POLYPUBLIE

# Approche de mesure et d'analyse des réunions de révision technique du processus de génie logiciel

Patrick D'Astous

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

| | |
|---|---|
| Date | 1999 |
| Type | Mémoire ou thèse / Dissertation or Thesis<br>Thèse de doctorat/Ph.D. thesis |
| URL de PolyPublie<br>PolyPublie URL | https://publications.polymtl.ca/8774/ |
| Conditions d'utilisation<br>Terms of use | © Tous droits réservés / All rights reserved |
| Référence<br>Citation | D'Astous, P. (1999). Approche de mesure et d'analyse des réunions de révision technique du processus de génie logiciel [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8774/ |

Détails de la thèse
Thesis details

| | |
|---|---|
| Directeurs de recherche<br>Advisors | Pierre N. Robillard |
| Programme<br>Program | Non spécifié |

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

# APPROCHE DE MESURE ET D'ANALYSE DES RÉUNIONS DE RÉVISION TECHNIQUE DU PROCESSUS DE GÉNIE LOGICIEL

PATRICK D'ASTOUS

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D)

(GÉNIE ÉLECTRIQUE)

MARS 1999

© Patrick d'Astous, 1999.

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

*Your file Votre référence*

*Our file Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-42820-6

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

# APPROCHE DE MESURE ET D'ANALYSE DES RÉUNIONS DE RÉVISION TECHNIQUE DU PROCESSUS DE GÉNIE LOGICIEL

présentée par: D'ASTOUS Patrick

en vue de l'obtention du diplôme de: Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. BOIS Guy, Ph.D., président

M. ROBILLARD, Pierre N., Ph.D., membre et directeur de recherche

M. CLÉMENT, Bernard, Ph.D., membre

M. LETHBRIDGE, Timothy, Ph.D., membre

*À mes parents: je vous aime.*

# REMERCIEMENTS

Ce travail de recherche n'aurait pu être complété dans sa forme actuelle sans le support continu et enthousiaste de mon directeur de thèse Pierre N. Robillard. Un gros merci aux chercheuses de l'Institut de Recherche en Informatique et Automatique (Rocquencourt, France), Mmes Françoise Détienne et Willemien Visser qui ont fourni leurs idées et connaissances aux moments opportuns. Enfin, merci à Sylvain et Sophie pour leur aide précieuse lors de la touche finale à cet ouvrage.

Finalement, il faut remercier le Conseil de la Recherche en Science Naturelle et en Génie ainsi que le Groupe Conseil DMR pour leur aide financière conjointe dans le cadre d'une Bourse CRSNG Industrielle.

# RÉSUMÉ

L'objectif du génie logiciel est de définir des pratiques et des méthodes dans le but de produire du logiciel de qualité, et ce dans les délais et les budgets prévus. Cet objectif requiert donc des ingénieurs logiciels non seulement des connaissances techniques exhaustives mais aussi des connaissances dans l'art de diriger et coordonner les ressources humaines et matérielles d'un projet.

En génie logiciel, un processus de développement identifie les pratiques individuelles et coopératives devant être utilisées par les participants à un projet de développement. Il existe deux grands types de pratiques coopératives utilisées dans le processus de génie logiciel: les réunions de travail et les révisions techniques. Les réunions de travail permettent aux spécialistes d'établir les spécifications du produit et d'échanger les mérites de solutions envisagées de façon individuelle dans le but de déterminer la meilleure marche à suivre ou la meilleure solution à un problème. Les révisions techniques sont des réunions qui apparaissent tout au long du processus de développement et qui servent à s'assurer que les spécifications du produit répondent aux besoins. Les participants aux révisions techniques parcourent un artefact afin de s'assurer qu'il n'existe aucune ambiguïté quant à son contenu.

Ce travail propose une approche structurée permettant d'observer et d'analyser les

réunions de révision technique sous un nouvel angle: la nature des activités effectuées par les individus. Le développement de logiciels requiert la coopération continuelle des individus. Cette coopération doit donc influencer toutes les pratiques du génie logiciel. Une réunion de révision technique peut donc être caractérisée par des activités individuelles et coopératives.

La démarche de recherche consiste à observer dans un environnement naturel et professionnel le comportement d'ingénieurs en logiciel, et d'élaborer, à partir de ces données, un modèle de la pratique. Une nouvelle forme de représentation des données est proposée dans ce travail. Cette représentation utilise une structure basée sur l'amalgamation de deux formalismes distincts: la linguistique et l'orienté-objet. La linguistique propose plusieurs façons de décrire les conversations tenues lors de réunions. L'unité de base est l'intervention d'un individu qui participe à un échange dans le cadre d'une séquence distincte. L'orienté-objet, par sa nature, est le formalisme utilisé pour faciliter la représentation des conversations. Les liens sémantiques entre les interventions des individus sont décrits à l'aide de classes, hiérarchies et associations. La syntaxe utilisée pour détailler cette représentation formelle permet de faciliter les analyses subséquentes et de ce fait, diminuer les coûts de ce genre de recherche. Cette syntaxe permet l'utilisation d'outils automatiques pour effectuer les mesures et analyses.

Ce travail présente en détails trois types d'analyses pouvant être effectuées à partir du codage des réunions: une analyse statique des interventions afin de déterminer la nature

des discussions, une analyse de l'influence du rôle des participants sur l'occurrence des interventions et une analyse des patrons de discussions afin de déterminer le cheminement des discussions au cours d'une réunion. Un des avantages du schéma de codage et de la syntaxe utilisée pour le représenter est l'utilisation d'outils automatiques d'analyses. L'analyse statique des interventions a été réalisée grâce au progiciel *Excel*, l'analyse de l'influence des participants à l'aide du progiciel *Statistica* et l'analyse des patrons de discussions à l'aide de logiciels développés à l'interne. Cela est un avantage indéniable puisque l'ajout de biais dans les analyses est réduit au codage seulement et non pas dans les étapes d'analyses suivantes.

La réunion de révision technique est le cadre d'un travail coopératif qui a un impact important sur le succès d'un projet de développement. Cette réunion permet aussi aux membres de l'équipe de développement d'améliorer les solutions proposées ou d'y apporter des modifications importantes. Finalement, la réunion de révision technique est une plate-forme privilégiée pour la dissémination des connaissances nécessaires au projet. Un résultat du projet de recherche est l'importance relative des échanges de synchronisation cognitive comparés à ceux de révision, qui sont pourtant l'objectif principal d'une réunion de révision technique. Cette activité constitue d'ailleurs la majorité de l'effort dépensé au cours d'une réunion de révision technique.

La représentation formelle utilisée dans cette recherche et plus particulièrement le schéma de codage des interventions, sont dissociables de l'approche exploratoire. Ils pourraient

être utilisés comme base pour d'autres types de recherche similaires où il existe le besoin de représenter les données brutes dans une syntaxe plus structurée.

Ce travail de recherche propose une nouvelle méthodologie qui pourrait avoir un impact sur le développement futur du génie logiciel. On distingue deux avenues de recherche possibles: l'utilisation de la représentation formelle dans l'étude d'autres pratiques coopératives du génie logiciel et la conception de pratiques plus "ergonomiques" de révision technique.

# ABSTRACT

The goal of software engineering is to define practices and methods to produce quality software on schedule and under budget. This requires from software engineers not only exhaustive technical skills, but also some knowledge in the art of managing and coordinating project human resources.

In software engineering, the software development process identifies individual and cooperative practices to be used by project team members. There are two main cooperative practices in software engineering: the working meeting and the technical review meeting. Working meetings enable specialists to identify product specifications and exchange opinions on previously developed individual solutions in order to come up with the best possible solution. Review meetings are held throughout the development process to ensure that the product fulfills the specifications. Review meeting participants go through a specific document to make sure there are no content ambiguities.

This document proposes a structured approach to observe and analyze technical review meetings from a new angle: the nature of the activities performed by individuals. Software development requires continuous individual cooperation. This cooperation must therefore influence all software engineering practices. It must therefore be possible to characterize a technical review meetings with individual and cooperative activities.

The research methodology consists in observing software professionals in their natural environment and, from the analyses, come up with a model of the practice. A new data representation scheme is proposed. Its structure is based on two distinct formalisms: linguistic and object-oriented. Linguistic proposes many ways from which meeting conversations can be described. The base unit is an individual intervention which participates in an exchange. The object-oriented formalism, by its nature, facilitates the representation of conversations. Semantic relationships between individual interventions are described using classes, hierarchies and associations. The syntax used to represent the structure improves subsequent analyses by enabling automatic analyses and measurement.

Three different analyses from the representation of the conversation are proposed: a static analysis of interventions to determine the nature of the discussions, an analysis of participants influence on the occurrence of there interventions and an analysis of possible patterns of discussions to determine likely subject of discussions during the meetings.

It was found that the technical review meeting is the scene of some cooperative work that has a major impact on a project success. The meeting enables team members to improve all proposed solutions and to bring major modifications if required. Finally, the technical review meeting is a unique platform to provide information dissemination throughout the project team. A result of this research is the relative importance of cognitive synchronization in this kind of meeting.

The formal representation used in this research may be dissociated from the method itself.

It could be used as a basis for other similar empirical studies where there is a need to represent raw data in a structure format.

This research proposes a new methodology which could have a major impact on the future development of software engineering. There are two possible research directions: the use of the formal representation in the study of other software engineering cooperative practices and the improvement of the technical review meeting.

# TABLE DES MATIÈRES

**DÉDICACE** ..... IV

**REMERCIEMENTS** ..... V

**RÉSUMÉ** ..... VI

**ABSTRACT** ..... X

**TABLE DES MATIÈRES** ..... XIII

**LISTE DES TABLEAUX** ..... XVIII

**LISTE DES FIGURES** ..... XIX

**LISTE DES ABBRÉVIATIONS** ..... XXI

**LISTE DES ÉCHANTILLONS** ..... XXII

**AVANT-PROPOS** ..... XXIII

**INTRODUCTION** ..... 1

**CHAPITRE I LE TRAVAIL COOPÉRATIF EN GÉNIE LOGICIEL** ..... 9

**1.1 L'approche technique au travail d'équipe** ..... 10

1.1.1 Les principes du travail d'équipe ..... 10

1.1.2 Les équipes à succès ..... 12

1.1.3 L'art du travail d'équipe ..... 13

**1.2 La prise de décision ..... 14**

1.2.1 Termes importants ..... 15

1.2.2 Approches à la prise de décision ..... 16

1.2.3 Le modèle en spirale ..... 20

1.2.4 Sommaire de la prise de décision ..... 22

**1.3 La solution de problèmes ..... 22**

**1.4 Des équipes de travail efficaces ..... 25**

1.4.1 Le modèle de diagnostique du travail ..... 25

1.4.2 L'indicateur de type Myers-Briggs ..... 26

1.4.3 Le modèle des rôles de Belbin ..... 27

**1.5 Les réunions en génie logiciel ..... 30**

**1.6 Sommaire ..... 36**

**CHAPITRE II LA RÉUNION DE RÉVISION TECHNIQUE ..... 38**

**2.1 Définition du contexte ..... 38**

2.1.1 Le produit logiciel ..... 38

2.1.2 Le génie logiciel ..... 39

2.1.3 Cycle de vie du logiciel ..... 41

2.1.4 Processus de développement ..... 44

**2.2 La Réunion de Révision Technique** ..... 45

2.2.1 L'inspection formelle et le *walkthrough* ..... 45

2.2.2 Différentes méthodes de révision technique ..... 47

2.2.3 Modèle de révision générique ..... 53

2.2.4 Variations entre les méthodes de révision ..... 54

2.2.5 Bénéfices et efficacité de la révision technique ..... 57

2.2.6 Définition générique de la réunion de révision technique ..... 59

**2.3 Travail individuel et coopératif** ..... 59

**2.4 Sommaire** ..... 61

**CHAPITRE III PROBLÉMATIQUE DE RECHERCHE** ..... 63

**3.1 Amélioration de la réunion de révision technique** ..... 64

3.1.1 Compréhension avant amélioration ..... 65

**3.2 Description des problématiques** ..... 66

3.2.1 Représentation des activités cognitives et coopératives ..... 66

3.2.2 Application des résultats au génie logiciel ..... 69

**3.3 Sommaire** ..... 71

**CHAPITRE IV DESCRIPTION DE LA MÉTHODE DE RECHERCHE** ..... 73

**4.1 Stratégie de recherche** ..... 73

4.1.1 Objectif de la recherche ..... 73

4.1.2 Approche théorique ..... 75

4.1.3 Principes fondamentaux ..... 76
4.1.4 Traditions et techniques de l'ESDA ..... 78
4.1.5 Définition et vue globale de la démarche ..... 82
4.1.6 Opérations de transformation ..... 85
4.1.7 L'aspect temporel du projet ..... 86
4.1.8 Adaptations et problèmes potentiels ..... 90
**4.2 Représentation formelle des données ..... 92**
4.2.1 Fondement théorique du schéma de codage ..... 93
4.2.2 Théorie de la linguistique ..... 93
4.2.3 Paradigme orienté-objet ..... 95
4.2.4 Représentation formelle des données ..... 98
**CHAPITRE V ÉTUDE DES RÉUNIONS DE RÉVISION TECHNIQUE ..... 114**
**5.1 Analyse statique de la nature des interventions ..... 115**
5.1.1 Mesure de l'importance relative des interventions ..... 115
5.1.2 Nature des interventions ..... 117
5.1.3 Niveaux de conversation ..... 121
**5.2 Influence du rôle des individus sur les interventions ..... 128**
5.2.1 Modélisation log-linéaire ..... 131
5.2.2 Tests d'hypothèses ..... 133
5.2.3 Outils d'analyse ..... 135
5.2.4 Ajustement d'un modèle théorique ..... 135
5.2.5 Analyse du modèle théorique ..... 142

**5.3 Identification des échanges** ..... 146

5.3.1 Lag Sequential Analysis ..... 147

5.3.2 Analyse grammaticale ..... 149

5.3.3 Utilisation de ces techniques ..... 150

5.3.4 Identification des échanges significatifs ..... 156

5.3.5 Sommaire de l'analyse des échanges ..... 165

**CHAPITRE VI MODÉLISATION DE LA PRATIQUE** ..... 167

**6.1 Utilisation des résultats en génie logiciel** ..... 167

6.1.1 Nature des échanges caractéristiques d'une réunion de révision technique 167

6.1.2 Rôle des participants à la réunion de révision technique ..... 172

6.1.3 Comparaison avec un modèle générique existant ..... 175

6.1.4 Révision technique à distance ..... 177

**6.2 Efficacité de la représentation formelle** ..... 179

6.2.1 Apport du formalisme ..... 179

6.2.2 Apport de la syntaxe ..... 180

**CONCLUSION** ..... 182

**RÉFÉRENCES** ..... 189

# LISTE DES TABLEAUX

Tableau 1.1 Le modèle de Bales ..... 19

Tableau 1.2 Processus d'émergence de décisions ..... 21

Tableau 4.1 Groupes d'interventions ..... 99

Tableau 4.2 Types de comportements lors des réunions de révision technique ..... 102

Tableau 4.3 Critères de forme et de contenu ..... 103

Tableau 4.4 Syntaxe de codage des interventions ..... 106

Tableau 5.1 Règles d'imbrication des échanges ..... 121

Tableau 5.2 Caractéristiques des participants lors d'une réunion de révision technique 130

Tableau 5.3 Tableau croisé de données de deux variables ..... 131

Tableau 5.4 Échantillon des données empiriques ..... 136

Tableau 5.5 Probabilité des interactions ..... 138

Tableau 5.6 Associations marginales et partielles ..... 139

Tableau 5.7 Exemple de valeurs théoriques ..... 142

Tableau 5.8 Résultats du LSA sur les données brutes ..... 151

Tableau 6.1 Signification des échanges ..... 168

# LISTE DES FIGURES

Figure 1.1 Le modèle Myers-Briggs ..... 27

Figure 2.1 Cycle de vie en cascade ..... 42

Figure 2.2 Modèle de révision technique générique à objectif unique ..... 54

Figure 4.1 Perspectives d'études du travail de l'ingénieur logiciel ..... 74

Figure 4.2 Approche générique de recherche ..... 75

Figure 4.3 Processus ESDA du projet ..... 83

Figure 4.4 Aspect temporel des données ..... 88

Figure 4.5 Identification des unités d'une conversation ..... 94

Figure 4.6 Diagramme de classes du schéma de codage ..... 105

Figure 4.7 Exemple d'utilisation du modèle ..... 107

Figure 4.8 Distribution des interventions dans les groupes ..... 109

Figure 4.9 Distribution des comportements du schéma de codage ..... 110

Figure 4.10 Unités d'analyses à l'intérieur d'un réunion de révision technique ..... 113

Figure 5.1 Comparaison entre le temps et le nombre de mots ..... 117

Figure 5.2 Courbe du nombre d'interventions en fonction du nombre de mots ..... 118

Figure 5.3 Distribution des groupes ..... 119

Figure 5.4 Distribution des groupes dans séquences de révision ..... 120

Figure 5.5 Distribution des interventions par niveau ..... 123

Figure 5.6 Distribution des interventions par niveau ..... 124

Figure 5.7 Distribution relative des interventions par niveau ..... 124

Figure 5.8 Distribution relative des sujets de deuxième niveau ..... 125

Figure 5.9 Distribution relative des interventions de niveau 2 ayant pour sujet les solutions alternatives et les critères ..... 126

Figure 5.10 Distribution du propos par sujets ..... 128

Figure 5.11 Graphe des fréquences observées en fonction des fréquences théoriques ... 140

Figure 5.12 Graphe des résiduels en fonction des fréquences théoriques ..... 141

Figure 5.13 Influence des critères sur les responsabilités de projet ..... 142

Figure 5.14 Contribution relative aux discussions de forme et de critère ..... 144

Figure 5.15 Occurrence relative des interventions ..... 144

Figure 5.16 Exemples de séquences ..... 150

Figure 5.17 Représentation graphique des liens significatifs entre interventions ..... 152

Figure 5.18 Séquences avec les interventions jointes ..... 153

Figure 5.19 Liens significatifs entre les échanges ..... 154

Figure 5.20 Structure des séquences ..... 156

Figure 5.21 Liens significatifs après regroupement des niveaux ..... 157

Figure 5.22 Structure des séquences après regroupement des échanges JUS ..... 158

Figure 5.23 Liens significatifs après regroupement des échanges JUS ..... 159

Figure 5.24 Regroupement hiérarchique des échanges ..... 161

Figure 5.25 Réécriture des échanges ..... 162

Figure 5.26 Liens significatifs entre échanges ..... 163

Figure 5.27 Transformations par réécriture des séquences ..... 164

Figure 6.1 Modélisation de la réunion de révision technique ..... 175

# LISTE DES ABBRÉVIATIONS

| ABBRÉVIATION | DÉFINITION |
|---|---|
| ACC | Acceptation |
| CMM | Capability Maturity Model |
| CRIT.C | Critère de contenu |
| CRIT.F | Critère de forme |
| DCG | Definite Clause Grammar |
| DEM | Demande |
| DEV | Développement |
| EDA | Exploratory Data Analysis |
| ESDA | Exploratory Sequential Data Analysis |
| EVA | Évaluation |
| Freq | Fréquence |
| GES | Gestion |
| HYP | Hypothèse |
| ID | Identificateur |
| INFO | Information |
| Interv | Intervention |
| INTRO | Introduction |
| ISO | International Standards Organization |
| JUS | Justification |
| LSA | Lag Sequential Analysis |
| MLL | Modélisation log-linéaire |
| OO | Orienté-objet |
| PRES | Présentation |
| REJ | Rejet |
| SOL | Solution |
| TA | Temps d'analyse |
| TS | Temps des séquences observées |
| UML | Unified Modeling Language |

# LISTE DES ÉCHANTILLONS

Échantillon 1 Exemple d'utilisation du schéma de codage ..... 108

Échantillon 2 Identification des séquences ..... 112

# AVANT-PROPOS

*"Le besoin de tenter de mesurer le non-mesurable afin d'améliorer notre compréhension d'un phénomène est aussi urgent en génie logiciel que dans d'autres disciplines."*

Norm E. Fenton, auteur du livre Software Metrics, a rigourous approach

# INTRODUCTION

L'objectif du génie logiciel est de définir des pratiques et des méthodes dans le but de produire du logiciel de qualité, et ce dans les délais et les budgets prévus. Cet objectif requiert donc des ingénieurs logiciels non seulement des connaissances techniques exhaustives mais aussi des connaissances dans l'art de diriger et coordonner les ressources humaines et matérielles d'un projet.

Le développement de logiciels complexes est nécessairement un processus coopératif. Il existe cependant plusieurs façons d'organiser le travail des individus pour parvenir aux résultats attendus. La formation d'équipes ou de groupes de travail est la solution employée par les entreprises modernes pour regrouper les individus ayant les connaissances et l'expertise nécessaires pour entreprendre un projet particulier. Une équipe doit parvenir à partager ses connaissances du domaine d'application, de sa solution et du design résultant.

Le génie logiciel est une discipline de génie unique en son genre parce que le produit, le logiciel, est une entité abstraite qui consiste en un ensemble d'instructions utilisables par un automate et qui n'existe qu'avec le support de média. Le processus de développement demande donc au concepteur une imagination lui permettant de voir les structures ou les caractéristiques d'un produit abstrait. Contrairement à d'autres disciplines du génie qui

permettent au concepteur de se baser sur des techniques éprouvées et des données empiriques lui permettant de créer un design en utilisant en grande partie son expérience, chaque logiciel est unique car c'est le logiciel qui, par sa grande flexibilité, doit se plier aux exigences de son environnement.

Cette notion place donc une vaste portion du risque non pas sur les caractéristiques techniques du produit mais sur les aptitudes des concepteurs à pouvoir comprendre et solutionner les problèmes techniques. L'ingénieur logiciel responsable du projet doit être capable de comprendre les principes fondamentaux de sa discipline et il doit aussi devenir une personne capable de comprendre le travail d'équipe et appliquer des principes de gestion du personnel et de leadership afin que le projet soit un succès.

Un processus de développement identifie les pratiques individuelles et coopératives devant être utilisées par les participants à un projet de développement. Il existe deux grands types de pratiques coopératives utilisées dans le processus de génie logiciel: les réunions de travail et les révisions techniques. Les réunions de travail permettent aux spécialistes d'établir les spécifications du produit et d'échanger les mérites de solutions envisagées de façon individuelle dans le but de déterminer la meilleure marche à suivre ou la meilleure solution à un problème. Les révisions techniques sont des réunions qui apparaissent tout au long du processus de développement est qui servent à s'assurer que les spécifications du produit répondent aux besoins. Les participants aux révisions techniques parcourent un artefact afin de s'assurer qu'il n'existe aucune ambiguïté quant

à son contenu.

Une limite de ces processus et pratiques utilisés en génie logiciel est qu'ils rendent peu compte de l'activité de développement de logiciel dans ses aspects individuels et coopératifs. On remarque donc souvent l'occurrence de comportements individuels et coopératifs non conformes à la pratique prescrite. D'un point de vue cognitif et coopératif, les déviations témoignent d'une certaine non compatibilité entre les modèles du génie logiciel et les modèles cognitifs de l'activité. Il semble donc important d'étudier ces activités d'un nouveau point de vue qui permette de guider le développement et l'évolution de certaines pratiques du génie logiciel.

L'expression "réunion de révision technique" désigne une rencontre (inspection, walkthrough, etc.) où plusieurs spécialistes vérifient un document (analyse, design, code, etc.) afin de l'accepter. Il existe un grand nombre de méthodes de révision qui sont toutes plus ou moins basées sur l'expérience spécifique de leurs auteurs.

On s'accorde pour dire que la réunion de révision technique, sous toutes ses formes, est utile pour le génie logiciel. Cependant, on s'accorde aussi pour dire qu'on ne sait pas exactement pourquoi est elle est si utile au domaine du logiciel. On utilise souvent le taux de détection d'anomalies comme mesure d'efficacité d'une méthode de révision, mais est-ce vraiment juste ? Par sa position privilégiée dans le processus de développement, la réunion de révision technique attire tous les intervenants majeurs à un projet de développement. Elle peut donc faire partie d'un vaste mécanisme de synergie pour

l'équipe qui l'utilise afin de disséminer la connaissance générale et plus spécifique. La détection d'anomalies n'est peut-être pas la seule activité présente lors de ce type de réunion.

