouter, éditer, supprimer et déplacer les catégories, les idées contribuées et les commentaires sur ces contributions.

Adaptation de ThinkTank à l'élicitation

Les figures V.18 et V.19 montrent l'exécution des activités "Présentation Générale" et "Exigences Fonctionnelles" de l'agenda. Il est possible de contribuer directement en donnant les idées en vrac en cliquant par exemple sur "Brainstorm" comme montré sur la figure V.18. Il est aussi possible de créer des catégories pour organiser les idées qui seront fournies comme montré sur la figure V.19. Une fois que les catégories sont créées, il est alors possible d'y inscrire les idées correspondant à chaque catégorie.

Une fois que les participants ont fourni les exigences au cours des étapes précédentes de l'agenda, il est possible de faire un vote pour choisir les exigences selon des critères de sélection. Cela se fait à l'aide d'une activité de vote, en l'occurrence "Prioritiser les Fonctions" sur la figure V.20. Les participants

The screenshot shows the thinktank software interface. The title bar indicates 'Session #49: Requirements Gathering Workshop' and 'Activity: Présentation Générale'. The interface is divided into several sections:

- Session Settings:** Includes 'Start Participants' and 'Stop Participants' buttons.
- Agenda:** A list of activities with icons and descriptions:
 - Présentation Générale:** Décrit le contexte, les contraintes et hypothèses du projet.
 - Portée du projet:** Le but, la convention, l'audience, la portée, et autres éléments pertinents pour ce projet.
 - Fonctions:** Identifier les principales exigences fonctionnelles de nouveau système.
 - Prioritiser les Fonctions:** Evaluer les fonctions demandées sur la base de priorité.
 - Interfaces:** Décrire les interfaces utilisateur, matérielles, logicielle et de communication requises pour ce projet.
 - Exigences Techniques:** Décrire les attributs de performance, de sûreté, de sécurité, de qualité, ou autres attributs requis pour ce projet.
 - Vote:** Sélectionner les 10 exigences les plus importantes.
- Main Workspace:** Contains a toolbar with 'Brainstorm', 'Organize', 'Comment', 'Transfer Data', and 'Report' buttons. Below the toolbar, it shows 'Ideas: Classes d'utilisateurs et caractéristiques' and a text area with a blue instruction: 'Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes.' Below the text area is a prompt: '(Type here to submit an idea.)'
- Dialog Box:** 'Select Users to Start' dialog box is open, showing a table of users and their current activities.

User Name	Current Activity
Abd-El-Kader	Prioritiser les Fonctions
fallou	Prioritiser les Fonctions
frede	Prioritiser les Fonctions
gsdf	Prioritiser les Fonctions
jacque	Prioritiser les Fonctions
N'importe quoi	Prioritiser les Fonctions
pasca	Vote
student7@tudelft.nl	Présentation Générale
- Bottom Panel:** Includes 'Instructions', 'Documents', 'Activity Settings', and 'Router' buttons.

FIGURE V.18 – Exécution de l'activité "Présentation Générale" de l'agenda.

doivent donc se prononcer sur les différentes contributions. Une fois que le vote est terminé, les résultats peuvent être affichés (voir figure V.21) avec le score, la moyenne et l'écart-type.

Session #49: Requirements Gathering Workshop
Activity: Fonctions

Categories: 1. Exigences Fonctionnelles 36 (35)
2. Autre 0 (0)

Ideas: Exigences Fonctionnelles

1. Je propose des enseignements individuels	0 (0)
2. supprimer les séances programmées de TD	2 (2)
3. supprimer les cours theoriques	1 (1)
4. Faire pplus de pratiques que de theories	0 (0)
5. Les cours du système ne doivent durer plus de 50 minutes	2 (2)
6. créer un site spécifique (ressources educative) pour chaque cours	1 (1)
7. Plus de stages (en entreprise) ou dans les labos	1 (1)
8. améliorer la pédagogie en enseignement supérieur, il ne me semble pas obligatoire d'apprendre à apprendre aux jeunes	1 (1)
9. Le système doit favoriser l'autonomie des étudiants	1 (1)
10. le site doit contenir les objectifs du cours, les cours (Ytransparents) + exerv'cices à faire et tous les sites connxes au cours	0 (0)
11. Utiliser les nouvelles TIC qui sont incontournables *	0 (0)
12. Le système doit former à la recherche les étudiants le plus tot possible	0 (0)
13. Développer le systeme de tutorat entre etudiants entre etudiant et professor	0 (0)
14. l'enseignant se rend disponible pour au moins 10h/semaine pour recevoir les étudiants (prise RDV sur le site, pas de tel), le RDV est individuel	1 (1)
15. associer les élèves dans les cours et surotu les fiare participer	0 (0)
16. Les horaires RDV sont programmés et fixes d'avances par exemple, indiquer les indisponibles de l'enseignant	0 (0)
17. Augmenter les heures de TP et de TD	0 (0)
18. Favoriser les rapports entre l'enseignant et les etudiants	1 (1)

FIGURE V.19 – Exécution de l'activité "Exigences Fonctionnelles" de l'agenda.

thinktank by GroupSystems

Session #49: Requirements Gathering Workshop
Activity: Prioritiser les Fonctions

Criteria=> Priority Effort
Low/Med/High Point Scales

Ballet Items	Voting Method=>	Priority	Effort
1. Je propose des enseignements individuels [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
2. supprimer les séances programmées de TD [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
3. supprimer les cours theoriques [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
4. Faire pplus de pratiques que de theories [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
5. Les cours du système ne doivent durer plus de 50 minutes [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
6. créer un site spécifique (ressources educative) pour chaque cours [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
7. Plus de stages (en entreprise) ou dans les labos [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
8. Exigences Fonctionnelles [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
8.1. Je propose des enseignements individuels [fallou]	0 (0)	L M H	1 2 3 4 5 6 7
8.2. supprimer les séances programmées de TD [Abd-El-Kader]	0 (0)	L M H	1 2 3 4 5 6 7
8.3. supprimer les cours theoriques [fallou]	0 (0)	L M H	1 2 3 4 5 6 7
8.4. Faire pplus de pratiques qua de theories [fallou]	0 (0)	L M H	1 2 3 4 5 6 7
8.5. Les cours du système ne doivent durer plus de 50 minutes [student7, student7]	0 (0)	L M H	1 2 3 4 5 6 7
8.6. créer un site spécifique (ressources educative) pour chaque cours [Abd-El-Kader]	0 (0)	L M H	1 2 3 4 5 6 7
8.7. Plus de stages (en entreprise) ou dans les labos [fallou]	0 (0)	L M H	1 2 3 4 5 6 7
8.8. améliorer la pédagogie en enseignement supérieur, il ne me semble pas obligatoire d'apprendre à apprendre aux jeunes [frede]	0 (0)	L M H	1 2 3 4 5 6 7

FIGURE V.20 – Exécution de l'activité "Prioritiser les Fonctions" de l'agenda.

Les couleurs montrent le niveau de consensus parmi les participants sur les différents points. Les contributions en vert sont ceux pour lesquels il y a un consensus dans le groupe ; et les contributions en rouge sont celles pour lesquelles il y a moins d'accords dans le groupe.

Ballot Items	Priority (1.0)	Effort (1.0)	Weighted Total	Total	Avg. Score
1. Je propose des enseignements individuels	2,00	2,00	4,00	4,00	2,00
2. supprimer les séances programmées de TD	1,00	4,00	4,00	4,00	1,67
3. supprimer les cours theriques	1,00	8,00	8,00	8,00	3,00
4. Faire plus de pratiques que de theories	2,50	2,80	7,50	7,50	3,33
5. Les cours du système ne doivent durer plus de 50 minutes	2,50	1,70	4,50	4,50	2,33
6. créer un site spécifique (ressources educative) pour chaque cours	2,50	3,40	8,50	8,50	3,67
7. Plus de stages (en entreprise) ou dans les labos	2,00	2,50	5,00	5,00	2,33
8. Exigences Fonctionnelles	2,00	3,50	7,00	7,00	3,50
8.1. Je propose des enseignements individuels	2,00	2,50	5,00	5,00	2,50
8.2. supprimer les séances programmées de TD	2,00	1,50	3,00	3,00	1,50
8.3. supprimer les cours theriques	2,00	3,00	6,00	6,00	3,00
8.4. Faire plus de pratiques que de theories	2,00	3,50	7,00	7,00	3,50
8.5. Les cours du système ne doivent durer plus de 50 minutes	2,00	2,00	4,00	4,00	2,00
8.6. créer un site spécifique (ressources educative) pour chaque cours	2,00	3,50	7,00	7,00	3,50
8.7. Plus de stages (en entreprise) ou dans les labos	2,00	2,00	4,00	4,00	2,00
8.8. améliorer la pédagogie en enseignement supérieur, il ne me semle pas obligatoire d'apprendre à apprendre aux jeux	2,00	4,50	9,00	9,00	4,50
8.9. Le système doit favoriser l'autonomie des étudiants	2,00	3,50	7,00	7,00	3,50
8.10. le site doit contenir les objectifs du cours, les cours (ytransparents) + exers*ices à faire et tous les sites connexes a	2,00	3,00	6,00	6,00	3,00
8.11. Utiliser les nouvelles TIC qui sont incontournables *	2,00	3,00	6,00	6,00	3,00
8.12. Le système doit former à la recherche les étudiants le plus tot possible	2,00	3,50	7,00	7,00	3,50
8.13. Développer le systeme de tutorat entre étudiants entre étudiant et professor	2,00	2,50	5,00	5,00	2,50
8.14. l'enseignant se rend disponible pour au moins 10h/semaine pour recevoir les étudiants (prise RDV sur le site, pas de	2,00	3,00	6,00	6,00	3,00
8.15. associer les élèves dans les cours et surout les faire participer	2,00	4,00	8,00	8,00	4,00