Les activités coopératives sont de plusieurs natures. On les regroupe cependant souvent pour former des comportements collectifs tels que la prise de décision et la solution de problèmes, deux comportements clés en génie logiciel. Le rôle et les responsabilités des individus peuvent avoir un impact sur les discussions lors de la réunion. Les activités coopératives telles que la solution de problème et la prise de décision sont présentes lors de ces réunions.

Les processus de développement prescrivent une structure de travail coopératif à partir de laquelle des pratiques telles que la révision technique sont élaborées. Il existe un très grand nombre de méthodes de révision technique en génie logiciel, la plupart ayant été élaborées à partir de l'expérience spécifique d'un individu ou d'un groupe d'individus. Ces méthodes ont toutes un but en commun, l'amélioration de la qualité d'un logiciel. Elles utilisent cependant des moyens plus ou moins différents d'y parvenir.

Les définitions des méthodes de révision technique existantes ne les décrivent jamais en fonction du travail individuel et coopératif. Cela provoque des difficultés à expliquer les comportements déviants et problématiques. De plus, on comprend mal les effets d'une décision de modifier une méthode existante. Le résultat de tout cela est l'existence d'un grand nombre de méthodes de révision qui sont spécifiques à leur environnement

particulier.

Cette situation témoigne d'une certaine non-compatibilité entre les modèles du génie logiciel et les modèles cognitifs de l'activité. Il semble donc important d'étudier ces activités d'un nouveau point de vue qui permette de guider le développement et l'évolution de certaines pratiques du génie logiciel.

Ce travail propose une approche structurée permettant la mesure et l'analyse des réunions de révision technique sous un nouvel angle: la nature des activités effectuées par les individus. Le développement de logiciels requiert la coopération continuelle des individus. Cette coopération doit donc influencer toutes les pratiques du génie logiciel. Une réunion de révision technique peut donc être caractérisée par des activités individuelles et coopératives.

Ce travail de recherche est exploratoire, c'est-à-dire que l'objectif n'est pas d'élaborer une nouvelle méthode de révision pour ensuite mesurer les effets de sont utilisation par une équipe de développement mais plutôt d'observer une situation existante. La recherche s'effectue à partir d'une question de recherche et non pas d'une hypothèse de travail.

Ce travail propose une démarche basée sur les principes de l'*Exploratory Sequential Data Analysis* (ESDA). Cette démarche est utilisée lors de recherches où la séquence des données et plus importante que leur occurrence dans le temps. L'objectif global de la démarche est de modéliser les réunions de révision technique à partir d'observation et de

mesures faites sur un groupe d'ingénieurs logiciels travaillant sur un projet de développement logiciel industriel.

Une des contributions de cette recherche est l'élaboration d'une nouvelle forme de représentation des données. Cette représentation utilise une structure basée sur l'amalgamation de deux formalismes distincts: la linguistique et l'orienté-objet. La linguistique propose plusieurs façons de décrire les conversations tenues lors de réunions. L'unité de base est l'intervention d'un individu qui participe à un échange dans le cadre d'une séquence distincte. L'orienté-objet, par sa nature, est le formalisme utilisé pour faciliter la représentation des conversations. Les liens sémantiques entre les interventions des individus sont décrits à l'aide de classes, hiérarchies et associations. La syntaxe utilisée pour détailler cette représentation formelle permet de faciliter les analyses subséquentes et de ce fait, diminuer les coûts de ce genre de recherche. Cette syntaxe permet l'utilisation d'outils automatiques pour effectuer les mesures et analyses.

Les données utilisées dans le cadre de cette recherche proviennent de l'observation d'une équipe de quatre ingénieurs logiciels qui ont travaillé sur un projet de développement industriel d'une durée de 6 mois. La représentation formelle a été utilisée pour coder huit réunions de révision technique où les quatre équipiers étaient présents. Les données sont analysées de trois façons différentes afin d'augmenter notre compréhension de ce type de réunion:

1. Les verbalisations des individus sont regroupés afin de déterminer de façon

statistique l'effort déployé sur les comportements marquants;

2. Les interventions individuelles sont analysées dans un contexte coopératif afin d'identifier des activités coopératives; et

3. Les participants sont décrits en terme de rôles, responsabilités et intérêts afin de mesurer leur influence sur les activités identifiées et sur le déroulement global de la réunion.

Ces trois orientations d'analyse permettent de décrire une réunion non plus seulement en fonction du résultat à atteindre, mais plutôt en termes d'organisation du travail. Un modèle de la pratique est présenté et comparé à un modèle existant. L'ajout de paramètres sur les participants permet de bien cibler les participants potentiels à une réunion. Le résultat est une définition de la pratique de révision qui est beaucoup plus ergonomique et qui par le fait même, aura probablement tendance à être plus efficace pour le génie logiciel.

Ce document est divisé en deux parties distinctes: la description du contexte de travail et la recherche proprement dite. Le chapitre 1 présente en détails les différentes caractéristiques du travail coopératif, en passant par l'investigation des processus de prise de décisions et de solution de problèmes jusqu'aux différents rôles qui l'influencent.

Le chapitre 2 définit le contexte dans lequel la révision technique est effectuée. Par la suite, ce chapitre présente une revue de la littérature des principales méthodes de révision

technique actuellement utilisées en génie logiciel.

Le chapitre 3 présente la problématique à l'origine de ce travail de recherche et décrit deux facteurs qui témoigneront de son succès: une représentation des données formelles et des résultats utilisables par le domaine du génie logiciel.

Le chapitre 4 décrit la démarche globale utilisée dans le cadre de cette recherche. Les deux formalismes (linguistique et orienté-objet) sont par la suite décrits dans le contexte cette recherche. La représentation formelle élaborée pour représenter les interventions des individus et le regroupement de ces interventions en séquences est aussi exposée en détails.

Le chapitre 5 présente les analyses effectuées sur les données et les résultats empiriques obtenus. Ces résultats sont obtenus à l'aide d'analyses automatiques faites à l'aide de progiciels tels que *Statistica* et de logiciels développés dans le cadre de ce projet.

Le chapitre 6 présente la modèle conçu à partir de l'analyse des données. Plusieurs propositions d'amélioration des pratiques courantes y sont aussi proposées.

Les annexes présentent les données utilisées dans le cadre de cette recherche, le détail des analyses effectuées ainsi que les programmes conçus pour les effectuer. De plus, quelques articles publiés qui ont été générés par ce travail de recherche sont aussi inclus.

# CHAPITRE I

# LE TRAVAIL COOPÉRATIF EN GÉNIE LOGICIEL

L'ingénierie du logiciel est un travail d'équipe.

Qui dit travail d'équipe dit communication. Les caractéristiques du produit logiciel (abstraction, complexité) lui-même influencent grandement la nature de la communication parce que chaque individu approche le développement d'un logiciel avec son propre bagage de connaissances, d'expériences et surtout de compréhension.

Van Scoy (1992) stipule que la communication est l'élément essentiel de la gestion du processus. Cette communication est souvent difficile parce que les caractéristiques du logiciel et les problèmes associés avec son développement sont mal compris. La réussite d'un projet de développement de logiciel est donc probablement intimement liée à la façon de communiquer l'information et l'expertise nécessaire à l'accomplissement du travail à l'intérieur de l'équipe.

En fait, le travail coopératif est l'élément clé permettant d'harmoniser les connaissances dans un groupe et de s'assurer que les besoins technologiques sont biens compris. Les processus de génie logiciel définissent plusieurs pratiques de travail coopératif où l'information peut être disséminée.

L'objectif de ce chapitre est de survoler rapidement les théories et recherches qui ont été faites dans le domaine du travail coopératif. Ces recherches tendent à démontrer que les très grandes connaissances techniques des membres d'une équipe ne sont peut-être pas les seuls facteurs de réussites d'un projet. Il semble que le succès de plusieurs pratiques coopératives, telles les réunions de révision technique, soit influencé par la propension des individus à se compléter et par la démarche coopérative elle-même. Le domaine du génie logiciel tient rarement compte, pour l'instant, des recherches effectuées en psychologie. Ce chapitre discute des différentes caractéristiques du travail coopératif en général afin de mieux cerner leur application au domaine spécifique du génie logiciel et de la réunion de révision technique.

## 1.1 L'approche technique au travail d'équipe

### 1.1.1 Les principes du travail d'équipe

Les ingénieurs savent qu'il existe habituellement une suite logique d'actions qui doivent être effectuées lorsqu'un processus est complexe et qu'une action dépend du résultat de l'action précédente. D'après Berger (1985), le travail d'équipe demande lui aussi plusieurs tâches qui doivent être accomplies dans chacune de ses phases:

**Préparation au travail.** La préparation au travail est une étape où les membres du groupe tentent de se trouver une place à l'intérieur de l'équipe. Le chef d'équipe doit donc s'assurer que tous les membres ont la chance de se connaître, de démontrer leurs

habiletés distinctives, de clarifier leurs attentes et les règles qui régiront l'équipe. Ils doivent aussi se mettre d'accord sur la mission et les objectifs de l'équipe.

**Défi à l'autorité.** Après que les équipiers aient compris leur rôle et la mission de l'équipe, plusieurs coalitions se formeront à partir de personnes ayant le même point de vue et une lutte s'ensuivra afin de déterminer qui contrôlera l'équipe et comment elle le sera. Le point de mire de ce défi à l'autorité sera en premier lieu le chef d'équipe mais cela ne tardera pas à s'étendre aux autres leader potentiels. Les tâches de leadership nécessaires sont: une réponse équitable et calme aux défis des membres, la renégociation des attentes, la médiation entre les coalitions et la répartition du travail selon la préférence des membres et les ressources disponibles.

**Production.** Lorsque les membres de l'équipe se sentent assez confortables et qu'ils croient que tous les mécanismes sont en place pour partager le travail et résoudre les dissentions de façon équitable, l'équipe entre dans la phase de production où la qualité et la quantité du travail augmente radicalement. Le travail ne fait plus l'objet de distractions causées par des discussions sur le processus qui gaspillent temps et énergies. Le chef d'équipe doit continuer de contrôler et supporter les attentes et les standards de l'équipe, explorer de nouvelles façons d'appuyer la performance des individus et de faciliter la résolution de problèmes.

**Rapprochement.** Les équipes productives ayant du succès ont souvent éprouvé des sentiments de joie et de camaraderie. La cohésion est issue d'une intégration délicate de

la recherche individuelle de l'autonomie et de la participation réussie dans un groupe. Le chef d'équipe doit donc fournir des chances de célébrer les bons coups, mettre en évidence les contributions des différents membres de l'équipe, partager la reconnaissance et les récompenses dues à l'équipe et démontrer sa satisfaction pour un travail bien fait.

### 1.1.2 Les équipes à succès

Voici donc une liste des caractéristiques essentielles d'une équipe ayant du succès:

**Leadership consistant et approprié.** Un leader doit s'ajuster à la situation. Un style participatif n'est peut-être pas approprié dans tous les cas. Un bon leader doit être capable d'exercer un leadership qui permettra à son équipe d'atteindre son but.

**Objectifs clairs et concis.** Il est important d'obtenir une vision claire, bien comprise et acceptée des objectifs. Une équipe qui a du succès doit permettre à ses membres de discuter, clarifier et négocier les objectifs de l'équipe. De cette façon, chaque membre connait les priorités et comment y arriver.

**Intégration des buts personnels et collectifs.** Les membres d'un équipe ne s'engagent pas envers les objectifs d'équipe automatiquement. Chaque membre doit parvenir à des objectifs personnels à l'intérieur de l'équipe. Leur motivation sera donc proportionnelle à leur possibilité de se développer et de grandir tout en travaillant.

**Règles équitables et consistantes.** Les règles qui gouvernent les interactions doivent

être équitables, appliquées constamment et sujets au changement si nécessaire. Les équipes à succès s'adaptent aux conditions en ajoutant de nouvelles règles et en enlevant celles qui sont désuètes.

**Équipiers qualifiés et diversifiés.** Les équipes performantes semblent avoir chacun de ses postes remplis. Le mélange d'habiletés et d'idées permet à l'équipe d'utiliser les bonnes ressources pour un problème particulier. Malgré que ce mélange amène des conflits, une bonne équipe parvient à gérer ces différences et à en tirer des avantages.

**Climat approprié.** Le climat de travail est aussi important que le processus de travail de l'équipe. Une bonne équipe prend soin de ses membres en créant un climat de respect, d'excellence, d'innovation et de volonté d'aller toujours plus loin.

**Engagement.** Un caractéristique commune des équipes à succès semble être un engagement total. Les équipes travaillent fort en s'occupant des détails, en anticipant les problèmes et en s'organisant. Lorsque le temps de relaxer survient, les activités de loisirs consolident les liens entre les membres et réduisent le stress du travail.

Le processus de travail d'équipe ne peut être réduit complètement en principes et techniques sans y ajouter un autre concept, l'art du travail d'équipe.

### 1.1.3 L'art du travail d'équipe

**Attendre le bon moment.** Un chef d'équipe doit être capable de déterminer le moment

propice pour introduire une nouvelle règle, identifier un nouveau problème ou proposer une nouvelle solution. Une bonne compréhension du rythme naturel des événements afin d'attendre le dernier moment avant d'intervenir augmentera la probabilité que cette intervention soit profitable.

**Association avec les autres membres.** Les leaders ne doivent pas ressortir du groupe mais plutôt tenter de se confondre en supportant les autres dans l'accomplissement des objectifs de l'équipe.

**Entrevoir les possibilités.** Le chef d'équipe doit être capable d'évaluer de nouvelles façons d'atteindre ses objectifs. Il doit avoir une connaissance de la situation, une bonne appréciation des possibilités et la volonté de s'éloigner des conventions pour accomplir le travail.

## 1.2 La prise de décision

La réunion de révision technique a pour but, qu'il soit explicite ou implicite, de prendre une décision afin de permettre au projet de continuer. Cette décision peut être de nature technique, de gestion ou de tout autre type mais elle affecte le déroulement du projet. Il est donc important, dans un premier lieu, de comprendre le processus qui régit la prise de décision en groupe. Il faut cependant stipuler qu'il n'existe aucune "formule" qui permette d'assurer que la décision prise par un groupe sera la bonne mais une bonne connaissance de ce processus ne peut qu'aider à sensibiliser les participants.

### 1.2.1 Termes importants

Cette section fera le survol des termes importants utilisés dans le processus de prise de décisions en groupe. Tous ces termes sont familiers et peuvent être immédiatement définis de façon globale, mais il est nécessaire de comprendre précisément ce qu'ils veulent dire.

La relation entre la *prise de décision* et la *solution à un problème* n'est pas universellement reconnue. Plusieurs voient ces deux expressions comme des synonymes tandis que d'autres y voient deux significations tout à fait différentes. Il existe deux types de problèmes: ceux pour lesquels il existe une solution correcte habituellement déterminée par des moyens externes ou objectifs et ceux dont il n'existe aucune solution pouvant être validée de façon externe. Ces solutions ne sont découvertes que par l'acceptation du groupe et l'engagement des individus à implanter cette solution.

Une *décision* est l'ultime résultat de l'interaction de groupe. Elle est inévitablement un choix, fait par les membres du groupe parmi un nombre d'alternatives disponibles. Malgré que la quantité d'alternatives ne soit pas une mesure de la productivité d'un groupe, il ne faut pas oublier que seulement les alternatives proposées durant l'interaction du groupe seront disponibles lors de la prise de décision finale. D'ailleurs, le processus de décision n'est qu'une suite d'essais, de la part de chacun des membres, à influencer directement ou indirectement les autres membres à accepter ou rejeter une

alternative proposée.

Les membres d'un groupe parviennent à une décision lorsqu'il y a *consensus*. Une décision prise par consensus demande donc l'accord de tous les membres, mais cela n'est pas une condition suffisante au consensus. En effet, le consensus implique l'engagement envers la décision prise. L'élément essentiel du consensus est donc l'importance de la loyauté au groupe partagée par ses membres.

### 1.2.2 Approches à la prise de décision

Il existe deux approches distinctes mais très utilisées pour l'explication du processus de prise de décision en groupe. Chacune ayant ses limitations et ses avantages. L'approche *prescrite* tente d'expliquer comment les groupes devraient prendre les décisions tandis l'approche *descriptive* documente la façon dont les groupes prennent leurs décisions.

Les approches prescrites sont basées sur quelques hypothèses. Premièrement, les méthodes prescrites assument que tous les membres du groupe sont constamment rationnels. Ces méthodes fournissent habituellement une liste des étapes que le groupe doit franchir pour parvenir à une décision. De plus, elles préviennent contre les émotions et les aspects non-rationnels de l'interaction du groupe qui peuvent déranger l'efficacité du processus de décision. Une seconde hypothèse est que l'utilisation de ces méthodes mènera invariablement vers un groupe plus productif et que cela améliorera la qualité des résultats du groupe de décision.

Le plus connu des modèles prescrits est sûrement la "pensée réflective" suggérée par Dewey (1910). Malgré que Dewey ait bâti son modèle en fonction du processus mental d'un individu seulement, il a été beaucoup utilisé comme guide à la décision de groupe. Le modèle de Dewey inclut les six étapes suivantes:

1. Une difficulté est ressentie ou exprimée.
2. La nature du problème est définie.
3. La nature du problème est analysée.
4. Les solutions possibles sont suggérées pour résoudre le problème.
5. Les solutions sont comparées en les évaluant par rapport à des critères pré-sélectionnés et la meilleure solution est choisie.
6. La meilleure solution est implantée.

Ce modèle sert donc d'agenda universel qui guide un groupe vers un consensus par une progression étapes par étapes. Le groupe doit discuter et compléter chaque étape avant de passer à la suivante. Lorsque ce modèle est utilisé pour guider la prise de décision en groupe, il ne permet pas l'aspect socio-émotionnel de s'ingérer dans le processus.

Les approches descriptives sont basées sur le fait qu'il existe déjà un processus "naturel" de décision dans un groupe. Ce processus est présent en fonction de la liberté que le groupe possède à développer ses propres tâches et ses propres dimensions sociales sans l'influence d'aucune source externe.

Le modèle le plus familier de l'approche descriptive est le modèle en trois phases avancé

par Bales (1950). Après l'observation de plusieurs groupes de décision, il a été signalé que les membres de ces groupes tendaient à discuter différentes sortes de problèmes ayant trait à leurs tâches à différentes périodes de l'interaction du groupe. A partir des données compilées grâce au système d'analyse du processus d'interaction (IPA) de Bales, trois phases ont été identifiées:

1. Emphase sur les problèmes d'orientation ( décision sur l'évaluation de la situation actuelle).
2. Emphase sur les problèmes d'évaluation ( décision sur l'attitude à prendre vis-à-vis cette situation).
3. Emphase sur les problèmes de contrôle ( décision sur les actions à prendre).

Le IPA classe chaque communication durant l'interaction du groupe selon douze catégories différentes qui sont en fait six paires de catégories bipolaires (Tableau 1.1).

**Tableau 1.1 Le modèle de Bales**

| CATÉGORIE | CATÉGORIE | PROBLÈME |
|---|---|---|
| Est solidaire | Est antagoniste | Problème d'intégration |
| Montre une baisse de tension | Montre un hausse de tension | Problème de tension |
| Est en accord | Est en désaccord | Problème de décision |
| Donne des suggestions | Demande des suggestions | Problème de contrôle |
| Donne son opinion | Demande un opinion | Problème d'évaluation |
| Donne de l'orientation | Demande de l'orientation | Problème d'orientation |

Chaque paire de catégories est associée à un problème particulier auquel le groupe est confronté. L'emphase est portée sur les problèmes d'orientation, d'évaluation et de contrôle par l'utilisation de trois phases successives durant le processus. Il semble que ces trois phases soient cycliques puisque lorsqu'un groupe complète un tâche de prise de décision en utilisant les trois phases, le groupe retourne à la phase d'orientation au début de la tâche suivante.

Afin d'être consistant avec la dimension socio-émotionnelle présente durant le processus, il est requis que la structure sociale soit stable pour obtenir de la productivité. Durant les phases médianes, le modèle reflète une période de conflit entre les membres et de différences d'opinion sur les tâches et les normes sociales.

### 1.2.3 Le modèle en spirale

Bien que les deux approches décrites dans la section précédente soient différentes, elles ont en commun le fait qu'elles soient linéaires. Chacun des modèles est basé sur une progression étapes par étapes vers le résultat escompté. Scheidel et Crowell (1976) ont découvert que le processus d'interaction dans la prise de décision n'était pas linéaire. Le modèle en spirale a donc été développé pour représenter ce phénomène.

Lorsqu'un membre du groupe introduit une nouvelle idée, les autres membres sont en accord ou en désaccord ou proposent des modifications. Cette nouvelle idée est l'objet de la discussion et elle est développée pour refléter le point de vue du groupe jusqu'au point où tous les membres du groupe sont en accord. Le groupe ancre donc sa position et introduit une nouvelle idée préliminaire à partir de l'idée précédente. Ce processus en spirale se sert donc de ces points d'ancrages comme base pour la prochaine discussion sur l'idée suivante. Si un consensus est établi pour cette nouvelle idée, un nouveau point d'ancrage est établi. Si l'idée est rejetée, alors le groupe retourne au dernier point d'ancrage pour repartir avec une nouvelle idée.

Le processus en spirale est cumulatif et progressif. Il permet de refléter les modifications continuelles des idées et le retour vers les consensus établis pour reconfirmer la position du groupe. De plus, ce modèle prend en compte l'inefficacité apparente du groupe de décision par rapport à l'utilisation du temps. De plus, il est raisonnable de conclure qu'un groupe ne prend pas des décisions mais plutôt que les décision émergent de

l'interaction du groupe.

En observant les groupes de prise de décisions, B. Aubrey Fisher (1974) a découvert quatre phases au processus d'émergence des décisions (Tableau 1.2). Contrairement à d'autres travaux d'observations, chacune des alternatives de décisions ont été identifiées et le processus d'interactions durant la conversion de ces idées préliminaires en un consensus a été observé. Les données ainsi recueillies ont permis d'établir quatre phases distinctes dans la décision de groupe, chacune d'elles caractérisées par un type d'interaction très différent.

**Tableau 1.2 Processus d'émergence de décisions**

| Phase | Définition |
|---|---|
| Orientation | Période servant à former des opinions et à se débarrasser des inhibiteurs sociaux. Elle sert à se connaître et à tenter d'exprimer des attitudes envers les autres. |
| Conflit | Période caractérisée par la dispute et les conflits idéologiques vis-à-vis les propositions alternatives. |
| Émergence | Il s'agit probablement l'étape cruciale du processus de décision de groupe. Durant cette phase le résultat possible de l'interaction de groupe devient de plus en plus visible. |
| Renforcement | Période servant à développer l'engagement des membres envers les décisions qui étaient sujets à conflit durant la seconde phase et qui ont émergé durant la troisième phase. |

### 1.2.4 Sommaire de la prise de décision

Le processus de prise de décision englobe la recherche d'une solution à un problème, ce qui demande un taux élevé d'acceptation de la solution. Une décision constitue un choix entre différentes solutions dont la somme peut être considérée comme la performance du groupe.

Le processus de prise de décisions en groupe est comparé au processus de leadership et de l'émergence des rôles afin de démontrer que les décisions ne sont pas prises par un groupe mais plutôt qu'elles émergent de l'interaction du groupe.

Un groupe parvient à un consensus sur une décision à prendre en modifiant une suite de propositions dans un processus où les alternatives sont introduites, discutées, écartées et réintroduites avec des petites modifications jusqu'au moment où un consensus est établi.

## 1.3 La solution de problèmes

Bien que le processus de prise de décisions en groupe soit connu et que cette notion soit claire dans l'esprit de tous les membres du groupe, il se peut que l'on se rende compte qu'il manque toujours quelque chose pour assurer le succès de la réunion. Il est donc important que la personne agissant comme facilitateur soit au courant de la dynamique présente lors de la solution de problèmes en groupe et qu'elle ait une bonne connaissance des nombreux outils de prise de décisions disponibles.

La première étape dans la solution de problèmes en groupe est de s'assurer que le problème est perçu de la même façon par tous les membres (Doyle et Strauss, 1976). Le meilleur moyen d'établir une définition commune est de demander à chacun des participants son point de vue quant à la nature du problème. L'étape suivante est logiquement d'obtenir une définition précise du problème dans des termes pouvant être utilisés lors des étapes futures. De plus, la définition du problème limite le nombre d'alternatives possibles, appelées l'étendue du problème, qui est elle-même proportionnellement relative au nombre de solutions possibles.

Durant la phase d'analyse, l'objectif est de partager le problème en parties et d'examiner comment elles interagissent entre elles. Tous les problèmes peuvent être subdivisés récursivement en plus petits sous-problèmes jusqu'au point où ils sont simples à comprendre. Il est aussi important de développer une liste de solutions existantes à un problème et d'en établir une liste des avantages et des désavantages.

Il est important de constater qu'il existe plusieurs méthodes qui peuvent être utilisées dans différentes phases du processus. Par exemple, le *brainstorming* peut servir à générer des problèmes, des définitions, des critères et même des alternatives. Le brainstorming est l'une des méthodes les plus simples et les plus efficaces pour générer des réponses aux questions du groupe. Les règles sont simples: tous et chacun lancent les idées qui leur passent par la tête afin qu'elles soient prises en note et aucun autre membre n'a le droit de critiquer l'idée avant que la session de brainstorming ne soit

terminée.

Une autre méthode plus systématique d'analyser un problème et de générer des alternatives est appelée la méthode de l'échiquier ou l'analyse morphologique. Cette technique oblige les participants à examiner les différentes interactions et les combinaisons de deux ensembles de variables. Les carrés formés à partir des intersections de deux variables sont considérés comme des alternatives distinctes. Cette technique permet de trouver un plus grand nombre de possibilités et d'examiner le problème sous plusieurs angles.

La clé pour évaluer les alternatives en groupe est d'établir les critères de sélection avant de juger celles-ci. La présélection des critères permet d'éliminer la sélection d'alternatives à partir de critères personnels conscients ou subconscients. Cette étape importante est souvent omise lors du processus de solution de problèmes ou de prise de décisions.

Lors de la phase de prise de décision, le groupe devrait revoir les résultats de l'évaluation et s'engager fermement envers un ou plusieurs champs d'actions. Les méthodes d'interaction de réunions de groupes sont établies en fonction de produire une situation où tous les partis y trouvent leur compte.

La rétroaction est le moyen utilisé pour examiner les résultats du processus de solutions de problèmes. L'information ainsi reçue permet de valider le processus et d'y apporter

les correctifs nécessaires.

## 1.4 Des équipes de travail efficaces

Il est indiscutable qu'une meilleure connaissance de la façon dont une équipe est formée, structurée et menée ne peut qu'améliorer la productivité et que ceci ne peut être atteint sans une évaluation plus rigoureuse des individus formant ces équipes. Il est aussi important de mentionner que les professionnels de l'informatique forment un bassin de personnes significativement différent des autres groupes habituellement présents à l'intérieur d'une organisation.

### 1.4.1 Le modèle de diagnostique du travail

Ce modèle est basé sur les modèles de motivation contemporaine de Emery et Emery (1975), Hackman et Oldham (1980) et autres. Ce modèle propose que la motivation est intrinsèque et qu'un déterminant majeur de la motivation est la nature et la structure du travail d'une personne. Chaque travail peut être évalué en fonction de cinq dimensions:

1. Le nombre d'habiletés et de talents demandés par le travail d'un individu.
2. La mesure à laquelle l'emploi nécessite l'accomplissement de tâches ayant un résultat visible.
3. La mesure de l'impact du travail accompli sur la vie d'autres individus.
4. Le degré de liberté et d'indépendance de l'individu à établir lui-même ses échéances et ses méthodes de travail.

5. La mesure jusqu'à quel point les activités de son emploi lui permette d'avoir une idée claire de son efficacité et de sa performance.

Un questionnaire permet aux individus d'évaluer leur travail en fonction des cinq dimensions selon une échelle de 1 à 7. Le total est appelé le potentiel de motivation (MPS). Il semble que le MPS d'un travail soit directement relié à la motivation, la productivité, la qualité du travail et autres. En améliorant le MPS des tâches de développement, les équipes pourront ainsi améliorer leur motivation et leur moral.

### 1.4.2 L'indicateur de type Myers-Briggs

En approfondissant le travail de Carl Jung (1921), Briggs et Myers (1980) stipulent que pareillement au fait qu'une personne soit née avec la prédisposition d'être droitier ou gaucher, elle est aussi prédisposée à être extravertie ou introvertie (E ou I), intuitive ou sensée (N ou S), réfléchie ou sensible (T ou F) et juge ou perceptive (J ou P).

Les extravertis sont orientés vers le monde extérieur tandis que les introvertis tendent à se concentrer sur les idées et les sentiments. Les personnes sensées sont concernées par les détails et les faits tandis que les intuitifs préfèrent les possibilités, le futur et l'ensemble. Le type réfléchi prend des décisions basées sur la logique et les critères objectifs pendant que le type sensible prend des décisions à partir de données plus subjectives comme les sentiments et les émotions. Les perceptifs sont flexibles, ouverts et toujours à la recherche de nouvelles informations tandis que les juges veulent garder les choses sous

leur contrôle. Les théoriciens ont développé une matrice contenant seize personnalités reliées aux quatre dimensions (Figure 1). Ce modèle a été utilisé par plusieurs chercheurs (Lyons, 1985; Bostrom et Kaiser, 1981; Bush et Schkade, 1985) dans leurs recherches sur les spécialistes en informatique.

| | | SENSÉE (S) | | INTUITIVE (N) | |
|---|---|---|---|---|---|
| | | RÉFLÉCHIE (T) | SENSIBLE (F) | RÉFLÉCHIE (T) | SENSIBLE (F) |
| INTROVERTIS (I) | JUGE (J) | **ISTJ** Sérieux, silencieux, il a du succès grâce à la concentration et la minutie. Il est pratique, ordonné, terre à terre, logique, réaliste et sûr. Il prend des responsabilités. | **ISFJ** Silencieux, amical, responsable et consciencieux. Il travaille sans relâche pour remplir les obligations. Il est minutieux, assidu et précis. Il est loyal et attentionné. | **INFJ** Il réussit grâce à sa persévérance, son originalité et son désir de faire tout ce qui est nécessaire. Il est une force silencieuse qui respecte et aide les autres. Il est respecté pour ses principes. | **INTJ** Il possède de l'originalité et une bonne attitude envers ses propres idées et choix. Il est sceptique, critique, indépendant, déterminé et souvent obstiné. |
| | PERCEPTIVE (P) | **ISTP** Spectateur froid, silencieux, reservé et analyste. Il s'intéresse aux principes, comment les choses fonctionnent. Il a des moments d'humour oiginal. | **ISFP** Retiré, amical, silencieux, sensible et modeste. Il évite les désaccords. Il est souvent relaxe quant aux choses à faire. | **INFP** Il aime apprendre, les idées, les langages et les projets personnels. Il tend à en faire trop mais il finit par le faire. Amical mais souvent préoccupé. | **INTP** Silencieux, reservé impersonnel. Il aime la théorie et les sujets scientifiques. Il s'intéresse aux idées et non aux conversations anodines. Il a des intérets très particuliers. |
| EXTROVERTIS (E) | PERCEPTIVE (P) | **ESTP** Terre à terre, il ne s'inquiète et ne se dépêche pas. Il prend ce qui vient. Il est un peu brusque et insensible. A son meilleur avec les choses qui s'assemblent et se désassemblent. | **ESFP** Amical et extériorisé. Il rend les choses plus intéressantes pour les autres. Il aime les sports et faire des choses. Il est meilleur à se souvenir des choses qu'à comprendre les théories. | **ENFP** Enthousiaste, ingénieux et imaginatif. Il est capable de faire tout ce qui l'intéresse. Il est rapide à trouver une solution et à aider les autres qui ont des problèmes. | **ENTP** Rapide, ingénieux et bon à faire des choses. Il peut argumenter une question pour le plaisir seulement.Très bon pour trouver une solution à un problème difficile, il peut négliger le travail de routine. |
| | JUGE (J) | **ESTJ** Pratique, réaliste, terre à terre et un naturel pour les affaires et la mécanique. Ne s'intéresse pas à des sujets s'il n'en voit pas l'utilité. Il aime organiser et contrôler une activité. | **ESFJ** Chaleureux, populaire consciencieux, coopératif et loquace. Il a besoin d'harmonie. Il est à son mieux quand il est encouragé et il a peu d'intéret pour l'abstrait et les sujets techniques. | **ENFJ** Responsable et émotif. Il est sensible à ce que les eautres pensent ou veulent. Il est sociable et populaire. Il est sensible à la critique et à l'encouragement. | **ENTJ** Franc, décisif, un leader. Il est bon avec tout ce qui demande du raisonnement et de l'intelligence. Il est souvent plus optimiste que ce que son expérience lui dicte. |

**Figure 1.1 Le modèle Myers-Briggs**

### 1.4.3 Le modèle des rôles de Belbin

Belbin (1981) a développé un modèle de rôles à l'intérieur de l'équipe selon lequel le

leadership n'émane pas d'une seule personne mais plutôt de chacun des membres de l'équipe à tour de rôle, tout dépendant de la situation ou le problème auquel ils font face. Les membres d'une équipe efficace contribuent en prenant un rôle de leadership ou d'équipe. De plus, il existe un lien important entre le type de personalité (Myers-Briggs) des membres de l'équipe et le rôle joué dans l'équipe. Belbin a identifié huit rôles d'équipe qui sont nécessaires afin d'obtenir une équipe efficace:

Président: Il contrôle la marche de l'équipe vers les objectifs de groupe en utilisant de la meilleure façon possible les ressources disponibles. Il reconnait où se situent les forces et les faiblesses et s'assure que le potentiel de chacun des membres est utilisé pleinement.

Façonneur: Il modèle la façon dont l'effort d'équipe est donné. Il dirige les discussions et cherche à imposer sa façon de faire sur les résultats des activités de groupe.

Implanteur: Il amène de nouvelles idées et stratégies en portant attention aux problèmes importants et en se mettant à la recherche de nouvelles approches à un problème auquel le groupe est confronté.

Moniteur-évaluateur: Il analyse les problèmes, évalue les idées et les suggestions de façon pratique pour permettre à l'équipe de prendre une décision plus équitable.

Travailleur: Il prend les concepts et les plans pour en faire des procédures pratiques. Il met à exécution les plans qui ont été approuvés de façon systématique et efficace.

Équipier: Il supporte les membres en position de force en ajoutant à leurs suggestions. Il aide ceux qui ont des problèmes. Il améliore la communication entre les membres et rehausse l'esprit d'équipe.

Investigateur: Il explore et rapporte sur les idées, développements et ressources disponibles à l'extérieur du groupe. Il crée les contacts externes pouvant être utiles à l'équipe et à des négociations ultérieures.

Minutieux: Il s'assure que l'équipe est protégée contre les erreurs possibles en recherchant activement les aspects du travail qui demandent une attention minutieuse. Il maintient un degré d'urgence dans l'équipe.

Afin de rendre ce modèle plus spécifique au domaine de l'informatique, Thomsett (1990) a ajouté un rôle de groupe:

Façonneur technique: Il donne une direction technique au travail soit en analyse, design et programmation. Cette direction est seulement basée sur son expertise technique.

En combinant les trois modèles décrits auparavant, il est possible de créer un outil de diagnostique permettant d'évaluer le rendement des équipes. De plus, ces modèles permettent de restructurer et de former des équipes balancées et efficaces à l'aide de lignes directrices qu'ils rendent disponibles.

Selon le modèle de Belbin (1981), les équipes de développement logiciel types sont composées d'un programmeur type et d'un chef d'équipe qui partagent le même rôle parce que la nature de la formation en informatique tend à filtrer les autres types de personnalités. Il semble donc qu'un grand nombre d'équipes de développement informatique manque les habiletés nécessaires à la gestion du personnel et à la négociation si elles sont comparées à d'autres types d'équipes.

Il est donc important de reconnaître la nature spéciale des professionnels de l'informatique dans le développement de pratiques telles que les réunions de révision technique.

## 1.5 Les réunions en génie logiciel

Le domaine du génie logiciel est de plus en plus attentif aux besoins des individus pratiquant la discipline afin de leur fournir les meilleurs outils disponibles. C'est dans cette optique que l'étude de la coopération entre les individus devient de plus en plus importante. Pourtant il n'existe que très peu de littérature qui traite de ce sujet en génie logiciel.

La plupart des logiciels qui sont produits en équipe le sont à cause de leur complexité, le besoin d'une expertise très spécialisée ou d'un échéancier très serré. Le développement de logiciels est un exemple classique d'un problème mal structuré (Reitman, 1965) et est donc considéré comme une tâche difficile, même pour une seule personne. La coordination de ces activités à l'intérieur d'un groupe est donc passablement difficile et les réunions sont donc très importantes pour coordonner les activités des membres de l'équipe.

Olson (1992) a observé des groupes d'ingénieurs logiciels discutant les premières ébauches de systèmes qui seraient produits plus tard, habituellement par d'autres. Ces groupes avaient reçu une description générale du système en question et leurs discussions furent un mélange de spécifications des besoins et de design de haut niveau. En termes de processus logiciel, ces réunions faisaient parties des étapes d'analyses et de conception. Toutes ces réunions comprenaient un groupe de personnes assis autour d'une table avec pour seuls outils du papier, un crayon et un tableau.

L'analyse des discussions a permis de développer un ensemble de catégories d'activités qui reflètent la nature des discussions:

Sujet: Les questions et aspects majeurs concernant l'objectif du projet. L'emphase est mis sur des sujets tels que "Devrait-on offrir cette fonctionnalité à l'utilisateur ?" ou aussi "Comment peut-on implanter cette fonction ?". Cela inclut aussi l'élaboration d'une

idée ou d'une description qui n'est pas une réponse à une question d'un membre du groupe.

Parfois, les sujets ne sont pas indiqués explicitement mais plutôt inferrés par la présentation de deux alternatives. Ceux-ci marqueront typiquement la première partie d'une phrase qui présentent ces alternatives.

Alternatives: Solutions ou propositions concernant l'objectif du projet. Ce sont des caractéristiques pouvant être offertes à l'utilisateur ou des façons d'implanter des caractéristiques préalablement choisies. Cela inclut l'élaboration d'une idée dont la description n'est pas une réponse à une question d'un membre du groupe. Parfois, cette catégorie inclura l'élaboration des implications de l'idée présentée.

Critères: Les raisons, arguments ou opinions qui évaluent une solution alternative ou une proposition. Les critères apparaîtront parfois sous la forme de systèmes analogues en supposant que si une solution fonctionne avec l'autre système, elle devrait le faire avec le système actuel. Un critère peut être mentionné sans connexion avec aucune alternative afin que les membres du groupe soient rappelés d'évaluer les solutions futures en en tenant compte.

Gestion de projet: Déclarations concernant des activités indirectement reliées au contenu du design, par lesquelles les individus se font assigner certaines tâches, décident de la prochaine rencontre, font un rapport sur un activité (ne contenant aucun design) assignée à la réunion précédente, etc.

Gestion de la réunion: Déclarations à propos de la gestion de la réunion actuelle, telles que brainstorming, décisions, votes, arrêts des discussions, etc.

Sommaire: Revues de l'état du design ou de l'implantation, des sujets, alternatives et critères. Le sommaire doit être une déclaration simple (liste).

Clarification: Questions et réponses que quelqu'un demande ou semble ne pas comprendre. Cela inclut les répétitions nécessaires à la clarification, l'association et des explications. Les clarifications servent à éliminer les mauvaises interprétations des autres individus. Cette catégorie est divisée en dix catégories plus spécifiques.

Interruption: Arrêts des discussions à cause d'une plaisanterie ou une interruption causée par des éléments externes à la réunion. Le changement de cassettes de la caméra devrait être considéré

comme une interruption.

But: Déclaration de la raison et l'objectif de la réunion du groupe. Les contraintes de travail tels que l'échéancier et la motivation de l'importance de ces contraintes.

Walkthrough: Une collection du design fait jusqu'à présent ou les étapes nécessaires à l'utilisateur pour utiliser le design afin de le réviser ou clarifier la situation. Cela correspond habituellement à l'identification des besoins de l'usager, le flux des données ou messages dans l'architecture du système.

Autre: Temps utilisé dans la réunion qui ne peut être identifié à une catégorie en particulier.

Après examen des discussions, il a été démontré que les membres de ces réunions mentionnaient clairement les sujets, les alternatives et les critères d'évaluation de ces alternatives en plus de partager leur expérience. Ces catégories ne sont cependant pas complètes et il existe une multitude d'autres choses à expliquer à propos de telles réunions. De plus, cette étude n'a pas tenu compte de la participation des membres et des rôles joués par ceux-ci dans les discussions. Il est certain qu'une meilleure connaissance de ces deux dimensions ne pourrait qu'aider les personnes ayant pour tâche de faciliter ces réunions.

Herbsleb et al. (1995) ont utilisé une représentation similaire afin de comparer des réunions de design dans un environnement orienté-procédure à d'autres dans un environnement de développement orienté-objet. Leur objectif était de comparer les aspects coopératifs de ces deux paradigmes de développement afin de déterminer si l'orientation-objet était supérieure.

Très peu d'études ont été effectuées sur les réunions de révision technique. Celles qui l'ont fait ont identifié des activités d'évaluation, de compréhension et de négociation. Contrairement aux réunions de design, les réunions de révision ne semblent pas contenir de dialogues de design. Cette absence peut être expliquée par différentes raisons. Karsenty (1991) a utilisé une approche différente dans son étude de réunions de validation de bases de données entre des concepteurs et des utilisateurs. La représentation était basée sur quatre types de dialogues (évaluation, clarification, négociation et analyse de problèmes). La présence des utilisateurs a sûrement réduit la volonté des concepteurs d'effectuer du design. Letovsky et al. (1987) ont analysé les réunions d'inspection de code chez IBM. La nature de très bas niveau des artefacts de la phase d'implantation prévient sûrement aussi la présence de dialogues de design.

L'étude de Seaman et Basili (1998) a mesuré jusqu'à quel point la structure organisationnelle elle-même influençait sur la communication dans une réunion de révision technique. Ils affirment que la mesure et la compréhension des phénomènes organisationnels ont le potentiel d'influencer profondément le succès de futurs projets de

développement logiciel.

## 1.6 Sommaire

Il n'existe pas beaucoup de données empiriques qui permettent d'évaluer l'efficacité des pratiques utilisées lors du développement logiciel. On sait cependant que les réunions de révision technique occupent un place prépondérante dans le cheminement d'une équipe vers son but final, le développement d'un logiciel. Il est donc important de mieux comprendre les phénomènes coopératifs présents lors du développement de logiciels par une équipe de professionnels. Ce sommaire des principales recherches sur la solution de problèmes en groupes a permis de déterminer trois domaines distincts:

1. La formation des équipes de travail performantes;
2. Les techniques de solution de problèmes en groupe et la dissémination de la connaissance par les questions et les réponses des développeurs lors des réunions; et
3. L'organisation de l'équipe de développement, son leadership et le rôle joué par les membres de l'équipe.

On a pu voir que la solution de problèmes en groupe est un processus complexe en lui-même et que son efficacité dépend en grande partie de la connaissance de ses techniques par les membres du groupe de décision. Il est facile de comparer un groupe type de solution de problèmes et de prises de décision avec un groupe d'ingénieurs logiciels

tentant de valider une solution complexe lors d'une réunion de révision technique.

La communication est la pierre angulaire du travail en équipe. Un manque de communication ne peut que rendre le travail en groupe inefficace et peut-être inutile. Les utilisateurs du processus de génie logiciel ne peuvent que bénéficier d'une meilleure connaissance des méthodes de communication utilisées dans leur travail de tous les jours.

# CHAPITRE II

# LA RÉUNION DE RÉVISION TECHNIQUE

Avant d'aborder spécifiquement la description de la recherche, il est nécessaire de placer la réunion de révision technique dans son contexte. L'utilisation de mécanismes de révision en génie logiciel n'est pas nouvelle, il est donc nécessaire d'effectuer une liste des différentes méthodes existantes. Ce survol permettra de définir exactement ce qu'est une réunion de révision technique. Cette définition sera utilisée dans la suite de ce travail de recherche.

## 2.1 Définition du contexte

### 2.1.1 Le produit logiciel

Une définition intéressante du terme logiciel a été citée par Naur et al. (1969) dans leur compte rendu d'une réunion de l'OTAN, la première sur le génie logiciel. Ils décrivent le logiciel comme suit:

> *"Le produit logiciel peut être considéré comme un objet abstrait qui a évolué à partir d'un énoncé d'un besoin pour se finaliser par un logiciel, lequel comprend tout aussi bien le code, objet ou source, que les*

*diverses formes de documentation produites tout au long du processus de développement."*

Un produit logiciel est un ensemble de procédures ou de règles, de données pertinentes à ces procédures et la documentation nécessaire pour répondre à un besoin spécifique. Les caractéristiques du produit logiciel dont il est question dans ce travail sont les suivantes:

1. Il s'agit d'un logiciel de grande dimension:
2. Il est utilisable par les usagers "naïfs";
3. Il a un cycle de vie assez long ( de cinq à dix ans);
4. Une documentation est nécessaire par ses utilisateurs; et
5. Il peut être complété en plusieurs versions.

Cette définition du logiciel est celle qui sera utilisée tout au long de ce texte. Pour les besoins de cette recherche, les expressions "produit logiciel" et "logiciel" seront utilisées pour exprimer le même concept.

### 2.1.2 Le génie logiciel

Le génie logiciel est une discipline basée sur l'établissement et l'utilisation de principes fondamentaux dans le but de développer de façon économique des produits logiciels fiables qui fonctionneront sur de vraies machines (Naur et al., 1969). On peut décrire de façon plus générique le génie logiciel en terme d'objectifs à atteindre. Le génie logiciel doit faciliter l'accomplissement des tâches suivantes (Pressman, 1997):

1. Description du problème à résoudre;

2. Description des caractéristiques de l'artefact utilisé pour résoudre le problème;

3. Description du processus de développement de l'artefact; et

4. Description des approches utilisées pour identifier et corriger les erreurs trouvées lors du développement.

Boehm (1981) définit le génie logiciel à partir des définitions de base du dictionnaire:

> *"Logiciel: l'ensemble complet des programmes, des procédures et de la documentation connexe associée à un système, et particulièrement à un système informatique".*
>
> *"Génie: l'application de la science et des mathématiques grâce auxquelles les propriétés de la matière et des sources d'énergie de la nature sont rendues utiles à l'homme par des structures, des machines, des produits, des systèmes et des processus".*
>
> *"Génie logiciel: l'application de la science et des mathématiques grâce auxquelles la capacité de l'équipement informatique sont rendues utiles à l'homme via des programmes informatiques, des procédures et de la documentation associée".*

Le développement d'un logiciel se fait toujours en fonction de critères précis qui influencent grandement la conception d'un logiciel unique. Il est donc très important de bien définir à priori tous les éléments susceptibles d'influencer le produit logiciel afin de s'assurer qu'il répondra aux besoins. Le génie logiciel est une discipline qui amène une certaine rigueur dans le développement de logiciels en définissant un cycle de vie qui décrit précisément les phases par lesquelles un produit logiciel doit passer.

### 2.1.3 Cycle de vie du logiciel

La vie du logiciel commence par l'élaboration d'un certain besoin pouvant être rempli par un produit logiciel pour se terminer par l'installation de ce logiciel dans son environnement de travail et la maintenance de celui-ci. Les phases intermédiaires comprennent tout le travail de design, de réalisation et de tests nécessaires à l'implantation du logiciel. La Figure 2.1 présente la suite logique de ces phases telles que décrites par un cycle de vie en cascade type. Ce cycle de vie consiste en une suite de phases devant être satisfaites individuellement avant de pouvoir passer à la suivante.