FIGURE V.21 – Résultats de l'activité "Prioritiser les Fonctions" dans l'agenda.

Dans le but d'obtenir un consensus dans le groupe, un vote peut être envisagé pour choisir les 10 plus importantes exigences parmi celles qui ont été proposées. Les données sont transférées vers l'activité de vote, les participants votent pour définir l'ordre de priorité en glissant les contributions soit vers le haut, soit vers le bas comme montré sur la figure V.22. Le résultat est affiché dans la figure V.23 sous forme d'histogramme. Les dix premières exigences seront donc retenues pour la spécification du cahier de charge. Il se peut qu'il n'y ait toujours pas de consensus dans le groupe concernant ce vote ; dans ce cas, d'autres votes peuvent toujours être organisés pour obtenir un consensus.

Ballot Items
1. Utiliser les nouvelles TIC qui sont incontournables * [fallou]
2. Apprendre aux étudiants à présenter leurs travaux [fallou]
3. Le système doit favoriser l'autonomie des étudiants [student7, student7]
4. Les cours du système ne doivent durer plus de 50 minutes [student7, student7]
5. Plus de stages (en entreprise) ou dans les labos [fallou]
5. Une appréciation/évaluation de l'enseignant se fera sur des critères mesurables afin de réduire la subjectivité; critère à définir avec les étudiants et feron'a l'objet d'un workshop spécifique de Thinktank [Abd-El-Kader]
7. Les horaires RDV sont programmés et fixes d'avances par exemple, indiquer les indisponibles de l'enseignant [Abd-El-Kader]
8. favoriser leur insertion dans le pole universitaire pour se sentir au mieux avec les personnes à rencontrer, notamment si l'étudiant à des difficultés comme par exemple des difficultés financiers [frede]
9. réduire le nombre d'étudiants notamment en amph, en soit bien que la moitié ne suivent pas [frede]
10. supprimer les séances programmées de TD [Abd-El-Kader]
11. La prise en parole en public [fallou]
12. Aider plus étudiants à faire leur choix des la premiere année pour la poursuite de leur carrière [fallou]
13. supprimer les cours theriques [fallou]
14. créer un site spécifique (ressources educative) pour chaque cours [student7, student7]
15. Faire plus de pratiques que de theories [student7, student7]
16. Développer le systeme de tutorat entre étudiants entre étudiant et professor [fallou]
17. Augmenter les heures de TP et de TD [fallou]
18. Favoriser les rapports entre l'enseignant et les étudiants [fallou]
19. Le système doit former à la recherche les étudiants le plus tot possible [student7, student7]

FIGURE V.22 – L'activité "Vote" dans l'agenda.