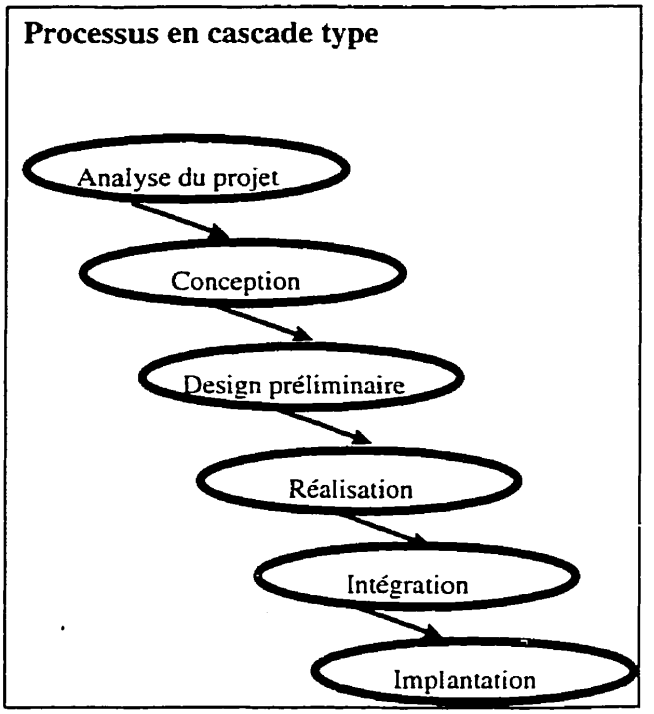



**Figure 2.1 Cycle de vie en cascade**

La vie du produit logiciel débute lorsqu'un besoin est identifié. La première phase du cycle de vie consiste à identifier tous ces besoins et ainsi définir tout le travail nécessaire pour produire le logiciel en question. Souvent, cette phase permettra aussi de définir tous les intrants nécessaires pour compléter le travail (budget, personnel, échéancier, etc.).

C'est durant la phase de conception que les besoins sont décrits en tant que fonctionnalités du produit. Comme son nom l'indique, on identifie les concepts opérationnels qui seront à la base des phases subséquentes. On doit aussi identifier précisément quelle technologie sera utilisée pour réaliser le logiciel. Cette phase résulte donc en un modèle simplifié de l'environnement contenant ses contraintes de

performance et d'implantations.

Le design préliminaire sert à détailler les différentes fonctionnalités identifiées dans la phase précédente. C'est lors de cette phase que les besoins (fonctionnalités) sont présentés sous formes techniques pour la première fois. On obtient donc une liste de spécifications techniques qui représentent les tâches discrètes du projet. Plus concrètement, l'architecture du système sera identifiée et détaillée.

La phase de réalisation sert à compléter ces tâches discrètes. Il s'agit ici de terminer le design technique de chacune de ces tâches pour permettre de les coder dans le langage de programmation choisi. On obtient donc un design des structures de données et des algorithmes à coder. À la fin de cette phase, les tâches auront été codées individuellement et seront prêtes à être intégrées ensemble pour former le logiciel proprement dit.

La phase d'intégration sert à relier toutes ces tâches individuelles afin de les faire fonctionner ensemble. Certains tests sont faits sur les tâches individuelles ainsi que sur l'ensemble du logiciel.

La phase d'implantation consiste à installer le logiciel complet dans son environnement opérationnel. De plus, la documentation produite tout au cours du projet est recueillie pour former un document informatif sur les différentes fonctionnalités du logiciel.

Le cycle de vie décrit précédemment n'est qu'un exemple concret des différentes phases

de développement qui peuvent être utilisées pour développer un logiciel. Une équipe de développement se base sur un cycle de vie en particulier pour identifier toutes les étapes nécessaires au développement d'un logiciel.

### 2.1.4 Processus de développement

Un processus de développement logiciel est le procédé par lequel une équipe de développement logiciel produira le logiciel final. Un processus est habituellement basé sur un cycle de vie du logiciel précis. Le processus décrit précisément tous les extrants de chacune des étapes, incluant les documents exigés et une définition des tests. De plus, il spécifie aussi très clairement toutes les pratiques de vérification et validation devant être accomplies durant chacune des étapes du développement. Ces pratiques incluent:

1. Gestion de projet et suivi;
2. Assurance qualité;
3. Gestion de la configuration;
4. Préparation et production de la documentation;
5. Gestion du risque;
6. Mesure; et
7. Réunions de révision technique.

Chacune de ces pratiques produit un certain nombre de documents qui permettent de suivre le cheminement de l'équipe de développement. De plus, ces documents permettent de connaître le rationnel des décisions faites durant le projet et ainsi de

comprendre les différentes raisons derrières le choix des différentes implantations des fonctionnalités du produit logiciel.

## 2.2 La Réunion de Révision Technique

L'une des pratiques clés du processus de développement est la réunion de révision technique qui consiste à vérifier l'état courant du projet de développement et de valider les spécifications des tâches subséquentes. Une réunion de révision technique requiert la présence de plusieurs réviseurs pour une certaine période et doit se dérouler selon des procédures établies (IEEE, 1993; Bell, 1987). Boehm (1987) inclut la réunion de révision technique dans les dix pratiques les plus importantes du génie logiciel parce qu'elle permet de détecter rapidement et économiquement 60% des anomalies créées au cours du développement.

Il existe présentement, en génie logiciel, plusieurs façons de décrire ce qu'est une réunion de révision technique. Inspection et walkthrough sont des termes plus spécifiques souvent utilisés pour identifier une réunion de révision technique.

### 2.2.1 L'inspection formelle et le *walkthrough*

Une inspection permet de détecter et résoudre des erreurs le plus tôt possible lors du processus de développement logiciel et assure que les erreurs ne sont pas propagées d'une étape de développement à l'autre (Fagan, 1976). Une inspection est une période de temps

où un groupe de personnes effectuent l'évaluation technique d'un artefact afin d'y découvrir des erreurs ou anomalies (Tjahjono, 1996).

Johnson (1996) définit l'inspection comme une fonction de transfert où un groupe d'experts techniques évaluent le bien-fondé d'un artefact (intrant) en utilisant une démarche spécifique. L'extrant est un nouvel artefact amélioré.

Un walkthrough est une revue informelle effectuée par les pairs qui est moins rigoureuse que l'inspection proprement dite (Brykczynski, 1994). Il s'agit d'une revue informelle parce qu'il n'est pas nécessaire de documenter les résultats de la réunion (Johnson, 1996).

Lorsqu'on tente de comparer les deux types de révision technique dans la littérature, on décrit l'inspection comme une rencontre très structurée impliquant la lecture et la vérification de l'artefact révisé à l'aide d'une liste de critères connus. Le walkthrough requiert plutôt de tenter d'effectuer une simulation du fonctionnement du système décrit par l'artefact grâce à une série de scénarios (Adrion et al., 1993).

Bien que l'inspection et le walkthrough aient toutes deux comme objectif la détection d'anomalies, le walkthrough évolue souvent en une session où l'on effectue de la résolution de problème plutôt que de la vérification. L'inspection, par sa structure extrêmement formelle, ne permet pas ce genre de digression (Bisant et Lyle, 1989).

Existe-t-il vraiment une dichotomie entre les deux termes ? D'une part, on décrit l'inspection comme une étape de vérification formelle et d'autre part, le walkthrough

comme une revue informelle. Est-ce vraiment la cas ? Malgré ces différences, leur objectif principal est le même: améliorer la qualité du produit logiciel final.

### 2.2.2 Différentes méthodes de révision technique

Cette section décrit les méthodes de révision technique existantes. Ces méthodes ont toutes des similarités et des différences qui les rendent uniques. Ces méthodes serviront de base à l'établissement d'une définition générique de la réunion de révision technique.

#### 2.2.2.1 Les révisions par les pairs (Peer Reviews)

Cette méthode a été développée par Shneidermann (1980) et Myers (1979). Cette méthode fut à l'origine conçue pour l'évaluation du code source. Elle s'effectue en deux phases, soient la révision et le compte-rendu. Les participants assument un des deux rôles lors de la réunion: administrateur ou réviseur. Le code source est évalué au niveau de la qualité, la maintenabilité, l'extensibilité, la facilité d'utilisation et la clarté. Cette évaluation est effectuée anonymement par des ingénieurs logiciels ayant les mêmes antécédents et participant au même projet (des pairs). L'administrateur recueille le code source et le distribue aux réviseurs avant la réunion. Durant la phase de révision, les réviseurs évaluent la qualité du code source individuellement et de façon asynchrone. Cette phase consiste à lire le code source et l'évaluer afin de le noter sur un questionnaire en fonction des critères mentionnés auparavant. La révision est terminée lorsque le questionnaire est complété. Lorsque tous les réviseurs ont terminé, l'administrateur

recueille les questionnaires et prépare un compte-rendu.

#### 2.2.2.2 Le walkthrough structuré de Yourdon

Yourdon (1989) a conçu cette évolution des walkthroughs qui existaient déjà depuis longtemps dans l'industrie. Il s'agit d'une méthode de révision où l'auteur d'un artefact présente la description de son travail à un groupe de révision qui fournit ses commentaires tout au long de la présentation. Cette méthode commence par une préparation où les réviseurs se mettent au fait du contenu de l'artefact individuellement ou en discutant avec l'auteur. La réunion proprement dite suit cette préparation dans le but de détecter des anomalies. On utilise la paraphrase, c'est-à-dire que l'auteur présente son artefact un item à la fois. Un compte-rendu est préparé après la réunion.

#### 2.2.2.3 La révision "Round-Robin"

Cette méthode (Tjahjono, 1996) consiste à distribuer une copie de l'artefact à tous les réviseurs. Un par un, ceux-ci formulent leurs commentaires et les passent au réviseur suivant. La révision est terminée lorsque l'auteur reçoit l'artefact révisé. Le concept de cette méthode est basé sur l'identification d'anomalies et rien d'autre. L'interaction entre les réviseurs est limitée, si ce n'est le fait que seul l'auteur peut réviser les commentaires faits par tous les réviseurs.

#### 2.2.2.4 L'inspection de Fagan

Fagan (1976) fut le premier à tenter de structurer la révision de code en concevant l'inspection. Cette méthode est encore aujourd'hui considérée comme la base de plusieurs autres méthodes. Cette méthode débute par une brève phase de survol. Cette phase a pour but la compréhension de l'artefact et est exécutée grâce à une présentation informelle de l'auteur où les participants peuvent poser des questions. Cette première phase est suivie de la préparation indépendante et asynchrone des réviseurs qui tentent de comprendre le propos de l'artefact. La réunion d'inspection examine et valide par la suite les défauts identifiés individuellement.

L'apport de cette approche au domaine est la définition de rôles importants durant la réunion d'inspection:

1. Réviseur: Sa tâche est de détecter les anomalies dans l'artefact. Ces anomalies peuvent être des défauts, des omissions, des incohérences ou de la confusion;
2. Secrétaire: Il documente toutes les décisions ou discussions effectuées lors de la réunion.
3. Auteur: Il identifie et répare les anomalies.
4. Modérateur: Il gère la réunion d'inspection et facilite son déroulement.
5. Lecteur: Il paraphrase, section par section, l'artefact révisé durant la réunion.

#### 2.2.2.5 L'inspection de Gilb

Gilb (1993) a formalisé les étapes d'inspection. Cette méthode débute par une réunion de départ où le modérateur décrit les procédures administratives de l'inspection. La vérification individuelle permet aux réviseurs de vérifier l'artefact afin d'y déceler des anomalies potentielles. Les réviseurs ont des responsabilités qui influencent leurs révisions: utilisateur, inspecteur, système, qualité, normes, etc. Des listes de critères sont utilisées pour chacune des responsabilités. Le catalogage suit, où les réviseurs se réunissent pour mettre ensemble leurs travaux respectifs. Durant cette réunion, le modérateur avance dans l'artefact section par section et les réviseurs émettent leurs commentaires au moment donné. Cette méthode comprend aussi des phases administratives telles que la modification et l'amélioration de la méthode d'inspection elle-même.

#### 2.2.2.6 L'inspection à 2 personnes

Bisant et Lyle (1989) ont conçu une méthode d'inspection n'impliquant que deux ingénieurs en logiciel qui chacun révise le travail de l'autre. La méthode est composée de deux étapes: (1) INSPECT-1, la conception et le code source du premier ingénieur sont révisés et (2) INSPECT-2, la conception et le code source du deuxième ingénieur sont révisés. L'objectif est de détecter des anomalies et non pas de trouver de nouvelles solutions. Une telle inspection doit être limitée à moins de 20 minutes.

#### 2.2.2.7 L'inspection sans réunion

Votta (1993) a conçu une méthode qui va à l'encontre de celle de Fagan. Cette méthode utilise des rencontres entre chacun des réviseurs et l'auteur individuellement, et non pas une rencontre unique avec tous les réviseurs présents.

#### 2.2.2.8 La révision de conception active

La révision de conception active (Parnas et Weiss, 1987) a été spécifiquement conçue pour la révision de documents de conception. Les réviseurs sont actifs en élaborant des affirmations à propos du rationnel de conception. Seuls des réviseurs ayant une expertise du domaine d'application peuvent y participer. La méthode comprend trois phases, la compréhension de groupe, la révision individuelle asynchrone et la consolidation des anomalies potentielles.

Ce type de révision a été conçu pour faire face à deux inconvénients majeurs:

1. Si les réviseurs ne peuvent adéquatement comprendre le document, il est peut probable qu'ils y détectent des anomalies; et
2. Une réunion de tout un groupe est inutile pour consolider les anomalies.

#### 2.2.2.9 L'inspection de Humphrey

Cette méthode (Humphrey, 1989) débute par une phase de survol effectuée lors d'une rencontre de groupe. Les réviseurs passent au travers de l'artefact présenté par l'auteur.

Cette phase est suivie de la préparation qui consiste dans la révision individuelle de l'artefact. Une nouveauté: une phase de pré-réunion qui a pour but de corréler les anomalies détectées individuellement. Le modérateur ou l'auteur regroupent les anomalies ensemble. La réunion de groupe a pour objectif de consolider les anomalies une à la fois lorsque les participants discutent et confirment la signification d'une anomalie.

2.2.2.10 Inspection par phases

Cette méthode (Knight et Myers, 1993) consiste en une série d'inspections partielles coordonnées (phases). Chaque phase cible un certain nombre de propriétés de l'artefact. Ces propriétés sont regroupées de façon à former un tout conceptuel, sinon une phase peut être divisée en sous-phases.

2.2.2.11 Inspection à N itérations

Cette méthode (Martin et Tsai, 1990) tente de reproduire une approche déjà existante (celle de Fagan par exemple) en utilisant N équipes indépendantes. Ils ont trouvé expérimentalement que chaque inspection découvre moins d'anomalies que la précédente. Cependant, on découvre que la plupart des anomalies de spécification est trouvée après la fin de la deuxième phase d'inspection.

### 2.2.3 Modèle de révision générique

Dans le cadre de son doctorat, Tjahjono (1994) a développé un modèle de révision technique générique. Le modèle à objectif unique est caractérisé par des étapes successives qui n'ont qu'un seul objectif intrinsèque (Figure 2.2). D'après ce modèle, chacune des étapes peut être effectuée individuellement ou en coopération. La révision technique est donc composée d'une étape de compréhension de l'artefact, suivie d'une étape d'analyse qui identifie les anomalies. Lorsque l'analyse est effectuée individuellement, une étape de corrélation est nécessaire afin d'établir une liste d'anomalies commune au groupe. La dernière étape, la consolidation, consiste à vérifier la validité des analyses faites, c'est-à-dire d'évaluer la pertinence des anomalies trouvées.

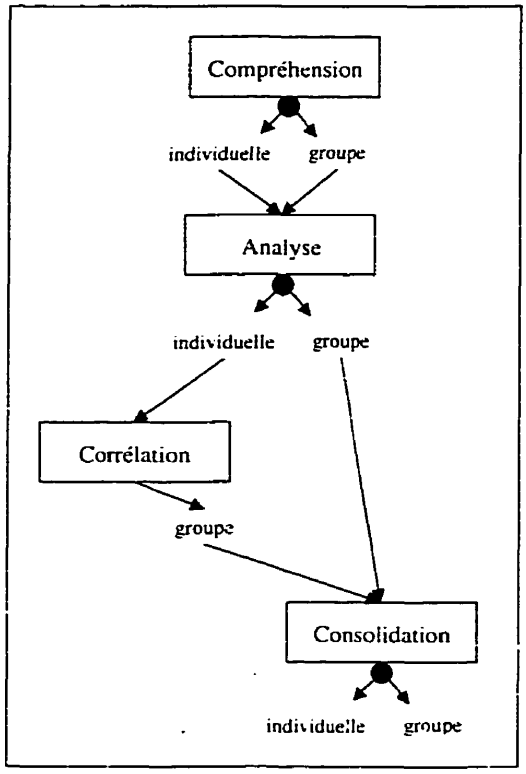



**Figure 2.2 Modèle de révision technique générique à objectif unique**

Les trois plus importantes méthodes (Yourdon, 1989; Fagan, 1976; Humphrey, 1989) utilisées dans l'industrie en ce moment peuvent être caractérisées par ce modèle générique.

### 2.2.4 Variations entre les méthodes de révision

Il existe un grand nombre de méthodes de révision et de nouvelles méthodes sont encore conçues. Bien que leurs objectifs soient identiques, leurs façons d'y parvenir diffèrent. Ces variations, qui sont souvent des contradictions, existent sous plusieurs formes, que cette section tente de regrouper.

#### 2.2.4.1 Composition des équipes

Dans la plupart des méthodes, le nombre de réviseurs varie entre un et six. La taille des équipes de révision est pourtant souvent un point de litige. Weller (1993) a mesuré qu'une équipe de quatre réviseurs était plus de deux fois plus efficace qu'une équipe composée de trois réviseurs. Buck (1981), pour sa part, a démontré qu'il n'y avait aucune différence entre l'efficacité d'équipes composées de trois, quatre ou cinq réviseurs. Porter et al. (1997) conclut que la taille d'une équipe n'influence pas la taux de détection des anomalies. Les techniques de détection des anomalies sont, d'après eux, le facteur qui influence l'efficacité des révisions techniques et non la taille de l'équipe.

#### 2.2.4.2 Une session vs multi-session

Traditionnellement, les révisions techniques sont effectuées en une seule session (Fagan, Gilb et Humphrey). Cependant de nouvelles méthodes telles que l'inspection par phases et l'inspection à N itérations ont été conçues récemment. Porter et al. (1997) ont démontré que l'efficacité de détection d'anomalies n'était pas significativement liée au nombre de sessions de révision utilisées. Ils concluent qu'une révision technique en une seule session effectuée par une petite équipe est la meilleure combinaison de facteurs. On devrait maintenant se concentrer sur l'amélioration des techniques de révision utilisées.

#### 2.2.4.3 Révision individuelle vs coopérative

On s'entend pour affirmer qu'une forte proportion des anomalies sont détectées dès la

phase de compréhension à cause de la communication entre l'auteur et les réviseurs. Pourtant, Ackermann et al. (1989) et Russell (1991) découragent encore la participation active de l'auteur de l'artefact révisé dans la réunion de révision. D'autre part, Humphrey (1989) démontre que la majorité (75%) des anomalies sont détectées lors du travail de préparation individuel.

#### 2.2.4.4 Révision avec ou sans réunion

Parnas et Weiss (1987) affirme que les réunions sont inutiles pour la détection des anomalies. Votta (1993) a démontré qu'une méthode sans réunion a été plus efficace dans la détection des anomalies. Pourtant, les méthodes généralement utilisées dans l'industrie (Fagan, Humphrey) sont des révisions coopératives.

#### 2.2.4.5 Détection des anomalies

La détection d'anomalies peut s'effectuer intuitivement ou systématiquement. Les listes de critère et les techniques *ad hoc* sont intuitives tandis que l'utilisation de scénarios ou de preuves d'exactitude sont très systématiques. Bien que les techniques intuitives soient largement utilisées, quelques uns (Porter et al., 1994) stipulent que les techniques systématiques donnent des responsabilités spécifiques aux réviseurs afin d'assurer une couverture complète de l'artefact révisé. Le résultat est un plus haut taux de détection et des bénéfices plus importants que les techniques intuitives.

### 2.2.5 Bénéfices et efficacité de la révision technique

Le génie logiciel fêtera son trentième anniversaire d'ici peu. L'utilisation de la révision technique comme pratique de vérification d'un produit logiciel avant sa réalisation a émergé en 1974 grâce aux recherches de Fagan. Depuis ce temps, plusieurs entreprises ont implanté avec succès cette pratique dans leurs processus de développement (Ackermann et al., 1989). La détection et la correction d'anomalies le plus tôt possible dans le processus ne pouvait qu'améliorer la qualité des logiciels. Fagan (1986) a pour sa part observé que l'utilisation des révisions techniques diminuait le nombre de défauts générés par un ingénieur logiciel lors de la conception et de la réalisation, que ce soit dans des petites ou grandes organisations. Les rapports de projets utilisant la révision technique qui ont été étudiés indiquent clairement que la qualité a été améliorée et que les coûts de développement ont diminué.

La révision technique est avantageusement comparée aux tests puisqu'elle peut être effectuée tôt dans le processus de développement, même avant qu'une seule ligne de code n'existe (Porter et al., 1995). Plusieurs recherches ont démontré qu'en moyenne 31,2% des défauts retrouvées lors des tests finaux d'un logiciel proviennent du travail effectué lors de la phase d'analyse et de conception, 56,7% durant l'implantation et 12,1% lors des tests de modules et d'intégration (Ragland, 1992). Ces mêmes recherches ont aussi démontré que l'utilisation des réunions de révision technique comme pratique du processus de développement permet de détecter 65% des anomalies générées par ces

phases et qui ne sont donc pas présentes lors des tests finaux du logiciel.

Collofello (1988) résume l'importance de la révision technique comme suit:

1. les réunions de révision technique facilitent la détection d'anomalies tôt dans le développement, avant que le coût de réparation ne devienne trop grand;
2. les révisions techniques complètent les tests qui ne sont pas suffisants;
3. les révisions techniques permettent de faire un suivi adéquat du projet;
4. les révisions techniques fournissent de l'information à propos du produit et du processus de développement;
5. les révisions techniques sont une plate-forme d'apprentissage où les réviseurs peuvent acquérir une meilleure compréhension du produit logiciel en lisant la documentation et en observant le travail des autres;
6. les révisions techniques améliorent la maintenabilité du logiciel en assurant une meilleure documentation, donc une meilleure compréhension globale du produit lui-même.

Selon Porter et al. (1995), deux problèmes fondamentaux diminuent l'efficacité des révisions techniques: le ratio coût-bénéfice des révisions techniques n'est pas mesuré et l'agent responsable de l'augmentation des bénéfices ou de la diminution des coûts n'est pas identifié. L'efficacité de la révision technique dépend grandement des techniques utilisées puisqu'une démarche systématique révèle un plus grand nombre d'anomalies qu'une démarche intuitive.

Selon Myers (1978), la limite d'efficacité des réviseurs est certainement l'accent trop prononcé de la démarche de révision sur la logique du programme au dépend des entrées-sorties ou des cas spéciaux.

### 2.2.6 Définition générique de la réunion de révision technique

À partir de la revue de littérature présentée dans ce chapitre, il est maintenant possible de définir les réunions de révision techniques comme suit (Robillard et als. 1998):

> *Une rencontre formelle durant laquelle un groupe de pairs effectue l'analyse systématique d'un artefact (analyse, design, code, test, etc.) afin d'y déceler des erreurs, en fonction de barèmes existants, et de les corriger. Le produit d'une telle réunion est un nouvel artefact utilisable lors de l'étape subséquente de développement.*

L'objectif de cette définition est de regrouper toutes les méthodes de révision (inspection, walkthrough, etc.) coopératives et de leur donner un seul nom générique.

## 2.3 Travail individuel et coopératif

La révision technique est le cadre d'un travail individuel (cognitif) et coopératif de la part des réviseurs et de l'auteur. L'implantation d'une démarche de révision efficace doit donc tenir compte de ces facteurs. La nature même de la révision technique, dont les coûts

sont presque entièrement humains, influence son efficacité.

Wienberg (1971) a été l'un des premiers à étudier l'aspect psychologique de la révision. Le concept *egoless* en a découlé:

1. L'habileté de trouver soi-même une erreur est modifiée par la tendance d'une personne à justifier ses actions. Il est donc plus efficace d'avoir une autre personne pour vérifier son travail; et

2. L'on doit être détaché de son travail. Un artefact doit être considéré comme une propriété de l'équipe de développement et peut donc être critiqué librement.

Ce concept a contribué à la formation d'équipes de révision et plus tard, à la mise en branle de groupes indépendants d'assurance qualité.

Humphrey (1989) donne deux raisons autres que l'amélioration de la qualité du produit pour justifier l'utilisation de la révision technique:

1. Les révisions techniques permettent à l'individu de se créer une image mentale du fonctionnement du programme et pour quelles raisons. Lorsque le programme ne fonctionne pas comme prévu, il est plus facile de trouver et réparer le problème; et

2. La révision technique permet de détecter l'anomalie directement tandis que

d'autres techniques telles que les tests n'identifient que les symptômes.

Letovsky et al. (1987) ont démontré qu'un réviseur tente d'accomplir trois choses lors d'une révision technique: la clarté, la cohérence et l'exactitude. Pour ce faire, trois comportements ont été identifiés lors de ces réunions: la reconstruction conceptuelle, la simulation mentale et la vérification de documents. Ces comportements représentaient plus de 89% du temps consacré à la révision d'un artefact.

L'amélioration de la réunion de révision technique passe certainement par l'identification des comportements clés qui caractérisent ce type de réunion. La littérature s'accorde pour dire que ce type de réunion n'a peut-être pas un seul objectif de vérification mais aussi plusieurs autres cachés. Une avenue de recherche pourrait passer par l'identification et la mesure des activités individuelles et coopératives lors de ces réunions afin de déterminer exactement le genre de travail effectué.

## 2.4 Sommaire

Tjahjono (1996) affirme que la raison principale de l'existence des contradictions entre spécialistes est le manque de spécificité dans l'identification des facteurs qui influencent l'efficacité d'une méthode de révision.

La section précédente a présenté les questions que se posent quelques spécialistes quant à la vraie utilitée des réunions de révision technique. Ce type de réunion est peut-être plus que seulement un exercice de détection d'anomalies. Bien qu'une étude ait conclu

qu'une réunion de révision effectuée en une seule étape soit la méthode d'évaluation la plus efficace, elle était basée sur la comparaison et non sur les caractéristiques intrinsèques de ce type de réunion. Il est maintenant important de tenter de comprendre les activités retrouvées lors de ces réunions afin de les améliorer (Porter et al., 1997).

Plus récemment, Seaman et Basili (1998) ont effectué une étude empirique de la communication lors de réunions de révision technique (inspections de code) dans le cadre d'un projet de développement logiciel. Leur objectif était de mesurer jusqu'à quel point la structure organisationnelle avait un impact sur l'effort requis pour la communication. Leurs résultats démontrent que les relations de travail passées et présentes entre les réviseurs influencent directement le temps consacré dans différents types de discussions. La conclusion de cette étude est qu'une nouvelle avenue de recherche est requise pour approfondir notre compréhension des phénomènes coopératifs en génie logiciel. La mesure et la compréhension des phénomènes organisationnels ont le potentiel d'influencer profondément le succès de futurs projets de développement logiciel.

# CHAPITRE III

# PROBLÉMATIQUE DE RECHERCHE

Le développement d'un logiciel dépend grandement du travail mental (cognitif) des individus. Il est donc raisonnable de penser que la démarche coopérative utilisée lors du développement aura un impact important sur la qualité du produit final. Un processus de développement logiciel définit un cadre de travail coopératif où les intervenants peuvent avoir différents rôles (par ex, chef de projet, analystes, programmeurs, clients). Sur la base de ces processus, des pratiques sont développées: elles spécifient quelles sont les différentes étapes nécessaires au développement d'un logiciel, leur ordonnancement, les différents résultats attendus, la structure et la notation des documents utilisés, et elles permettent d'estimer l'effort requis dans le développement d'un logiciel.

Une limite de ces processus et pratiques utilisés en génie logiciel est qu'ils rendent peu compte de l'activité de développement de logiciel dans ses aspects individuels et coopératifs. Cela amène quelques inconvénients:

1. Même si une pratique prescrite permet de guider l'activité, son utilisation effective peut être déviante et problématique par rapport aux prescriptions; et

2. On modifie la prestation d'une pratique sans se soucier de l'impact de cette

modification comme, par exemple, lorsqu'on décide de modifier unilatéralement une méthode de révision technique.

D'un point de vue cognitif et coopératif, les déviations témoignent d'une certaine non-compatibilité entre les modèles du génie logiciel et les modèles cognitifs de l'activité. Il semble donc important d'étudier ces activités d'un nouveau point de vue qui permette de guider le développement et l'évolution de certaines pratiques du génie logiciel.

## 3.1 Une pratique clé : la réunion de révision technique

La réunion de révision technique est l'une de ces pratiques du génie logiciel qui semble dévier des méthodes prescrites. La revue de la littérature effectuée sur les réunions de révision technique a permis de se rendre compte qu'il semble exister autant de méthodes de révision que de spécialistes qui se sont penchés sur la question.

Fagan a développé sa méthode d'inspection formelle à partir de son expérience de travail chez IBM. Les autres qui ont suivi, et ce jusqu'à aujourd'hui, ont toujours justifié l'élaboration d'une nouvelle méthode de révision par l'observation de comportements déviants lors de l'utilisation de la méthode originale dans un environnement spécifique. Bref, la littérature est pleine de méthodes de révision prescrites à partir de l'expérience de travail unique d'un auteur. Malgré qu'on ait réussi à développer un modèle générique de révision (Tjahjono, 1996), il existe toujours des comportements qui ne semblent pas appartenir à ce modèle.

Le génie logiciel est rempli de pratiques définies à partir de l'expérience particulière d'un individu. Bien que cette façon de faire soit loin d'être mauvaise, elle est contestable scientifiquement. En tant que domaine scientifique, le génie logiciel doit pouvoir justifier ses pratiques à l'aide de fondements théoriques valables et non pas seulement à partir d'expériences. La difficulté réside dans l'origine de ces fondements théoriques: on tente de formaliser l'ingénieur logiciel lui-même et non pas le produit logiciel. La réponse peut venir d'autres domaines de recherche tels que les sciences cognitives et l'ergonomie. L'ergonomie se spécialise, entre autres, dans l'observation et la mesure d'une pratique existante dans le but d'y apporter des améliorations. Le domaine du génie logiciel peut emprunter les techniques et outils développés pour l'ergonomie dans sa volonté de vouloir améliorer ses pratiques.

### 3.1.1 Compréhension avant amélioration

On ne peut améliorer ce que l'on ne comprend pas. On peut raisonnablement affirmer que la réunion de révision technique, sous toutes ses formes, est utile pour le génie logiciel. On ne peut toutefois pas indiquer précisément quels aspects de la réunion de révision technique la rendent utile. En fait, on n'a peut-être pas identifié toutes les activités coopératives qui caractérisent cette pratique.

Un retour en arrière est nécessaire. Il faut prendre un pas de recul pour observer comment les ingénieurs effectuent la révision technique, comprendre comment ils le font pour, par la suite, suggérer des améliorations potentielles. Cette compréhension formelle servira de

base à l'élaboration de nouvelles méthodes de révision technique, qu'elles soient traditionnelles ou automatisées.

## 3.2 Description des problématiques

Cette recherche tente donc d'identifier et de mesurer le travail coopératif afin d'utiliser les résultats pour proposer des améliorations aux pratiques de révision technique. Une recherche de ce genre fait face à deux importantes problématiques: (1) comment représenter efficacement les activités cognitives et coopératives et (2) comment obtenir des résultats utiles et généralisables au domaine du génie logiciel.

### 3.2.1 Représentation des activités cognitives et coopératives

La représentation est un filtre, une abstraction des éléments jugés importants par le chercheur, de la situation observée. Cette représentation est par la suite utilisée pour obtenir des résultats.

Une représentation doit être capable de reproduire une situation de façon cohérente dans le temps et dans l'espace. Une représentation qui est faussée de façon systématique peut être parfaitement fiable mais manquer de validité. La validité se définit comme la capacité d'une représentation à mesurer le phénomène étudié, c'est-à-dire l'adéquation qui existe entre les variables retenues et le concept théorique à mesurer.

Une représentation formelle est utilisée pour représenter, dans une syntaxe formelle, les

comportements observés durant les réunions. Étant donné que cette représentation est à la base de toutes les mesures subséquentes, sa fiabilité comme instrument de mesure est très importante. La validation de cette représentation formelle doit être effectuée de deux façons: quantitativement en observant les résultats et qualitativement à l'aide de la coopération entre experts en cognition et en génie logiciel.

La représentation doit permettre d'identifier, analyser et comprendre les comportements cognitifs et coopératifs importants qui caractérisent une réunion de révision technique et qui la rendent utile au processus de développement de logiciels.

La représentation doit répondre aux intérêts théoriques et être objective (Olson et al., 1994). L'objectivité dépend de la validité et de la confiance données à la représentation tandis que l'intérêt théorique dépend principalement du domaine et de l'objectif de la recherche. La représentation doit permettre de coder les données adéquatement tout en étant assez formelle pour supporter des analyses quantitatives qui seront faites automatiquement. La représentation doit répondre aux critères suivants:

1. La fiabilité de toute mesure est souvent représentée par la reproduction des résultats obtenus. La représentation formelle doit donc être suffisamment précise et orthogonale pour permettre d'éliminer le plus possible les interprétations personnelles. Il doit exister un minimum d'ambiguïté et l'introduction de "bruits" dans la représentation doit être réduit.

2. La représentation doit être décrite dans un formalisme précis afin d'assurer sa validité et sa complétude. Puisque la représentation formelle est le fruit d'une certaine démarche scientifique et théorique, sa formalisation doit assurer l'existence d'une rigueur intellectuelle dans son choix et son utilisation.

3. La représentation formelle doit permettre la transformation des données brutes en un ensemble de données qui pourront ultérieurement être analysées. Il est donc important qu'elle représente le plus précisément possible tous les comportements retrouvés dans une réunion de révision. La représentation doit donc être relativement proportionnelle, c'est-à-dire qu'il n'y ait pas de composantes de la représentation qui caractérisent la majeure partie des données contre d'autres qui n'en caractérisent qu'une infime partie.

Une représentation qui s'appuie sur des bases théoriques formelles provenant autant du domaine de la cognition que du génie logiciel a été conçue dans le cadre de cette recherche. Cette représentation utilise quelques caractéristiques provenant d'autres représentations existantes mais est dans son ensemble une contribution nouvelle au domaine du génie logiciel. Elle est la fondation de l'approche de recherche utilisée.

La démarche de recherche ne doit pas être influencée par des prémisses ou des hypothèses mais plutôt être considéré comme une démarche exploratoire. On ne tente pas de prouver qu'une certaine façon de faire des révisions techniques est la meilleure, on tente plutôt de

comprendre les phénomènes présents lors de ces révisions techniques.

### 3.2.2 Application des résultats au génie logiciel

L'utilité de l'approche préconisée dans ce travail est basée sur deux caractéristiques: (1) sa capacité à produire des résultats dans un délai raisonnable et (2) la généralisation des résultats au domaine du génie logiciel.

La plus grande difficulté de cette recherche est de pouvoir obtenir des résultats généralisables de nature à pouvoir améliorer le génie logiciel. L'une des premières considérations est le temps d'analyse requis. La représentation formelle doit permettre une analyse simple et relativement rapide afin de réduire le temps d'analyse et de mesure. Il est certain qu'une représentation détaillée permet de coder plus en détails les comportements observés, mais cela comporte des risques d'erreurs plus grands. Il s'agit de trouver l'équilibre entre la simplicité d'analyse et le détail de la couverture de cette représentation.

La syntaxe utilisée pour exprimer cette représentation formelle doit faciliter la mesure à l'aide d'outils automatiques. Le temps d'analyse en sera considérablement réduit et, ce qui est très important, l'ajout de biais dans les analyses sera circonscrit au codage des données seulement. L'utilisation d'analyses automatiques assure une analyse uniforme des données.

Le développement d'une représentation formelle des données est définitivement la plus

importante étape de cette démarche exploratoire. Cette représentation constitue la pierre angulaire de tout le travail d'analyse ultérieur. Le codage des données brutes dans un vocabulaire précis permet d'identifier des patrons types et d'obtenir des mesures statistiques fiables. Le succès de la démarche d'analyse est dépendant en grande partie de la précision de cette représentation.

La pertinence de cette approche sera fonction de l'applicabilité au génie logiciel. Les réunions de révision technique devront être décrites selon trois aspects fondamentaux:

1. La mesure des activités individuelles des participants à la réunion permet d'identifier des comportements importants. L'identification de ces comportements permettra de comprendre comment l'effort des individus est utilisé lors de ce type de réunion;

2. L'identification de rôles dans la réunion et leurs influences sur les comportements des participants ajoutera de l'information quant aux raisons d'occurrence de certaines conduites dites "déviantes". Les rôles d'auteur et de réviseur sont-ils les seuls à exister ?

3. Une réunion de révision, par sa nature, doit être caractérisée par d'autres activités coopératives que la détection d'anomalies. La prise de décision et la solution de problème sont sûrement présentes. Existe-t-il d'autres activités coopératives intrinsèques à la réunion de révision technique qui rendent

l'utilisation de ce type de réunion nécessaire dans le processus de développement ?

## 3.3 Sommaire

L'objectif de notre projet est de construire un modèle des activités cognitives individuelles et coopératives mises en œuvre lors des réunions de révision technique. Les documents révisés ont une double fonction, fonction d'aide à la conception et fonction communicative, ce qui entraîne des activités cognitives de nature différente.

La démarche de recherche consiste à observer dans un environnement naturel et professionnel le comportement d'ingénieurs en logiciel, et d'élaborer, à partir de ces données, un modèle de la pratique. Le génie logiciel requiert de chaque ingénieur logiciel des aptitudes à pouvoir imaginer et créer des entités abstraites et surtout à être capable de relier ces entités ensemble grâce à des liens qui eux aussi sont abstraits. Ce sont ces activités cognitives individuelles qui rendent le travail en équipe très difficile car il n'existe pas deux personnes qui imaginent la même solution à un problème précis. Le génie logiciel vient en aide à ces gens en leur donnant des processus et des pratiques qui permettent de mieux comprendre leur rôle dans le développement d'un logiciel. Ces pratiques ont été créées dans le but de donner un cadre relativement rigide de travail aux ingénieurs et en leur permettant de visualiser les étapes de développement d'un logiciel.

Olson et Herbsleb (1995) mentionnent que le rôle joué par les membres de l'équipe lors

des réunions est un aspect important mais qu'il n'a pu encore être mesuré. D'un autre côté, Thomsett décrit un modèle (Belbin) qui permet d'établir les rôles nécessaires à la bonne marche d'un groupe. Il est bien vrai que l'efficacité d'une équipe est basée sur les membres qui en font partie, mais il s'agit de déterminer si les rôles (postes) qu'ils jouent dans l'organisation ont un impact sur le contenu et la fréquence des conversations.

Une étude systématique des réunions de révision technique qui détermine les activités coopératives faites par les participants ainsi que les facteurs qui influencent l'occurrence de ces activités permettra de mieux comprendre leur fonctionnement intrinsèque. Cette meilleure compréhension permettra d'explorer de nouvelles méthodes qui amélioreront peut-être ce type de réunions.

# CHAPITRE IV

# DESCRIPTION DE LA MÉTHODE DE RECHERCHE

## 4.1 Stratégie de recherche

### 4.1.1 Objectif de la recherche

Une meilleure compréhension des méthodes de travail dans un domaine particulier est souvent basée sur des études empiriques et le génie logiciel n'y fait pas exception. L'étude des phénomènes cognitifs et coopératifs présents lors du développement de logiciels peut donc permettre d'améliorer cette discipline. Les études empiriques utilisent des techniques telles que l'analyse de tâches, l'analyse de protocoles ou l'analyse de discours pour obtenir les données nécessaires au développement de modèles représentatifs d'une situation donnée. Les sciences cognitives ont une longue tradition de recherches empiriques. Elles ont développé et validé des méthodes d'analyses efficaces.

Les activités effectuées par les ingénieurs logiciels peuvent être étudiées selon une perspective de travail individuel ou de travail coopératif (Figure 4.1). Les activités cognitives individuelles, qui sont le sujet de la plupart des études empiriques en génie logiciel, réfèrent aux mécanismes mentaux, stratégies et connaissances utilisés par les individus pour effectuer une tâche.

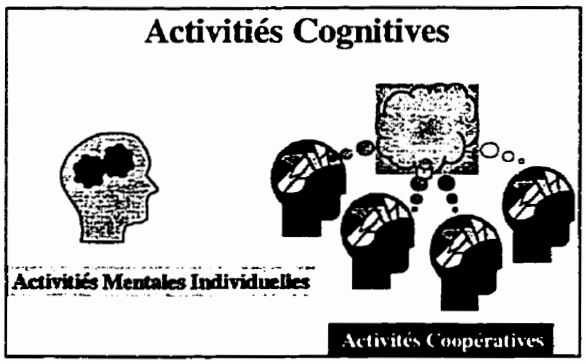



**Figure 4.1 Perspectives d'études du travail de l'ingénieur logiciel**

Le génie logiciel est pourtant un domaine où la coopération est de mise. Un processus de développement permet de structurer et faciliter le travail d'une équipe de développement en introduisant des méthodes de communication tels que documents et réunions.

On peut représenter l'approche de recherche de la façon suivante (Figure 4.2): on effectue des observations d'une pratique du processus logiciel telle que la réunion de révision technique et on représente ces observations dans une syntaxe formelle qui facilite les analyses subséquentes. Le résultat de cette démarche est une meilleure compréhension de la pratique qui permet d'y apporter des améliorations.

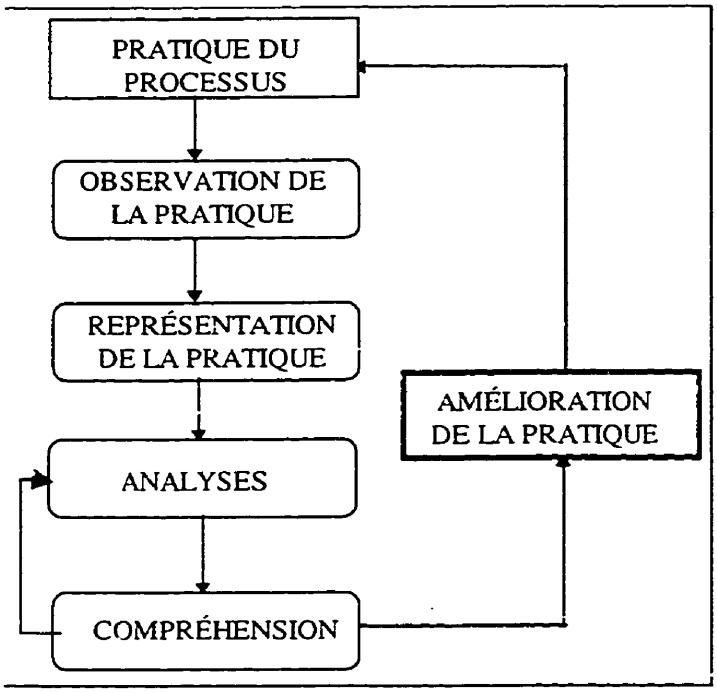



**Figure 4.2 Approche générique de recherche**

L'objectif global du projet de recherche est d'établir et valider une méthode qui permettra d'observer et surtout d'analyser le travail coopératif des ingénieurs logiciels lors de réunions de révision technique.

### 4.1.2 Approche théorique

L'objectif de la recherche est d'établir une méthode d'observation et d'analyse des activités coopératives lors de réunions de révision technique dans le cadre du processus de génie logiciel. L'optique de cette approche n'est pas d'utiliser cette méthode afin de déterminer un modèle théorique par hypothèse et de le tester mais plutôt de créer un modèle théorique à partir de données empiriques. Il s'agit donc d'un travail exploratoire.

Ce genre de recherche est connu sous le nom anglais de "Exploratory Sequential Data Analysis" (Analyse exploratoire et séquentielle des données) (ESDA) (Sanderson et Fisher, 1994). Cette approche favorise l'analyse de systèmes, environnements ou données comportementales dont l'intégrité séquentielle a été préservée. L'analyse de ces données doit correspondre à des questions de recherches, être guidée par une méthodologie, des pratiques et peut être considérée comme un travail exploratoire. L'utilisation de l'ESDA et son application dans la stratégie de recherche constituent la base de ce projet. Notre projet fait bien partie de la famille des études ESDA parce qu'il a les caractéristiques suivantes:

1. Les analyses nécessaires à l'étude des réunions de révision technique sont liées à la séquence et à l'organisation temporelle des activités observées:

2. La méthode utilisée pour obtenir les résultats escomptés à partir de la question de recherche en fonction du codage et des analyses est une approche viable et efficace; et

3. Les bénéfices conceptuels apportés par l'ESDA dans ce projet valent le temps additionnel nécessaire pour l'utilisation de cette approche.

### 4.1.3 Principes fondamentaux

L'ESDA est une méthode de recherche dérivée de l'EDA (Exploratory Data Analysis) (Tukey, 1977). C'est une méthode simple, visuelle et quantitative d'analyse de données

qui permet à un chercheur d'avoir une compréhension plus vaste en "observant les données afin de comprendre ce qu'elles disent".

Il existe quelques principes fondamentaux qui doivent être suivis afin de réussir dans l'utilisation d'une telle méthode, voici trois principes importants (Hartwig et Dearing, 1970):

1. **Ouverture d'esprit et itération**: Il ne faut pas imposer un modèle a priori, plutôt s'en tenir à des idées générales pour ensuite les détailler de plus en plus. Chaque itération permet de répondre à des questions mais aussi d'en générer de nouvelles. Le but est de parvenir à un niveau de compréhension du phénomène observé en un minimum d'itérations.

2. **Scepticisme**: Le chercheur doit avoir une approche soupçonneuse dans le cadre de ses observations. Les données numériques doivent être vérifiées et revérifiées pour assurer une modélisation stricte. Il faut tenter de comprendre et certainement expliquer pourquoi certaines données n'entrent pas dans le modèle théorique et utiliser des analyses quantitatives qui permettent de justifier les résultats.

3. **Exploration plutôt que confirmation**: Le but de cette démarche n'est pas de tenter de confirmer un modèle théorique par l'analyse de données mais plutôt d'utiliser ces données pour explorer le sujet afin de créer un modèle.

### 4.1.4 Traditions et techniques de l'ESDA

Les concepts formels forment la base à partir de laquelle une question de recherche est formulée et définissent les points importants qu'elle devrait contenir. Ces concepts indiquent aussi les types d'opérations et les transformations utilisables dans le travail de recherche. Sanderson et Fisher (1994) proposent trois différentes traditions: comportementale, cognitive et sociale. Chacune d'elle amène son propre bagage de techniques et théories. Voici le résumé de chacune de ces traditions:

Tradition comportementale. Cette tradition inclue des activités de recherche telles que l'analyse des facteurs humains, l'analyse temps-méthode et l'étude de théories de développement basées sur les analyses statistiques de données, en particulier les analyses de données séquentielles. Le but de cette tradition est d'obtenir des résultats qui sont objectifs et applicables dans plusieurs conditions.

Les chercheurs font appel à des théories d'échantillonnement et de mesure pour analyser les données. Ces techniques utilisent habituellement des critères tels que le temps, le choix des sujets, le matériel, l'utilisation d'une code unique et la gestion du biais de l'observateur. Les codages sont généralement des expressions simples et sommaires de données brutes.

Les théories d'échantillonnement guident le choix des données brutes en fonction des sujets et des situations. La verbalisation utilisée dans ce genre d'observation consiste en

activités faites naturellement dans le cadre d'interactions de groupe lors de l'accomplissement d'une tâche quelconque. Dans la plupart des cas, le codage de données brutes doit être fait par un humain.

Dans la tradition comportementale, il existe deux types d'analyses statistiques: séquentielles et non-séquentielles. Le calcul de la fréquence d'un événement particulier et la combinaison de tels événements sont les analyses non-séquentielles les plus communes. Les analyses séquentielles permettent de trouver des patrons dans les séquences de données et leurs dépendances dans le temps.

L'approche comportementale offre des procédures d'analyse qui demandent seulement que les données soient recueillies et préparées dans une forme standard afin qu'elles ne biaisent pas les hypothèses et les tests statistiques auxquels elles seront soumises. Lors de l'étape de confirmation des résultats, il est primordial que la grille de codage et les tests permettent de rendre l'hypothèse initiale opérationnelle.

Tradition cognitive. Cette tradition inclue des activités de recherche tels l'analyse de tâches, l'analyse de protocole, l'analyse de discours, le développement de modèles numériques à partir de comportements humains et la recherche de structure hiérarchique à partir d'approches d'analyses grammaticales. Cette approche est habituellement utilisée lors de recherches en laboratoire malgré qu'elle ait été utilisée lors d'études "*in the field*". La rigueur d'une étude est en fonction du rapprochement du modèle résultant avec les

données recueillies initialement.

L'approche cognitive dépend grandement de la verbalisation parce qu'elle permet aux analystes d'inférer les activités cognitives. Les transcriptions de verbalisations tendent à se concentrer sur le contenu de l'énoncé plutôt que la raison de son apparition. Elles sont habituellement divisées en segments (itérations) qui reflètent des éléments du processus cognitif. Dans plusieurs cas, la suite séquentielle est plus importante que le temps réel, il n'est donc pas toujours important de savoir à quel moment absolu une donnée est recueillie.