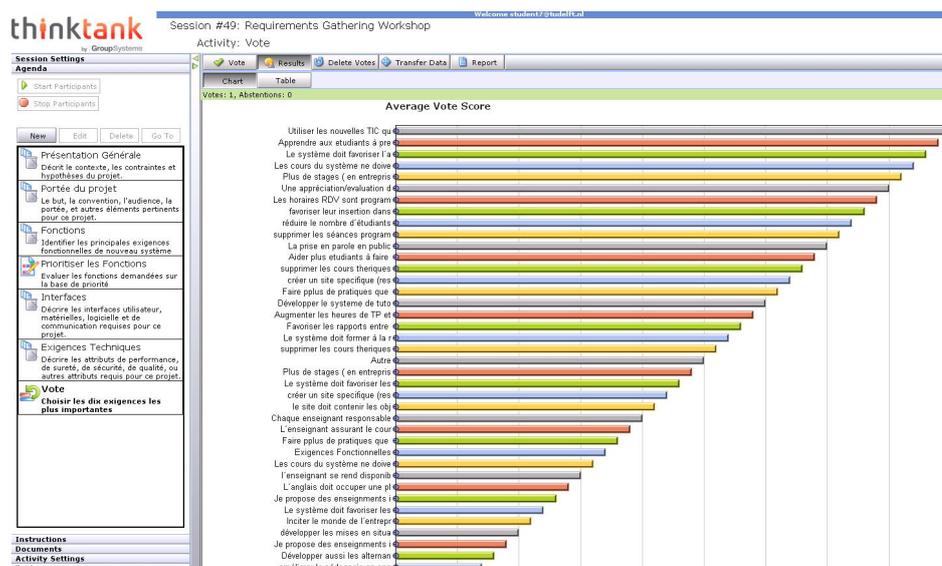


FIGURE V.23 – Résultats de l'activité "Vote" dans l'agenda.

Annexe 3 : Choix des outils utilisés dans SPECJ

L'application SPECJ est basée sur la technologie J2EE (Java to Enterprise Edition) et utilise le framework Java STRUTS avec AJAX (Asynchronone Java Script et XML), le serveur de base de données Mysql. Nous présentons brièvement ces différentes technologies et les raisons qui nous ont poussés à les choisir.

Le framework STRUTS et AJAX

Le framework STRUTS est basé sur la technologie J2EE. Struts, également considéré comme une boîte à outils, est un support de développement d'applications Web dynamiques avec java. En outre, STRUTS s'appuie sur l'une des architectures MVC (voir la section 5.4.1) les plus matures et les plus largement utilisées pour les applications Web écrites en Java. La version 1.6.0 de Java a été employée dans le développement de SPECJ.

AJAX, quant à lui, est une approche de développement pour les applications Web qui s'appuient sur différentes technologies. AJAX peut être utilisé avec d'autres langages de programmation comme le C. Nous l'avons utilisé avec Java (le Kit de développement Java, JDK), des navigateurs Web (Mozilla Firefox, Internet Explorer), Apache Ant, JavaScript et un serveur Web.

TOMCAT est le serveur d'application utilisé dans l'application SPECJ. Le serveur TOMCAT est bon moteur de servlet qui est souvent cité comme référence pour les technologies Java Servlet et JavaServer Pages (JSP). La version 6.0.18 a été utilisée pour le développement de SPECJ. Pour plus de détails sur le serveur Tomcat, se rendre sur le site <http://jakarta.apache.org/tomcat>.

JAVASCRIPT est un langage de programmation de sripts. L'avantage de ces sripts est qu'ils s'exécutent entièrement dans le navigateur. Il n'y a donc pas besoin de faire trop d'aller-retour entre le client et le serveur. Il apporte comme résultat, plus de rapidité à l'application qui l'utilise. Pour plus de détails sur JavaScript, consulter le site <http://www.javascript.com>.

ANT est un outil d'intégration, de construction et de déploiement d'applications. Il est largement utilisé dans tout type de développement Java. Il est pour Java ce que Makefile est pour le langage de programmation C. La vesrion 1.7.1 a été utilisée pour le développement de SPECJ. Les détails sur ce produits sont accessibles à l'adresse suivante : <http://ant.apache.org>.

Certains avantages de AJAX sont entres autres qu'il permet le développement des applications à moindre coût et leur confère une grande réactivité.

Le serveur de base de données Mysql

MySQL est un SGBD (Système de Gestion de Base de Données) en accès libre sur le site <http://dev.mysql.com>. La version 5.1.31 a été utilisée dans le développement de SPECJ. MySQL est téléchargeable à l'adresse <http://www.mysql.org>.