La plus grande différence entre l'approche comportementale et celle cognitive se situe dans les étapes d'analyse. Dans l'approche cognitive, l'analyste doit associer les relations entre chacun des éléments (acte, état ou vocalisation) à un tout qui définit un processus cognitif cohérent. Par exemple, bien que l'étude de G.M. Olson et al. (1994) ait été introduite comme une recherche dans la tradition comportementale, leur utilisation de grammaires constitue un trait de l'approche cognitive de l'ESDA.

Tradition sociale. La tradition sociale de l'ESDA est la réponse à un besoin d'observer les habitudes de travail des individus et leur utilisation de systèmes. Cette tradition est présentement celle qui présente le plus de travaux qui améliorent notre compréhension des interactions entre humains et ordinateurs. Cette approche se base donc sur la compréhension des aspects sociaux, organisationnels et matériels à l'intérieur desquels les individus et les groupes doivent interagir. Elle permet aussi de générer des comptes

rendus de la façon dont les individus et les groupes utilisent les ressources disponibles pour arriver à leur fin.

Le point de mire de cette approche est donc l'interaction sociale, et plus particulièrement la communication, le travail de groupe et aussi une grande variété d'orientations d'analyses. D'un côté, les chercheurs placent l'aspect cognitif individuel dans un contexte social tandis qu'à l'autre extrême, les chercheurs croient que d'inférer le processus mental individuel est inadéquat puisque la communication de la compréhension est un processus fondamentalement social. C'est certainement cette caractéristique qui permet aux utilisateurs de cette approche de reconnaître qu'il existe plusieurs interprétations des données observées.

Finalement, l'analyse des données est habituellement plus qualitative que quantitative. Elle dépend grandement de la recherche, cueillette, comparaison et interprétation des données plutôt que de la modélisation et des tests statistiques. L'analyse des interactions sociales se penche sur la compréhension des activités normales plutôt que sur des épisodes exceptionnels.

La démarche actuelle. Le résumé des trois approches traditionnelles nous donne une bonne idée de ce que l'utilisation de l'ESDA peut amener au projet. Il est donc maintenant nécessaire de déterminer où se situe le projet actuel par rapport à ces approches.

À première vue, le projet actuel est très similaire aux recherches d'Olson et al (1994) parce qu'il tente d'observer les comportements des individus lors d'une réunion de révision technique. La recherche peut donc être considérée comme faisant partie de la tradition comportementale puisqu'on tente de comprendre le comportement des participants à une réunion de révision dans le cadre du processus de développement de logiciels.

Un regard plus en profondeur des données recueillies et de l'objectif fixé montre cependant que les données verbalisées représentent aussi les activités cognitives individuelles dans le but d'obtenir une compréhension du problème en groupe. Nous avons donc affaire à un groupe d'individus ayant des activités cognitives qui influencent leurs propres comportements et ceux des autres participants. Ainsi, pour bien comprendre tous les aspects qui entrent en jeu lors d'une réunion telle que celle de révision, il faut travailler à l'intérieur de ces deux traditions (comportementale et cognitive).

La troisième tradition, sociale, semble se concentrer plus sur les interactions humains-machines, ce qui n'est pas le but de la présente recherche.

### 4.1.5 Définition et vue globale de la démarche

La Figure 4.3 présente une vue globale du projet de recherche en fonction des étapes du ESDA. La première ellipse identifie la question de recherche à élucider dans le cadre du

projet. Les carrés qui suivent représentent les données transformées à partir des réunions observées jusqu'au modèle de comportements. Les cercles identifient les tâches et outils nécessaires à la transformation et à l'analyse des données. Ces réunions ont habituellement des durées qui varient entre trente minutes et deux heures.

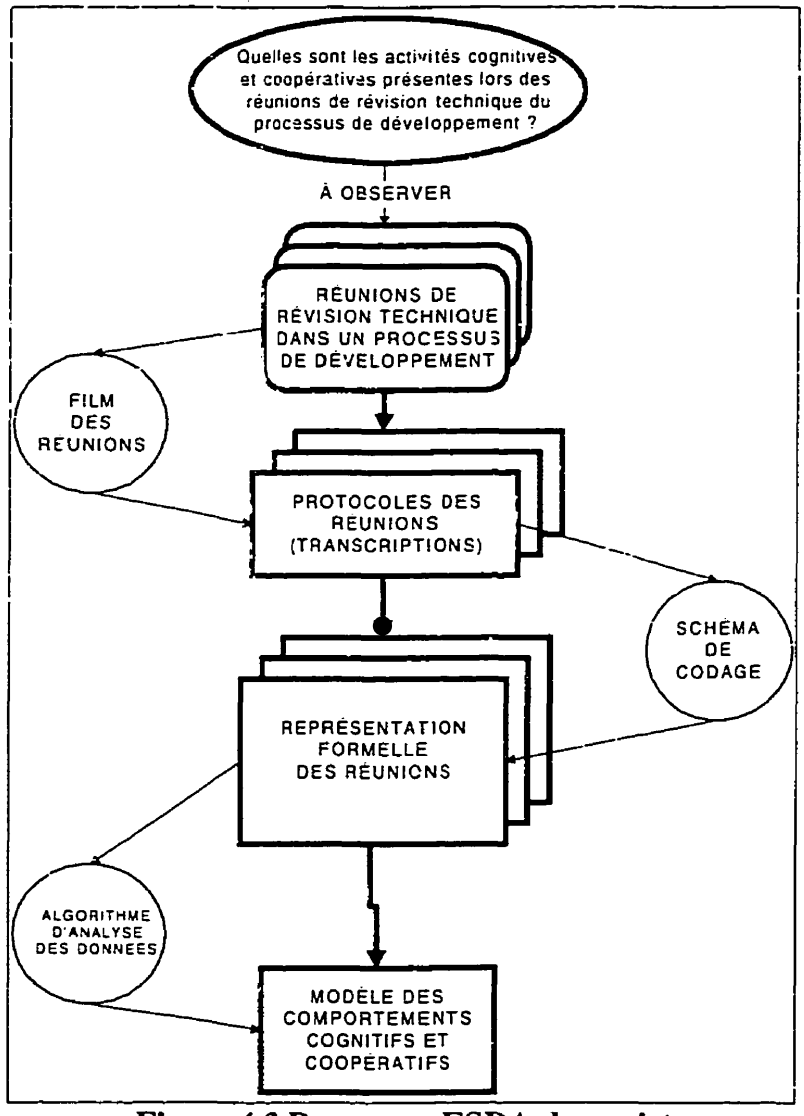



**Figure 4.3 Processus ESDA du projet**

Les enregistrements vidéos de ces réunions permettent de produire des transcriptions

détaillées des discussions à l'intérieur de celles-ci. Ces transcriptions sont le plus fidèles possibles à la réalité, c'est-à-dire qu'elles doivent refléter les différentes activités faites par les participants aux réunions. La principale difficulté provient du naturel des réunions. Les participants oublient qu'ils sont observés et utilisent des expressions familières ainsi que du vocabulaire extrêmement technique, parlent au même moment. s'interrompent et content des plaisanteries. Bien que cette conduite soit le bienvenu parce qu'elle démontre que le fait de les observer ne modifie que très peu le comportement naturel des participants, elle peut rendre la transcription des discussions difficile. La transcription produit les données de base des analyses subséquentes malgré qu'il soit quelquefois nécessaire de se référer à l'enregistrement pour valider une analyse.

L'étape suivante de la méthode consiste à coder les transcriptions dans un vocabulaire plus facilement analysable. Pour ce faire, un schéma de codage contenant une liste des principales activités retrouvées à l'intérieur d'une réunion doit être conçu et testé sur des transcriptions afin de s'assurer de sa validité. Une fois cela accompli, les données brutes sont converties dans un langage structuré ayant un nombre fini de possibilité. Ce schéma de codage est présenté plus loin.

Ces représentations formelles de réunion peuvent dès lors être analysées à l'aide de différentes techniques mathématiques et de solutions de patrons afin de déceler et identifier les comportements importants à l'intérieur d'une réunion. Un certain nombre d'algorithmes peuvent être conçus et utilisés à cette tâche. Le but de ces analyses est de

concevoir un modèle des comportements cognitifs des participants à une réunion.

Puisque la méthode ESDA est exploratoire, cette approche demande beaucoup de rétroactions et d'itérations. Le produit d'une étape du projet sera souvent revu et transformé en fonction de nouvelles informations découvertes. Le processus de transformation des données brutes en produits plus détaillés implique souvent plusieurs itérations qui pourront quelquefois changer les résultats attendus. Un chercheur devra quelquefois adapter ou tout simplement changer complètement de technique en fonction des circonstances.

### 4.1.6 Opérations de transformation

Le passage de données brutes vers un modèle théorique acceptable demande plusieurs transformations en ESDA. Sanderson et Fisher (1994) dénombrent huit transformations différentes, dont cinq sont utilisées dans la démarche actuelle:

1. **Paquets**: Les paquets correspondent aux regroupements des verbalisations contenues dans une réunion en entités distinctes et orthogonales pouvant être identifiés par une instance du schéma de codage.

2. **Commentaires**: Les commentaires sont des annotations utilisées pour ajouter de l'information considérée importante à un moment donné lors des différentes transformations. Lors du codage des transcriptions, par exemple, plusieurs commentaires sont ajoutés pour relier le contenu de la réunion ( à

partir d'un document) à la transformation.

3. **Codes**: Les codes sont le produit du codage des paquets en utilisant un schéma contenant les catégories de comportements cognitifs. Le processus de codage consiste à accoler une étiquette à chacun des paquets.

4. **Interactions**: Le codage des paquets traduit les données brutes en une série de codes structurés, appartenant au schéma de codage. L'analyse des interactions entre chacun des paquets donne un aperçu dynamique du comportement cognitif des participants à une réunion. Ces interactions peuvent être mesurées afin d'établir des patrons de comportements cognitifs et coopératifs.

5. **Calculs**: Le dernier type de transformation consiste à utiliser plusieurs algorithmes pour identifier les différents patrons dans les données. Ces calculs permettent d'établir des mesures statistiques utiles à la modélisation d'une réunion de révision technique.

### 4.1.7 L'aspect temporel du projet

Il est important de bien comprendre l'aspect temporel du projet de développement puisque la compréhension des données elles-mêmes et leurs séquences représentent l'objectif global de la démarche. Il faut donc examiner l'étendue des événements qui serviront de données et les contraintes qu'elles amènent sur les techniques d'ESDA utilisées:

1. **Processus cognitifs inférés**: Ces gestes correspondent à des activités qui ne sont pas directement observées, mais plutôt qui sont implicites à une certaine conversation. Les données brutes contiennent quelques activités de ce genre, telles que les fins de conversations, les coupures de paroles et les moments de pensées (les silences).

2. **Verbalisations**: Comportements des participants aux réunions qui sont directement observables à partir des vidéos et qui peuvent être facilement reproduits à l'aide d'une transcription écrite de la réunion.

3. **Réunions**: Les types de réunion (analyse, design, révision) influencent la nature des interventions. L'identification des activités cognitives et coopératives est peut-être liée au type de réunion comme par exemple, la réunion de révision technique.

4. **Projet**: L'incidence du moment de la réunion dans les phases du cycle de vie du logiciel ainsi que dans le projet lui-même peut influencer la nature des données recueillies.

La Figure 4.4 représente ces quatre importants critères de l'aspect temporel des données dans le projet.

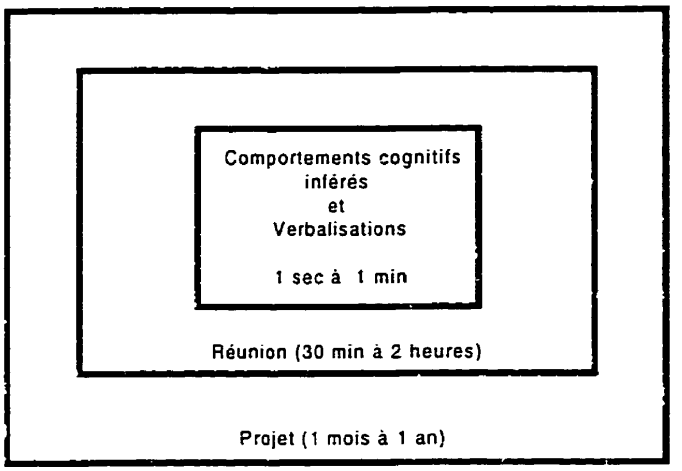



**Figure 4.4 Aspect temporel des données**

Le temps aura donc un impact sur les résultats obtenus. Il est donc très important de bien définir l'environnement utilisé en fonction du temps.

Une démarche exploratoire comme l'ESDA est un processus itératif qui mène vers une meilleure compréhension du phénomène observé. Le temps nécessaire à son application peut être considérable et donc un déterrant majeur à son utilisation. Le ratio TA:TS correspond au temps nécessaire à l'analyse des données (TA) par rapport au temps des séquences observées (TS). Le ratio TA:TS se situe habituellement aux alentours de 5:1 à 100:1 (Sanderson et Fisher, 1994), mais des ratios plus extrêmes sont possibles (5000:1, Ritter et Larkin, 1994). Le temps d'analyse devra être minimisé en identifiant bien les facteurs qui influencent sa durée et en utilisant des mécanismes qui le permettent. Voici quelques facteurs qui influencent le ratio TA:TS:

1. Préparation des données (travail effectué par l'humain):

a) Transcription des enregistrements vidéos (paquets); et

b) Codage des transcriptions grâce au schéma de codage (commentaires et codes);

2. Détermination des comportements (travail effectué par une machine):

   La façon de coder les paquets influence directement le degré de difficulté avec lequel la détermination des comportements peut se faire automatiquement.

3. Analyse et formation du modèle (humain et machine):

   Les analyses statistiques effectuées à l'aide de progiciels se font rapidement et les résultats obtenus doivent être interprétés par l'humain.

4. Complexité des concepts

   La complexité du codage et des analyses afin de créer un modèle véridique à partir des données recueillies aura un impact important sur le temps d'analyse.

Plusieurs recherches ont démontré que l'utilisation d'une notation structurée et d'outils d'analyses automatiques réduit le ratio TA:TS (Hoeim et Sullivan, 1994). Le ratio TA:TS peut être minimisé pour permettre une recherche avec des échéanciers et des résultats acceptables en utilisant des techniques d'analyses automatisées.

### 4.1.8 Adaptations et problèmes potentiels

Il n'est pas toujours simple d'intégrer la démarche à l'intérieur d'une tradition précise. Le travail consistera, en gros, à utiliser des techniques d'analyses statistiques et grammaticales dans le but de déterminer un modèle des comportements cognitifs. Une démarche utilisant l'approche ESDA doit être en mesure de répondre aux quatre questions suivantes: Quel est le sujet de la recherche, que doit-on observer, quelles opérations doit-on effectuer et quelles sont les réponses attendues ? Voici quelques problèmes potentiels qui pourraient surgir au cours du projet.

Connaissance du domaine d'application. Le travail de recherche en cours demande de bonnes connaissances dans le domaine des sciences cognitives ainsi que dans l'étude de comportements chez l'individu. Ces connaissances permettent de préciser les détails à observer pour obtenir des résultats significatifs et utilisables. Le défi majeur est sans aucun doute le codage des transcriptions de réunions à l'aide d'un schéma de codage. En effet, une expertise en cognition est nécessaire pour former un schéma adéquat. Cependant, cette expertise sera gaspillée s'il n'existe pas de codeurs capables de bien comprendre le domaine dans lequel ce schéma est appliqué, soit le génie logiciel. Il appert donc que les codeurs doivent absolument être des experts en informatique pour pouvoir effectuer un travail adéquat. La structure du schéma doit faciliter la tâche de ces experts.

Non correspondance d'une technique d'analyse avec le problème. La démarche ESDA

propose un grand nombre de techniques permettant d'analyser les données. Ces techniques doivent être en mesure de bien répondre aux objectifs de la recherche. Après que les questions de recherche sont identifiées, il faut donc faire une recherche des différentes techniques et décider si elles sont appropriées dans le cadre du projet actuel. Il est donc nécessaire d'employer les techniques les plus susceptibles de nous aider dans ce projet. Ce travail demande un minimum de connaissances de l'ESDA et de son application.

Manque de temps pour l'analyse. Le temps d'analyse des données représente la plus grande portion du temps nécessaire au présent projet. Il est possible de réduire TA en optant pour des techniques plus qualitatives ou par l'utilisation de logiciels pour automatiser une partie du travail. Pour l'instant, le TA a été réduit en choisissant un échantillon restreint mais représentatif de réunions à analyser. Ces réunions permettront de faire ressortir les points importants nécessaires à la réponse aux questions de recherche tout en minimisant le temps d'analyse.

Données trop riches. L'obtention d'analyses à partir de vocalisations humaines ne se fait pas sans difficultés. En effet, il s'agit de déterminer les aspects importants des conversations entre plusieurs participants à une réunion de révision technique. L'activité de codage est donc cruciale dans ce projet. Le schéma de codage doit refléter cette réalité en généralisant le plus possible les comportements des participants et en minimisant l'interprétation différente des données par plusieurs codeurs. Une formation adéquate de

chacun des codeurs réduit encore plus la chance d'erreur.

Non correspondance données-techniques. Le choix d'une technique d'analyse particulière doit être fait en fonction des données autant qu'en fonction de la tradition que l'on veut suivre. Un mauvais choix de technique peut entraîner des problèmes majeurs d'adaptations. Encore une fois, il est très important d'obtenir une grande expertise dans les différentes techniques possibles afin de décider lesquelles devraient être utilisées dans le cadre de ce projet.

Statistiques non révélatrices. Les procédures statistiques seront utilisées de façon exploratoire et pour obtenir certaines confirmations. Les résultats ne seront cependant pas significatifs si le codage initial des données (verbalisations) est erroné. Encore une fois, le succès de la recherche dépend de la précision du schéma de codage développé. Ce schéma doit contenir les catégories nécessaires à un codage significatif sans être trop compliqué pour les codeurs. De plus, ce schéma doit être suffisamment simple pour éviter toute ambiguïté.

## 4.2 Représentation formelle des données

L'analyse d'un protocole, où les verbalisations de chaque participant sont enregistrées, nécessite un outil qui représente formellement les données brutes. Cette première étape du processus d'analyse est sûrement la plus importante. Son objectif est de représenter les données empiriques sous une forme structurée et formelle qui favorise l'énoncé de

résultats. Les représentations utilisées lors de recherches similaires sont souvent ad hoc et empiriques, ce qui rend l'interprétation et la comparaison des résultats difficiles. Un schéma de codage est en fait l'élément de base sur lequel dépend tout le reste de la méthode d'analyse. Le schéma de codage doit permettre de formaliser les concepts représentant les données dans un format ou syntaxe qui facilite leur utilisation. Le formalisme employé doit améliorer l'efficacité et le niveau de confiance des mesures futures.

### 4.2.1 Fondement théorique du schéma de codage

Bien que les études empiriques présentées au chapitre 1 utilisent plusieurs méthodes intéressantes pour représenter les comportements, elles sont insuffisantes dans l'étude des réunions de révision technique. Il appert cependant, à la lumière des résultats obtenus par les recherches précédentes, que deux concepts sont nécessaires pour bâtir un schéma de codage: (1) un concept permettant de décrire les données brutes et (2) un concept qui facilite la représentation de cette description. Le schéma de codage présenté dans ce chapitre se base sur les principes d'études des interactions verbales développés en linguistique pour décrire les données brutes et le paradigme orienté-objet est utilisé pour doter le schéma d'une structure d'ordonnancement des descriptions.

### 4.2.2 Théorie de la linguistique

La réunion de révision technique est un type particulier de réunion de travail, elle peut

donc constituer un type particulier d'interaction verbale. Une conversation est considérée comme l'un des meilleurs environnements pour l'étude des conditions intrinsèques de déroulement des interactions parce qu'elle est moins influencée par des contraintes extérieures (Bange, 1992). Cinq unités (Figure 4.5), deux unités monologales et trois unités dialogales, sont considérées comme les plus pertinentes pour la description de l'organisation des conversations (Roulet et al., 1987).

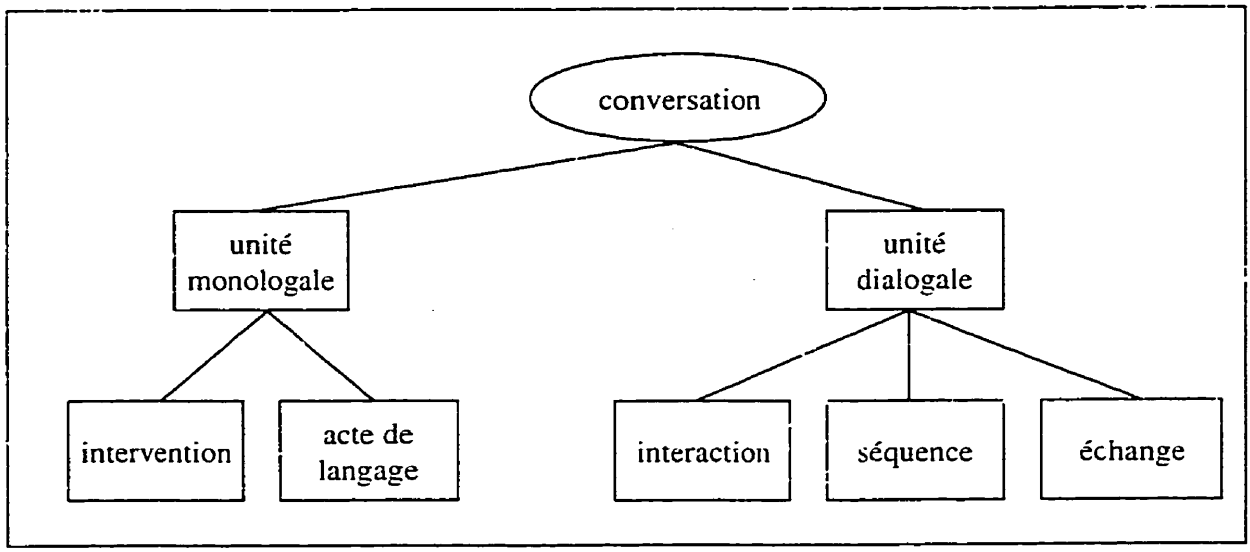



**Figure 4.5 Identification des unités d'une conversation**

Kerbrat-Orrecchioni (1998) définit les unités de la Figure 4.5 comme suit:

1. Acte de langage: Acte (question, requête, promesse, excuse, etc.) tel que décrit en pragmatique avec tout de même quelques aménagements visant à rendre la notion mieux adaptée aux réalités conversationnelles.

2. Intervention: Unité produite par un seul et même locuteur: c'est la contribution d'un locuteur particulier à un échange particulier.

3. Échange: La plus petite unité dialogale (c'est-à-dire construite par deux participants au moins).

4. Séquence: Un bloc d'échanges reliés par un fort degré de cohérence sémantique ou pragmatique, c'est-à-dire traitant d'un même thème ou centré sur une même tâche.

5. Interaction: Unité communicative qui présente une évidente continuité interne, alors qu'elle rompt avec ce qui précède et la suit.

Les actes de langage se combinent pour constituer des interventions, actes et interventions étant produits par un seul et même locuteur. Dès que deux locuteurs au moins interviennent, on a affaire à un échange. Les échanges se combinent pour constituer des séquences, lesquelles se combinent pour constituer les interactions, unités maximales de l'analyse.

### 4.2.3 Paradigme orienté-objet

Le défi principal lors de l'étude des activités cognitives et coopératives est la difficulté de déterminer une structure de représentation des données. L'élaboration d'un tel modèle est habituellement laissée à la discrétion du chercheur et il est souvent difficile de réutiliser et valider ce modèle dans une autre étude indépendante. La structure doit être capable de donner de la flexibilité et de la précision au chercheur. Elle doit isoler les aspects importants requis par le chercheur du reste de son environnement et permettre des niveaux de raffinements de description si jugé nécessaire. La structure du schéma doit faciliter la représentation des interactions potentielles entre les descriptions.

Le paradigme orienté-objet facilite l'abstraction des éléments importants d'un contexte et permet de les représenter sous une forme facile à comprendre. Une structure élaborée à l'aide du paradigme OO identifie les besoins du chercheur sous la forme d'objets et d'attributs trouvés dans le domaine d'application. Cette approche amène plusieurs bénéfices au chercheur: rationnel documenté, notation normalisée et une sémantique de modélisation définie, qui facilitent l'interprétation et généralisent son application.

#### 4.2.3.1 Abstraction et Encapsulation

La représentation d'une intervention est une abstraction de la réalité correspondant aux besoins et aux questions du chercheur. L'abstraction est la sélection de certaines caractéristiques essentielles d'un événement qui le distinguent des autres et qui définissent ses limites conceptuelles. Il est certain qu'une abstraction est fondamentalement incomplète et imprécise parce que toute description de la réalité est un raccourci. Son utilité n'est cependant pas remise en question. La difficulté majeure dans la conception d'une structure est de correctement identifier les abstractions nécessaires à la représentation d'une situation.

L'encapsulation regroupe les détails spécifiques d'une situation dans une seule entité et masque ces détails au monde extérieur. L'abstraction détermine l'implantation d'une entité, ne montrant que les caractéristiques utiles au problème tandis que l'encapsulation cache le fonctionnement interne d'une entité. Ces concepts sont très utiles dans la

conception d'une structure de codage puisqu'ils assurent que chacune des classes est exhaustive et exclusive.

4.2.3.2 Hiérarchie et Modularité

La structure doit être capable de représenter la réalité de la façon la plus réaliste et précise possible. L'implantation de la structure doit aussi tenir compte et inclure les objectifs du chercheur. L'instauration d'une hiérarchie dans la structure permet d'ordonner les abstractions selon les besoins du chercheur. La modularité de la structure assure que la situation représentée est décomposée dans un ensemble de classes associées ensemble. La structure est complétée à l'aide d'associations qui représentent les liens conceptuels entre les différentes parties.

4.2.3.3 Persistance

La persistance d'un objet est un concept fondamental dans la conception de ce schéma de codage. La persistance signifie que l'existence d'un objet transcende le temps, c'est-à-dire que l'objet continue d'exister longtemps après que son créateur soit disparu.

Une des principales caractéristiques des interventions observées lors de l'analyse des conversations est la mémoire. Les participants à la réunion n'interviennent pas de façon aléatoire dans la réunion mais plutôt en se synchronisant avec la discussion en cours. Le concept fondamental de ce schéma de codage est basé sur cette hypothèse parce qu'il représente le sujet d'une intervention comme l'objet d'une intervention faite au

préalable. Un participant réagit à l'intervention d'un autre participant tout de suite, après un certain temps ou peut-être jamais. Toutes les interventions doivent donc persister dans le temps. La représentation de ce phénomène est naturelle et facilitée par le paradigme OO.

La notation UML (Unified Modeling Language) est utilisée pour illustrer le schéma de codage. Le formalisme obtenu par l'utilisation de l'OO permet de former une base solide à partir de laquelle des mesures et conclusions peuvent être ultérieurement tirées.

### 4.2.4 Représentation formelle des données

La représentation formelle utilisée dans le cadre de ce projet est basée sur les concepts de la linguistique et du paradigme orienté-objet. Le niveau atomique de la représentation est l'intervention parce que l'on veut être en mesure de caractériser le travail coopératif des individus et non leurs cheminements cognitifs individuels. Le regroupement des interventions en séquences est par la suite déduit formellement à partir de cette représentation. L'utilisation d'une structure représentative solide et formelle comme base permet de conserver le formalisme dans les analyses subséquentes. Cette section résume les deux représentations formelles nécessaires pour modéliser les conversations à partir d'un protocole brut d'une réunion de révision technique du processus de génie logiciel.

#### 4.2.4.1 Représentation des interventions: schéma de codage

Un participant contribue à une conversation par le biais d'interventions uniques. Ces

interventions constituent les éléments de base à identifier et à caractériser lors d'une réunion de révision technique. La conversation est décomposée en interventions selon les règles suivantes:

1. une nouvelle intervention commence lorsque l'activité incluse dans une intervention ne peut plus être décrite par la même catégorie ; ou

2. l'auteur de l'intervention change.

La nature particulière du travail effectué lors d'une réunion de révision technique limite les types d'interventions possibles. L'identification des interventions appropriées pour décrire une réunion de révision technique est faite de façon empirique. Cette classification est l'opération critique de toute l'approche. Il existe quatre groupes d'interventions lors d'une réunion de révision technique (Tableau 4.1).

**Tableau 4.1 Groupes d'interventions**

| GROUPE | ABRÉ. | DESCRIPTION |
|---|---|---|
| Gestion | GES | Planification et coordination de tâches au niveau du projet (planification de la prochaine réunion de révision) ou de la réunion (modération de la réunion). |
| Introduction | INTRO | Introduction d'un nouvel artefact dans la conversation. L'introduction peut être implicite lorsqu'un nouveau sujet de conversation est utilisé. |
| Demande | DEM | Requête d'un comportement de révision précis |
| Présentation | PRES | Élaboration d'un comportement de révision précis. |

Une intervention de gestion n'est pas spécifique à la réunion de révision technique puisqu'il s'agit de comportements reliés à la modération de la réunion ou à la liaison de la réunion avec le projet lui-même. L'intervention d'introduction identifie le changement de sujet de conversation et ce changement de sujet s'effectue lorsque les participants passent à une nouvelle partie de l'artefact révisé. Les deux derniers groupes d'interventions (demande et présentation) regroupent la majeure partie des interventions d'une réunion de révision technique. Lors de la révision d'un artefact, un participant demande ou présente un comportement de révision précis. Ces comportements sont décrits en détails plus loin. Dans le cas où une intervention ne peut être classée dans un des quatre groupes, elle est considérée comme "autre". Ces interventions sont de deux types: les interventions incompréhensibles parce que trop courtes et les écoutes actives. Une écoute active est utilisée par les participants lorsqu'ils signifient qu'ils comprennent ce qui se dit. Une écoute active est habituellement caractérisée par un son tel que « henhen... » ou simplement une répétition de l'intervention précédente. Toutes ces interventions ajoutent du bruit à l'analyse mais ont un impact négligeable sur la nature intrinsèque des conversations.

Le Tableau 4.2 indique la liste des comportements considérés les plus appropriés pour la description des réunions de révision technique. Trois attributs sont nécessaires pour décrire un comportement spécifique, faciliter son identification et former une base cohérente d'analyse:

1. ACTIVITÉ: identification de l'action effectuée par le locuteur,

2. SUJET: définition de l'entité sur laquelle l'activité (attribut 1) est effectuée et,

3. CRITÈRE: attribut qui complète le sujet (attribut 2) en spécifiant s'il s'agit de forme ou de contenu. Cet attribut est par défaut le contenu du sujet.

Une activité est toujours effectuée sur un sujet. Le sujet est le lien qui maintient la cohésion dans une conversation.

**Tableau 4.2 Types de comportements lors des réunions de révision technique**

| COMPORTEMENT | ABRÉ. | DÉFINITION |
|---|---|---|
| Évaluation | EVA | Action de porter un jugement sur la valeur d'un sujet. Cette évaluation peut être négative ou positive. |
| Justification | JUS | Une justification est l'action d'argumenter ou de prouver le bien-fondé du POURQUOI on a fait un certain choix. Il est souvent nécessaire de répondre à des évaluations par une justification de l'approche prise. |
| Rejet | REJ | Refuser ou écarter un sujet comme valide. |
| Acceptation | ACC | Action de considérer un sujet comme valide. |
| Information | INFO | Action de porter des renseignements à la connaissance d'une ou de plusieurs personnes au sujet de la nature d'un sujet quelconque. |
| Hypothèse | HYP | Représentation qu'un participant se fait d'un sujet quelconque. Cette représentation a le statut cognitif d'hypothèse ce qui s'exprime verbalement par des expressions comme « je crois que », « je pense que... », « ...peut-être ». |
| Développement | DEV | Action d'exposer en détail une idée nouvelle. Il s'agit donc d'une activité créatrice. |

Il y a deux types de critère: forme et contenu. Les critères de forme sont utilisés lorsque les participants discutent du format de la solution plutôt que sa pertinence technique (critère de contenu). Les critères de contenu sont basés sur la norme ISO 9126 tandis que les critères de forme proviennent de la méthodologie utilisée par l'équipe de développement. Le Tableau 4.3 contient la liste des critères utilisés.

**Tableau 4.3 Critères de forme et de contenu**

| Critères de forme | Critères de contenu |
|---|---|
| a. Nomenclature | (ISO 9126) |
| b. Algorithmes | a. Fonctionnalité |
| c. Fonctions | b. Réutilisation |
| d. Fichiers | c. Portabilité |
| e. Types de données | d. Fiabilité |
| f. Éditeur | e. Maintenabilité |
| g. Déclaration des variables | f. Efficacité |
| h. Variables globales | g. Facilité d'implantation |
| i. Structure du document | |
| j. Sémantique | |
| k. Niveau de description | |

L'objectif d'une réunion de révision technique est de réviser un artefact (document, code, etc.). Cet artefact est divisé en sections qui deviennent les sujets des interventions d'introduction. Un critère peut aussi devenir le sujet de conversation si les participants élaborent sur sa nature mais ils sont surtout utilisés pour décrire plus en détail le sujet d'une intervention.

Une intervention introduit un nouveau sujet de conversation ou effectue une activité sur le sujet courant. Il existe une certaine continuité lors d'une conversation. Les participants introduisent rarement des sujets de façon aléatoire, mais plutôt en relation avec la conversation courante. Il est donc probable qu'une intervention devienne le sujet d'une

intervention ultérieure. Une phrase comme « Je suis d'accord avec ce que tu as dit... » est un très bon exemple où le sujet de l'activité acceptation est une intervention antérieure (« ...ce que tu as dit... »). La modélisation de ce phénomène nécessite que les interventions persistent dans le temps afin qu'un participant puisse y réagir.

Le schéma de codage est une représentation conceptuelle de l'association des différentes composantes de la description des interventions ainsi que le concept de la persistance. Le paradigme orienté-objet permet de formaliser le schéma de codage à l'aide de classes, attributs, opérations et associations. L'OO facilite l'interprétation et l'application du schéma.

Dans la Figure 4.6, le nom de la classe indique le type d'intervention ou de comportement, l'attribut d'identité permet d'identifier une instance de la classe et l'opération fournit l'activité précise associée avec un type d'intervention. D'autres attributs sont ajoutés selon les besoins, comme un identificateur du critère et un identificateur du sujet. Les interventions sont regroupées en quatre groupes distincts selon leurs natures et les objectifs de la recherche.

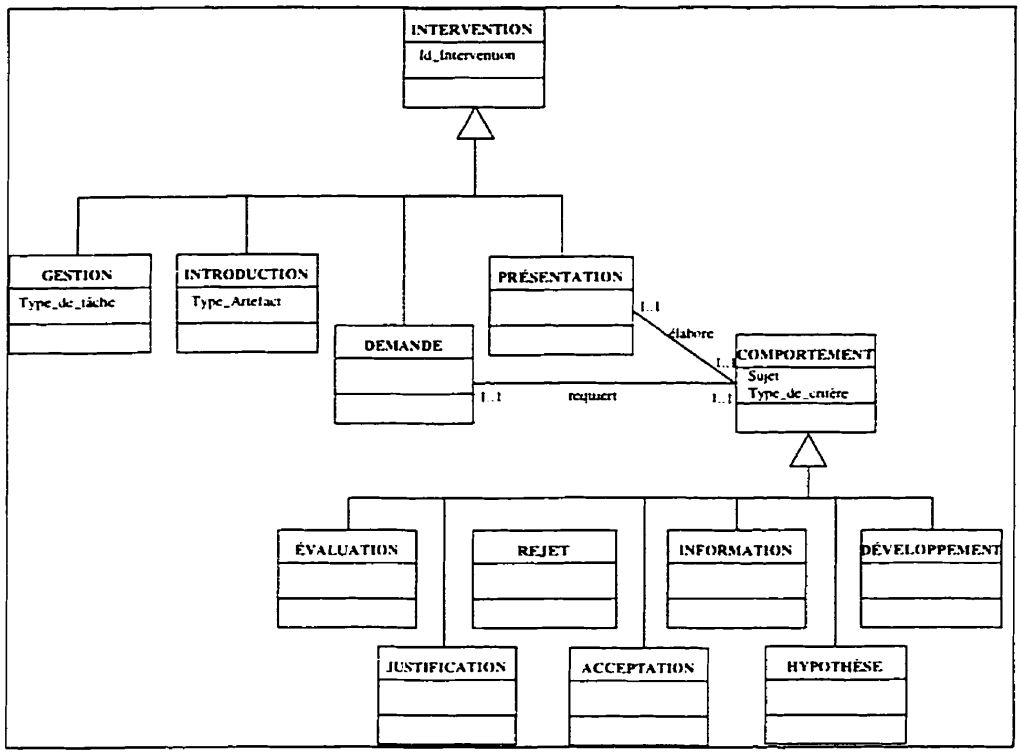



**Figure 4.6 Diagramme de classes du schéma de codage**

Les associations lient certains groupes d'interventions aux comportements. Les associations REQUIERT et ÉLABORE lient les classes DEMANDE et PRÉSENTATION avec un comportement. Ces associations représentent la description détaillée du comportement de l'intervention.

Une intervention sera codée en utilisant une structure syntactique très précise composée des composantes du schéma. Par exemple, lors de la centième (ID) intervention, le participant M (ID) présente (GROUPE) l'évaluation (ACTIVITÉ) de la version x du document de design (SUJET) en référant à la forme (CRITÈRE) utilisée pour décrire le design. Les attributs d'activité et de sujet forment le coeur de la description des

comportements.

La syntaxe élémentaire, composée d'un symbole de séparation (/) entre chacune des composantes, est utilisée pour coder les interventions de façon à faciliter les analyses automatiques. Le Tableau 4.4 montre la syntaxe utilisée (Robillard et als, 1998) pour coder les groupes d'interventions.

**Tableau 4.4 Syntaxe de codage des interventions**

| GROUPE | SYNTAXE |
|---|---|
| GESTION | ID/GES/TÂCHE |
| INTRODUCTION | ID/INTRO/ARTEFACT |
| DEMANDE | ID/DEM/ACTIVITÉ/SUJET/CRITÈRE |
| PRÉSENTATION | ID/*PRES*/ACTIVITÉ/SUJET/CRITÈRE |

Afin de réduire la taille des codes et étant donné que le groupe présentation représente la majorité des interventions d'une réunion de révision technique, l'abréviation PRES n'est pas utilisée lors du codage et doit être considérée comme implicite. Le critère est par défaut le contenu.

La Figure 4.7 est un exemple de codage de trois interventions. Chaque intervention est identifiée par le locuteur et un nombre unique (INTERV). Dans cet exemple, l'artefact SOLn, représentant une section précise du document révisé, est introduit lors de l'intervention x, générant ainsi l'intervention INTROx. Cette intervention devient ultérieurement le sujet de l'évaluation selon un critère de contenu (fonctionnalité) lors de

l'intervention x+1, dont le résultat est l'intervention EVAx+1. L'intervention suivante montre la justification d'EVAx+1 selon un critère de forme (structure des algorithmes). La colonne syntaxe montre la notation utilisée pour représenter le modèle conceptuel.

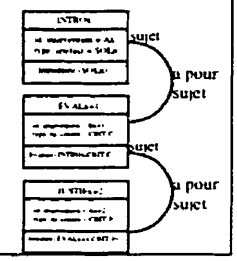
| LOC | INTERV | SYNTAXE | CONTENU DE L'INTERVENTION | REPRÉSENTATION CONCEPTUELLE |
|---|---|---|---|---|
| A | x | Ax/INTRO/SOLn | Introduction dans la discussion de l'élément SOLn |  |
| B | x+1 | Bx+1/EVA/INTROx/CRIT.C | Évaluation de l'introduction x par rapport à un critère de contenu | |
| A | x+2 | Ax+2/JUS/EVAx+1/CRIT.F | Justification de l'évaluation x à l'aide d'un critère de forme | |



**Figure 4.7 Exemple d'utilisation du modèle**

Afin de mieux illustrer ces explications et celles qui vont suivre, un court échantillon, tiré d'un des protocoles de la recherche (Annexe A), sera utilisé à titre d'exemple. Il s'agit d'une conversation entre les participants à une réunion de révision d'un document de design. Ils y discutent les mérites de deux aspects importants du design du module en question dans ce document. Les trois premières colonnes représentent le locuteur de l'intervention, un identificateur unique de l'intervention et le contenu de l'intervention.

L'Échantillon 1 contient le codage des interventions d'une réunion de révision technique. Un code tel que « SOLed » désigne une section de l'artefact révisé. SOLed et SOLee sont deux parties mutuellement exclusives du document révisé. Les critères sont représentés par CRIT.C (critère de contenu) et CRIT.F (critère de forme). Une lettre en minuscule est ajoutée pour différencier les critères entre eux. L'intervention 51 est une introduction

implicite de SOLed dans la conversation.

| Loc | ID | Intervention | Syntaxe |
|---|---|---|---|
| | 51 | | --51/INTRO/SOLed |
| B | 52 | Pouquoi tu l'as mis là ma cent cinquante. | B52/DEM/JUS/INTRO51 |
| M | 54 | C'est parce que j'crois pas plus au eh, defined. le cent cinquante defined. Ça va revenir au même, mais ça au moins c'est parce que le compilateur va les checker. | M54/JUS/INTRO51/CRIT.Ca |
| C | 57 | Des message d'erreurs, y va peut-être en avoir cinquante différents t'sais/ | C57/HYP/INTRO51 |
| M | 60 | Ah, non, mais ça c'est juste un type, comme le type du message comme tel. | M60/REJ/HYP57/CRIT.Ca |
| C | 61 | Euh, euh. | C61/ACC/REJ60 |
| M | 62 | Ça c'est juste parce que ça me prend, ça me prend des, des champs, ok, les quatres qui sont là, là, c'est parce que ça me prend des chaînes qui sont déjà fixées au départ pour, pour eh, qui sont comme, fixes au message. | M62/INTRO/SOLee |
| M | 63 | Fait que ceux-là je les fixe, j'mets les quatres, j'mets comme les quatres premiers, les messages qu'on va ajouter, ça va être, ça va être les suivants.Fait que là on va en mettre eh, on mettra ce qu'on voudra là, un message erreur, mémoire insuffisante | M63/JUS/INTRO62/CRIT.Ca |
| M | 64 | Ah, mais comme le temps passe vite, c'est amusant non. | M64/INFO/INTRO62/CRIT.Ca |
| B | 65 | Pourquoi faire dans le fond, si on pourrait les utiliser quand même. | B65/DEM/INFO/INTRO62 |
| B | 67 | Ouais, ok, ouais de toute façon on n'a pas le choix. | B67/ACC/INFO65/CRIT.Ca |
| M | 68 | On peux en faire là. | M68/INFO/INTRO62 |

**Échantillon 1 Exemple d'utilisation du schéma de codage**

4.2.4.2 Validation du schéma de codage

Étant donné l'importance du schéma de codage dans l'approche globale d'analyse des réunions de révision technique, il est nécessaire de pouvoir juger jusqu'à quel point ce schéma est approprié pour représenter les interventions de telles réunions. Les données utilisées pour valider le schéma proviennent de sept réunions de révision technique effectuées dans le cadre d'un projet de développement d'une durée de 19 semaines.

Quatre ingénieurs en logiciel ont participé au projet et ils ont tous été présents lors des réunions analysées.

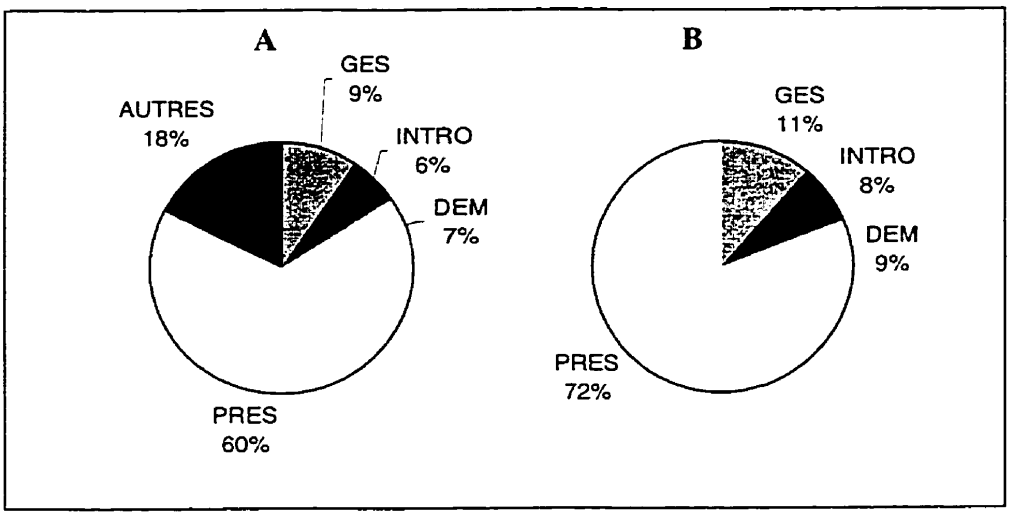



**Figure 4.8 Distribution des interventions dans les groupes**

Le schéma de codage permet de représenter 82% (Figure 4.8A)de toutes les interventions survenues lors des réunions de révision technique observées. Si l'on ne tient pas compte du 18% des autres interventions, on distingue que 81% (PRES, 72% et DEM, 9%) des interventions requièrent l'utilisation d'un comportement pour la décrire plus spécifiquement.

La distribution en fréquence permet de constater l'importance relative des interventions codées et est un indicateur de l'aptitude des classes du schéma à bien représenter les données. Par exemple, si un type d'intervention est dominant, c'est-à-dire que la plupart des interventions sont codées dans cette classe, alors on peut affirmer que le schéma de codage n'est pas très sensible à ce type d'interventions.

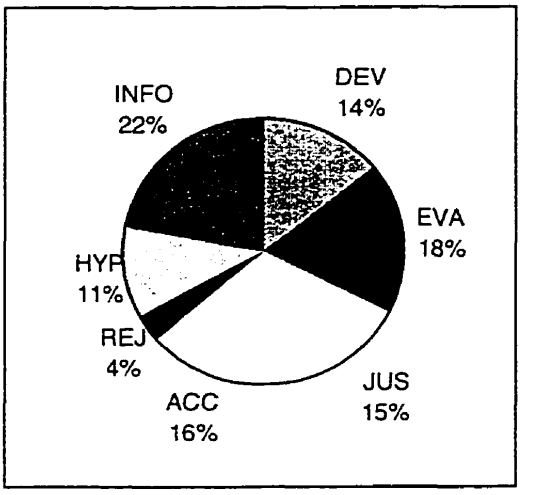



**Figure 4.9 Distribution des comportements du schéma de codage**

La Figure 4.9 montre la distribution relative des 7 comportements du schéma de codage à l'intérieur de DEMANDE et PRÉSENTATION. On remarque que la distribution des classes est relativement équivalente. Ce résultat permet de valider jusqu'à quel point le schéma de codage est sensible et approprié aux données.

Le schéma de codage peut aussi contenir des classes dont leurs significations sont ambiguës ou non orthogonales. Cela augmente la difficulté pour le codeur de choisir la classe appropriée. Des mesures statistiques peuvent être utilisées pour vérifier la cohérence du schéma de codage (Simon Labelle, à publier).