Annexe 4 : Description du modèle conceptuel de données avec MySql

- **CREATE TABLE USERS**(
USERNAME VARCHAR(50) PRIMARY KEY,
PASSWORD VARCHAR(50),
EMAIL VARCHAR(50),
JOINED VARCHAR(50),
LASTLOGIN VARCHAR(50),
UNIQUE (USERNAME),
UNIQUE (EMAIL));
- **CREATE TABLE LOGGEDUSERS**(
USERNAME VARCHAR(50) PRIMARY KEY,
CONSTRAINT fk_username foreign key references
USERS(USERNAME) on delete cascade);
- **CREATE TABLE CATEGORIES**(
CATEGORIENAME VARCHAR(50) PRIMARY KEY,
DESCRIPTION TINYTEXT,
NOMCAHIER VARCHAR(20),
CONSTRAINT fk_nomcahier foreign key references
CAHIERDESCHARGES(NOMCAHIER) on delete cascade,
UNIQUE (CATEGORIENAME));
- **CREATE TABLE SUBCATEGORIES**(
SUBCATEGORIENAME VARCHAR(50),
DESCRIPTION TINYTEXT,
CATEGORIENAME VARCHAR(50),
CONSTRAINT fk_categorienome foreign key references
CATEGORIES(CATEGORIENAME) on delete cascade);
- **CREATE TABLE CAHIERDESCHARGES**(
NOMCAHIER VARCHAR(20) PRIMARY KEY,
DESCRIPTION TINYTEXT,
STARTDATE datetime,
LASTUPDATED datetime,
USERNAME VARCHAR(50),
UNIQUE (NOMCAHIER));
- **CREATE TABLE SPECIFICATION**(
SPECIFICATIONID String PRIMARY KEY,
DESCRIPTION TINYTEXT,
CATEGORIENAME VARCHAR(30),
JUSTIFICATION TINYTEXT,
PRIORITE ENUM ('HAUTE', 'MOYENNE', 'BASSE'),
CONSTRAINT fk_categories foreign key REFERENCES
SUBCATEGORIES(SUBCATEGORIENAME) on delete cascade);

Annexe 5 : Signature de quelques méthodes utilisées dans SPECJ

Nous ne donnons pas la signature de toutes les méthodes de l'application. Seules les signatures de celles qui nous semblent être intéressantes pour la compréhension de l'architecture donnée à la section 5.4.1 sont mentionnées ci-dessus.

Classes de la couche de la Vue

Les composants de la couche de présentation sont des pages JSP (JavaServer Pages) et des classes JavaBeans de présentation héritant de la classe `ActionForm` de Struts.

– RechercheRapide

- `String rechercher(String phrase)` : renvoie l'ensemble des spécifications contenant le mot clé phrase sont renvoyées à l'issue.

– RechercheAvancee

- `boolean verifierSyntaxe(String motcle)` : renvoie vrai s'il existe au moins une exigence dont les caractéristiques contiennent le mot clé motcle.
- `String rechercher(String motcle)` : idem avec RechercheRapide sauf que motcle représente soit la catégorie, le priorité, la justification ou la description d'une spécification.

– GestionCahierDesCharges

- `String addCategorie()` : envoie d'une demande de création d'une catégorie et renvoie des informations concernant la nouvelle catégorie.
- `boolean delCategorie()` : envoie d'une demande de suppression d'une catégorie et renvoie vrai si la suppression réussit.
- `boolean updateCategorie()` : envoie d'une demande de modification d'une catégorie et renvoie vrai si la suppression réussit.
- `String addSpecification()` : envoie une demande de création d'une spécification et renvoie les informations sur la nouvelle spécification.
- `boolean delSpecification()` : envoie une demande de suppression d'une spécification et renvoie vrai si la suppression réussit.
- `boolean updateSpecification()` : envoie d'une demande de modification d'une spécification et renvoie vrai si la modification réussit.
- `String afficherCahier(String cdc_name)` : renvoie et affiche le contenu du cahier des charges d'identifiant cdc_name.

– GestionUser

- `User login(String username, String password)` : envoie une demande de connexion pour l'utilisateur de login username et de mot de passe password.
- `void logout(String username)` : envoie une demande de déconnexion pour l'utilisateur de login username.
- `boolean signUp(String username, String password)` : envoie une demande de création d'un compte pour l'utilisateur de login username et de mot de passe password.
- `String afficherUsers()` : envoie une demande d'affichage de la liste des utilisateurs ayant un compte.
- `String afficherLoggedUsers()` : envoie une demande d'affichage de la liste des utilisateurs connectés.

Classes de la couche du Contrôleur

Les classes contrôleurs sont réalisées avec servlets. Puisqu'il existe dans STRUTS une seule servlet qui représente le contrôleur (`ActionServlet`), pour ajouter de nouvelles fonctionnalités au contrôleur, il faut simplement créer de nouvelles classes qui héritent de la classe `Action` de STRUTS.

– ControlRecherche

- `String rechercheSpecifications(String motcle)` : renvoie l'ensemble des spécifications dont motcle fait partie des caractéristiques.

- *void quitterRecherche()* : met fin à la recherche et renvoie une autre page.
- **ControlCahierDesCharges**
- *String addCategorie(String categorie_name)* : crée une nouvelle catégorie d'identifiant *categorie_name* et renvoie les informations concernant cette catégorie.
 - *boolean delCategorie(String categorie_name)* : supprime la catégorie d'identifiant *categorie_name* et renvoie vrai si la suppression réussit.
 - *Categorie getCategorie(String categorie_name)* : renvoie la catégorie d'identifiant *categorie_name*.
 - *String getCategories()* : renvoie la liste des catégories.
 - *String addSpecification()* : crée une nouvelle spécification et renvoie les informations concernant cette spécification.
 - *boolean delSpecification(String specificationID)* : supprime la spécification d'identifiant *specificationID* et renvoie vrai si la suppression réussit.
 - *boolean updateSpecificationById(String specificationID, String att, String valeur)* : modifie la valeur de l'attribut *att* de la spécification d'identifiant *specificationID* et renvoie vrai si la modification réussit.
 - *Specification getSpecification(String desc)* : renvoie la spécification dont la description est *desc*.
 - *Specification getSpecificationById(String specificationID)* : renvoie la spécification dont l'identifiant est *specificationID*.
 - *String getSpecifications()* : renvoie la liste des spécifications.
- **ControlUser**
- *User loginImpl(String username, String password)* : crée une session pour l'utilisateur de login *username* et de mot de passe *password* et renvoie l'utilisateur.
 - *void logoutImpl(String username)* : ferme la session de l'utilisateur de login *username*.
 - *boolean addUser(User u)* : crée un nouvel utilisateur *u* et renvoie vrai si l'utilisateur a été créé, sinon faux.
 - *String afficherUsersImpl()* : crée et renvoie la liste des utilisateurs ayant un compte.
 - *String afficherLoggedUsersImpl()* : crée et renvoie la liste des utilisateurs connectés.

Classes de la couche du modèle

Les classes du modèle de données sont implémentées avec des *JavaBeans*. Des *EJB* ou d'autres framework de persistance pouvaient être utilisés, mais pour des soucis de simplicité, les *JavaBeans* simples ont été choisis.

- **CahierDesCharges**
- *String addCategorie(String categorie-name)* : crée une catégorie d'identifiant *categorie-name* et renvoie les informations concernant la nouvelle catégorie.
 - *boolean delCategorie(String categorie-name)* : supprime la catégorie d'identifiant *categorie-name* et renvoie vrai si la suppression réussit.
 - *boolean addSpecification()* : crée une spécification et renvoie les informations concernant la nouvelle spécification.
 - *boolean delSpecification(String specID)* : supprime la spécification d'identifiant *specID* et renvoie vrai si la suppression réussit.
 - *updateSpecificationById(String specificationID, String att, String valeur)* : modifie la valeur de l'attribut *att* de la spécification d'identifiant *specificationID* et renvoie vrai si la modification réussit.
 - *String rechSpecificationParMotCle(String motcle)* : crée et renvoie la liste des spécifications dont les caractéristiques contiennent le mot clé *motcle*.
- **Categorie**
- *boolean addSubCategorie(String subcategorie-name)* : crée la sous catégorie d'identifiant *subcategorie-name* et renvoie vrai si la création réussit.

- *boolean delSubCategorie(String subcategorie-name) : supprime la sous catégorie d'identifiant subcategorie-name et renvoie vrai si la suppression réussit.*
- *boolean updateSubCategorie(String subcategorie-name) : modifie la sous catégorie d'identifiant subcategorie-name et renvoie vrai si la modification réussit.*

– **Specification**

- *String rechSpecificationByPrio(String motcle) : crée et renvoie la liste des spécifications dont la priorité contient le mot clé motcle.*
- *String rechSpecificationByCat(String motcle) : crée et renvoie la liste des spécifications dont la catégorie contient le mot clé motcle.*
- *String rechSpecificationByJus(String motcle) : crée et renvoie la liste des spécifications dont la justification contient le mot clé motcle.*
- *String rechSpecificationByDescr(String motcle) : crée et renvoie la liste des spécifications dont la description contient le mot clé motcle.*

Pour la configuration de l'application et la connexion de tous ses éléments, il faut un fichier de configuration Struts-confi.xml.