4.2.4.3 Représentation des séquences

Une séquence est une série d'interventions successives liées par leurs sujets. Différents types d'échanges surviennent lors d'une séquence et leurs relations définissent l'empan de la séquence. Un échange primaire, souvent représenté par l'introduction d'un nouveau

sujet, constitue la base d'une séquence. Une séquence survient lorsque les échanges suivants élaborent sur le sujet de cet échange primaire ou sur des sujets provenant d'élaborations elles-mêmes effectuées sur cet échange primaire. Les interventions d'introduction et de gestion délimitent les séquences. Une séquence de révision commence toujours avec l'introduction de l'artefact révisé et dure jusqu'à ce qu'il y ait l'introduction d'un nouvel artefact ou la discussion d'une tâche de gestion.

| Loc | ID | Intervention | Syntaxe | Sujet | Séquence |
|---|---|---|---|---|---|
| | 51 | | --51/INTRO/SOLed | SOLed | |
| B | 52 | Pouquoi tu l'as mis là ma cent cinquante. | B52/DEM/JUS/INTRO51 | INTRO51 | |
| M | 54 | C'est parce que j'crois pas plus au eh, defined, le cent cinquante defined. Ça va revenir au même, mais ça au moins c'est parce que le compilateur va les checker. | M54/JUS/INTRO51/CRIT.Ca | INTRO51 | Séquence 1 |
| C | 57 | Des message d'erreurs, y va peut-être en avoir cinquante différents t'sais/ | C57/HYP/INTRO51 | INTRO51 | |
| M | 60 | Ah, non, mais ça c'est juste un type, comme le type du message comme tel. | M60/REJ/HYP57/CRIT.Ca | HYP57 | |
| C | 61 | Euh, euh. | C61/ACC/REJ60 | REJ60 | |
| M | 62 | Ça c'est juste parce que ça me prend, ça me prend des, des champs, ok, les quatres qui sont là, là, c'est parce que ça me prend des chaînes qui sont déjà fixées au départ pour, pour eh, qui sont comme, fixes au message. | M62/INTRO/SOLee | SOLee | |
| M | 63 | Fait que ceux-là je les fixe, j'mets les quatres, j'mets comme les quatres premiers, les messages qu'on va ajouter, ça va être, ça va être les suivants.Fait que là on va en mettre eh, on mettra ce qu'on voudra là, un message erreur, mémoire insuffisante | M63/JUS/INTRO62/CRIT.Ca | INTRO62 | |
| M | 64 | Ah, mais comme le temps passe vite, c'est amusant non. | M64/INFO/INTRO62/CRIT.Ca | INTRO62 | Séquence 2 |
| B | 65 | Pourquoi faire dans le fond, si on pourrait les utiliser quand même. | B65/DEM/INFO/INTRO62 | INTRO62 | |
| B | 67 | Ouais, ok, ouais de toute façon on n'a pas le choix. | B67/ACC/INFO65/CRIT.Ca | INFO65 | |
| M | 68 | On peux en faire là. | M68/INFO/INTRO62 | INTRO62 | |

**Échantillon 2 Identification des séquences**

L'Échantillon 2 présente l'identification des séquences pour l'échantillon d'une réunion de révision technique. La première séquence réfère à SOLed, introduite implicitement dans l'intervention 51. Dans le premier échange, B et M justifient la fonctionnalité de SOLed.

La participant C fait alors une hypothèse sur la nature de SOLed. Le rejet de l'hypothèse qui s'ensuit permet à C de mieux comprendre la situation (son acceptation).

La méthode de représentation formelle des données présentée dans ce chapitre est le résultat d'une démarche évolutive et itérative qui a permis de faire ressortir les détails importants retrouvés lors d'une réunion de révision technique. Le présent chapitre présente la dernière version du schéma de codage et donc celle utilisée pour recueillir les données et effectuer les analyses présentées dans le prochain chapitre.

L'identification des interventions et des séquences(Figure 4.10) permet de moduler la réunion en unités conceptuelles représentant le contexte de la réunion (interaction).

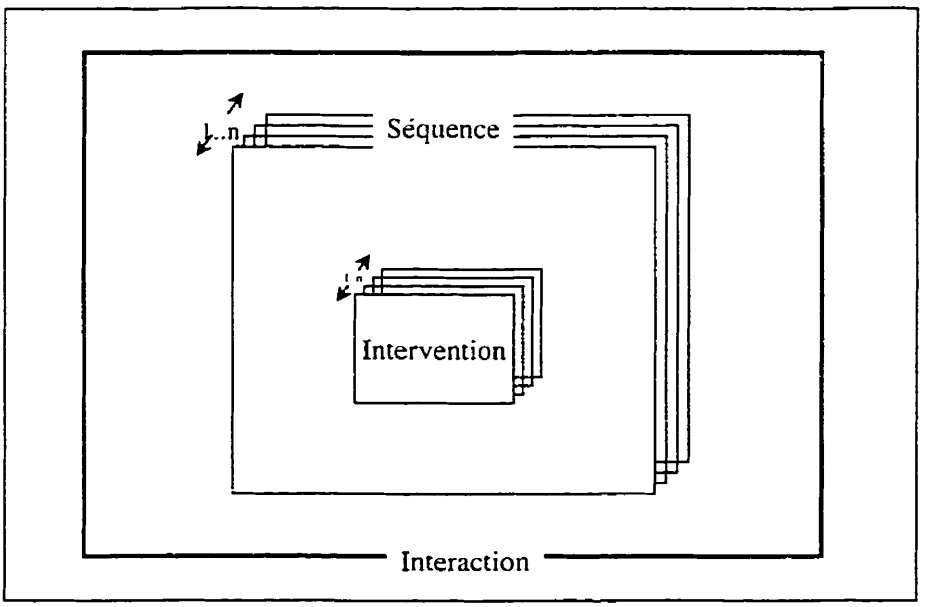



**Figure 4.10 Unités d'analyses à l'intérieur d'un réunion de révision technique**

Ces unités facilitent l'analyse rigoureuse des données en distinguant leur structure logique

intrinsèque. Le chapitre suivant proposera plusieurs perspectives d'analyses possibles à partir de la représentation détaillée dans ce chapitre.

# CHAPITRE V

# ÉTUDE DES RÉUNIONS DE RÉVISION TECHNIQUE

L'objectif du schéma de codage est de représenter les réunions de révision technique dans une forme qui facilite leurs analyses. Sept réunions ont été analysées dans le cadre de cette recherche. Ces réunions ont été effectuées dans le cadre d'un projet de développement logiciel à vocation industrielle. Le processus logiciel utilisé lors de ce projet était structuré et documenté. Il est évalué au niveau CMM-2 (Paulk et als., 1993). Ces sept réunions constituent un ensemble suffisant de données puisqu'elles sont une représentation fidèle du travail de révision effectué par les membres de l'équipe et parce qu'elles contiennent suffisamment de données pour obtenir des résultats statistiques significatifs (la mesure $10R^2$ de Fagen et Young).

Le présent chapitre présente en détails trois types d'analyses pouvant être effectuées à partir du codage des réunions: une analyse statique des interventions (page 115) afin de déterminer la nature des discussions, une analyse de l'influence du rôle des participants sur l'occurrence des interventions (page 128) et une analyse des patrons de discussions afin de déterminer le cheminement des discussions au cours d'une réunion (page 145). Un des avantages du schéma de codage et de la syntaxe utilisée pour le représenter est l'utilisation d'outils automatiques d'analyses. L'analyse statique des interventions a été

réalisée grâce au progiciel *Excel*, l'analyse de l'influence des participants à l'aide du progiciel *Statistica* et l'analyse des patrons de discussions à l'aide de logiciels développés à l'interne. Cela est un avantage indéniable puisque l'ajout de biais dans les analyses est réduite au codage seulement et non pas dans les étapes d'analyses suivantes.

## 5.1 Analyse statique de la nature des interventions

L'analyse statique consiste à identifier et à mesurer les interventions qui caractérisent les réunions de révision technique. La distribution en fréquence démontre l'importance relative des types d'interventions dans certaines situations et permet de mieux comprendre la nature de la réunion elle-même. L'analyse statique est une première étape qui permet de répondre à plusieurs questions et qui oblige aussi à se poser des questions qui seront répondus par d'autres types d'analyses.

### 5.1.1 Mesure de l'importance relative des interventions

Une intervention regroupe les verbalisations de même nature exprimées par un participant à un moment donné de la réunion. Une intervention a donc une durée variable. Par exemple, une intervention d'acceptation aura probablement une durée beaucoup plus courte qu'une intervention d'information. Bien qu'en tant qu'interventions, elles soient toutes deux équivalentes, leur importance relative dans la durée de la réunion diffère. On peut calculer la durée d'une intervention de deux façons: le temps et le nombre de mots exprimés.

Le temps est une mesure absolue mais qui est caractérisée par l'incertitude de l'instrument de mesure utilisé (chronomètre) et le mesureur lui-même. Le temps permet de savoir la durée moyenne des interventions en secondes. L'intervention est donc composée de deux types d'éléments: les mots et les temps morts. Il arrive en effet souvent qu'il y ait un temps mort entre deux verbalisations. Lors de la mesure du temps, on doit décider à quelle intervention ce temps mort doit être ajouté. Ce genre de décision ajoute donc un certain niveau d'incertitude. De plus, un participant commence souvent son intervention avant qu'un autre ait complété la sienne. Cet élément ajoute à l'incertitude lors du calcul de la durée d'une intervention. .

Le calcul des mots exprimés durant une intervention réduit l'incertitude inhérente à la mesure du temps. Par exemple, une intervention d'acceptation nécessite peut-être beaucoup moins de mots qu'une intervention d'information. De plus, on peut faire l'hypothèse que **si le débit de l'être humain se situe entre 120 et 150 mots/minutes, il devrait y avoir une forte corrélation entre la durée d'une intervention et son nombre de mots**. La durée des interventions a été mesurée lors d'une réunion de révision technique afin de comparer la mesure du temps au nombre de mots. La Figure 5.1 indique la durée moyenne des interventions (développement, évaluation, justification, acceptation, rejet, hypothèse et information) en fonction du temps et du nombre de mots.

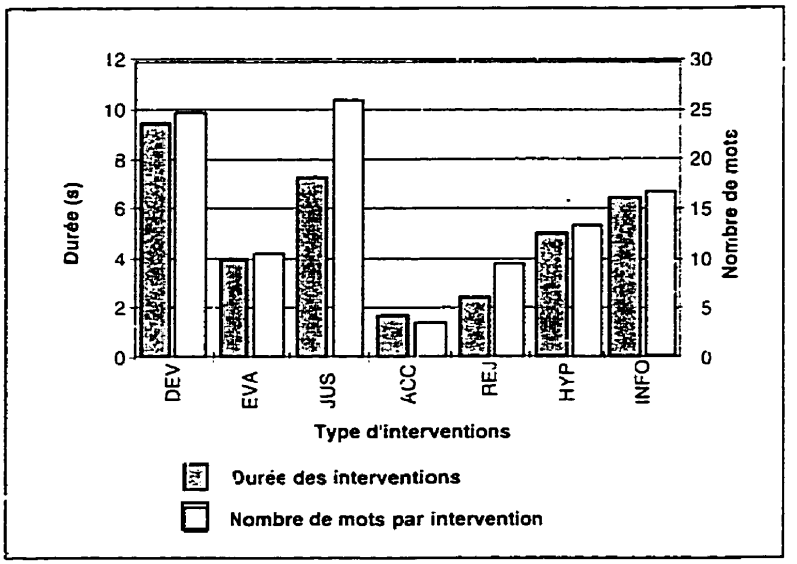



**Figure 5.1 Comparaison entre le temps et le nombre de mots**

Le taux de corrélation entre le nombre de mots et la durée des interventions a été calculé à 95%. On observe d'ailleurs que les deux graphiques sont en majeure partie similaires, ce qui tend à prouver notre hypothèse initiale. Par exemple, les interventions d'*évaluation* durent en moyenne 4 secondes (plus ou moins une certaine incertitude) pour un nombre moyen de mots environ égal à 10. Nous utiliserons donc le nombre de mots dans le reste de ce document pour discuter de l'importance des interventions dans la réunion.

### 5.1.2 Nature des interventions

Un premier regard sur les réunions analysées permet de se rendre compte que la majorité des interventions sont très courtes (Figure 5.2). Les interventions de 1 mot sont les plus fréquentes (204 interventions) tandis que la plus longue intervention observée contenait 135 mots.

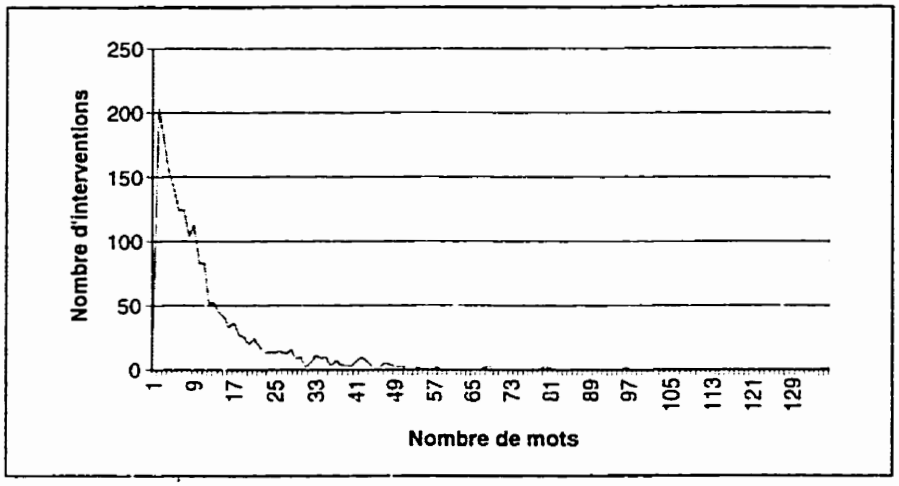



**Figure 5.2 Courbe du nombre d'interventions en fonction du nombre de mots**

En fait, du total des 1954 interventions analysées, 60% (1181) ont moins de dix mots et 87% (1701) ont moins de vingt mots. La révision d'un artefact est donc effectué par de courtes interventions rapides qui expriment l'activité cognitive d'un participant efficacement. Une réunion de révision technique est le cadre d'un travail coopératif important puisque les participants tour à tour échangent systématiquement leurs idées.

La Figure 5.3 indique la distribution des interventions et des mots par groupe (gestion, introduction, demande, présentation et autres). Cette distribution est basée sur le regroupement des interventions des sept réunions analysées. La figure démontre bien que toutes les réunions analysées (RE02-RE10c) ont sensiblement la même distribution des interventions par groupes. En se basant sur ce résultat, on peut déduire que les sept réunions observées ont des contextes semblables, ce qui nous permet de les regrouper en un seul ensemble lors de nos analyses. Bien que le schéma de codage ne puisse représenter 18% des interventions (autres), celles-ci ne représentent que 6% du propos

(mots) tenu lors des réunions, une proportion qui a été jugée fort acceptable.

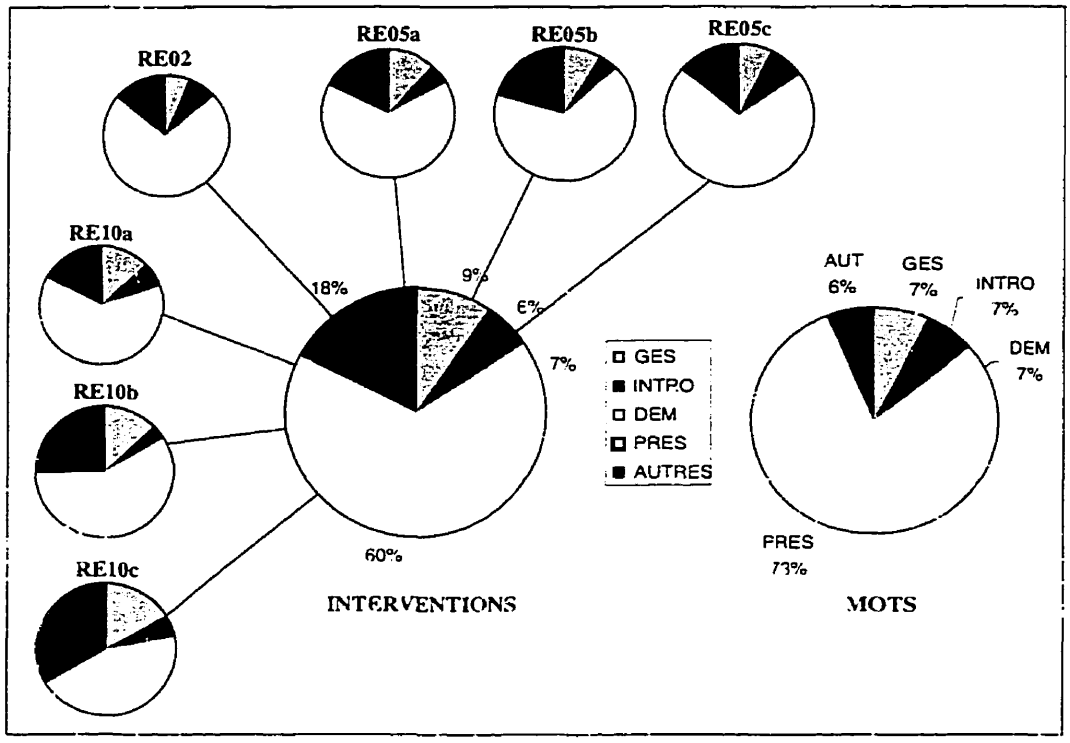



**Figure 5.3 Distribution des groupes**

On dénombre 1425 des 1954 (73%) interventions (introduction, demande et présentation) appartenant aux séquences de révisions d'un artefact (les autres étant de la gestion ou autres). Au total, 127 séquences de révision sont donc à l'étude pour une moyenne de 11 interventions par séquence. La plus longue séquence contient 53 interventions tandis que les plus courtes ne contiennent que l'introduction d'un artefact. Ces dernières sont considérées comme des révisions implicites de l'artefact puisqu'aucun participant n'avait de points à discuter. Ces 127 séquences représentent 87% du propos (mots) tenu lors des réunions et seront le sujet de nos analyses.

La Figure 5.4 indique la distribution des interventions par groupe à l'intérieur des séquences de révision. Seulement trois groupes (introduction, demande, présentation) y sont représentés. On voit que les demandes et les présentations occupent la majorité (91%) de ces séquences et plus de 92% du propos (mots). Ce résultat démontre clairement que la catégorisation par groupes n'est pas assez précise et que l'utilisation des comportements pour décrire ces 127 séquences sera plus utile. En fonction de ce résultat, il a été décidé de ne pas différencier ces deux groupes (présentation et demande) lors des analyses qui suivent et de plutôt se concentrer sur le comportement d'un participant lors d'une intervention. Ce genre d'analyse se concentre donc surtout sur les aspects fonctionnels d'une conversation (nature de la conversation) et non sur la nature de l'interaction entre les individus (requêtes vs réponses). L'annexe C contient un échantillon des interventions appartenant aux séquences de révision.

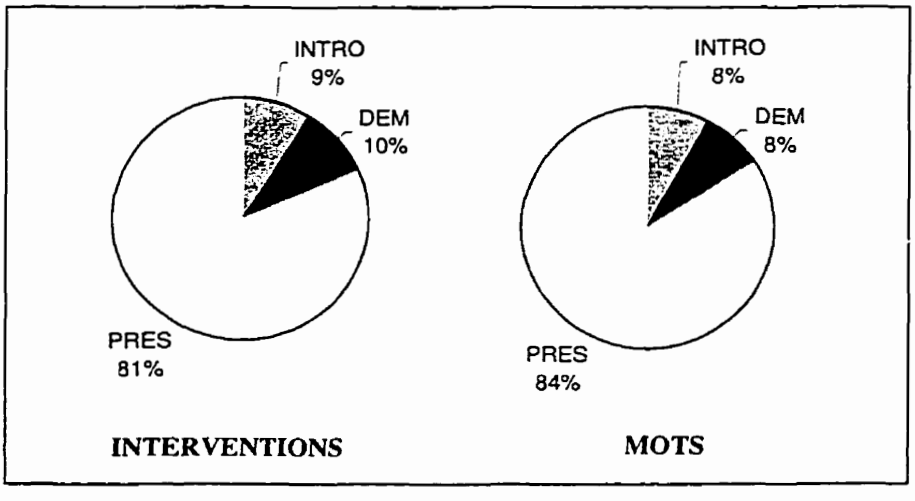



**Figure 5.4 Distribution des groupes dans séquences de révision**

### 5.1.3 Niveaux de conversation

Les 127 séquences de révision à l'étude sont chacune composées d'interventions qui les caractérisent. Ces interventions sont reliées à l'artefact révisé, directement ou indirectement. On peut parler de niveaux de conversation. Une séquence est donc une série d'interventions imbriquées. La détermination de ces niveaux agit comme un filtre sur les données: l'ajout des niveaux permet de délimiter la conversation en segments indépendants pouvant être analysés un par un. Par exemple, le niveau un correspond à toutes les interventions ayant comme sujet l'artefact révisé. Cette granularité de la représentation permet à l'expérimentateur de focaliser ses analyses s'il le désire et de mesurer les relations entre les niveaux s'il le juge nécessaire.

Le Tableau 5.1 présente les deux règles qui déterminent le niveau d'imbrication d'une intervention dans une séquence. Une intervention d'introduction est toujours considérée comme une intervention primaire (niveau 0).

**Tableau 5.1 Règles d'imbrication des échanges**

| INTERVENTION DE RÉFÉRENCE | INTERVENTION IMBRIQUÉE | RÈGLE # |
|---|---|---|
| $a_i / s_j / c_k$ | $a_{i+1} / \{O_1^{n-1}, \ldots, O_m^{n-1}\} / \{c_k\}^0$ | A |
| $a_i / s_j / c_k$ | $* / \{s_{j+1} = c_k\}$ | B |

À l'intérieur d'une séquence, une intervention est caractérisée par le triplet suivant: a: activité / s: sujet / c: critère. $a_i$ désigne le type d'activité, $s_j$ le sujet et $c_k$ le critère de l'intervention. Une intervention imbriquée est désignée par $a_{i+1}$ qui doit être différente

de $a_i$. Le sujet de l'intervention est un objet $O_m^{n-1}$ puisqu'il s'agit d'une intervention antérieure persistant dans le temps. $O_m^{n-1}$ désigne un des m objets persistants au niveau d'imbrication n-1. Les règles d'imbrication se définissent comme suit et sont applicables à tous les niveaux de conversation d'une séquence:

1. Règle A: Une intervention est imbriquée dans l'intervention de référence n si son sujet appartient à l'ensemble des objets instanciés par toute intervention du niveau de référence. Les critères n'ont aucune influence sur cette règle.

2. Règle B: Une intervention est imbriquée dans l'intervention de référence si son sujet est le critère utilisé au niveau de référence.

La Figure 5.5 indique la distribution des interventions par niveau. Environ 67% (niveaux 0 et 1) des interventions sont reliées à la discussion de l'artefact sous révision pour 74% du propos tenu lors de ces séquences. Le niveau 0 correspond à l'introduction d'aspects de l'artefact dans la discussion tandis que le niveau 1 couvre toutes les interventions ayant comme sujet ces aspects et seulement ceux-là. Ce résultat indique que la réunion de révision est principalement dédiée à la discussion de l'artefact. Seulement 26% de toutes les interventions des séquences de révision n'ont pas directement pour sujet l'artefact révisé.

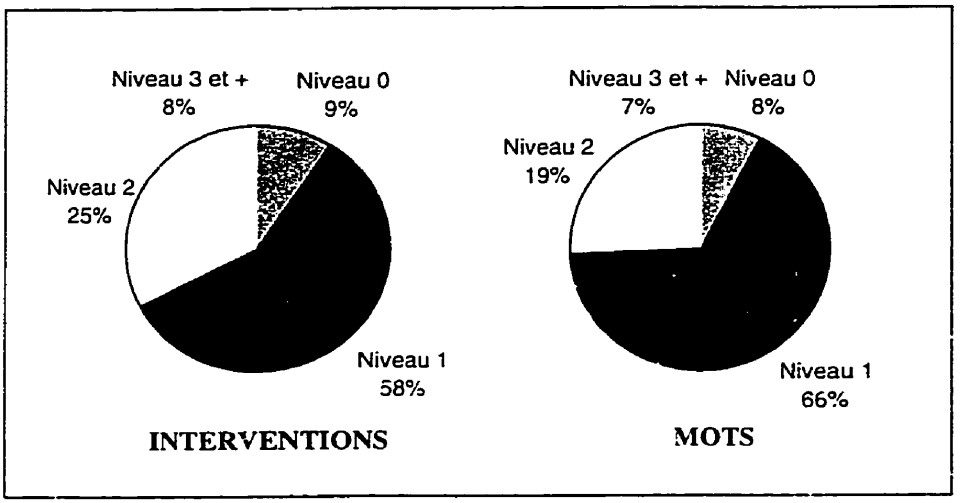



**Figure 5.5 Distribution des interventions par niveau**

La Figure 5.6 indique la distribution des types d'interventions à travers les niveaux. On y voit clairement que les interventions de développement (DEV), évaluation (EVA), hypothèse (HYP) et information (INFO) sont principalement regroupées dans le niveau 1. Le premier niveau n'a aucune intervention de rejet (REJ) ou d'acceptation (ACC), ce qui signifie que l'artefact révisé n'est jamais directement refusé ou accepté par les participants.

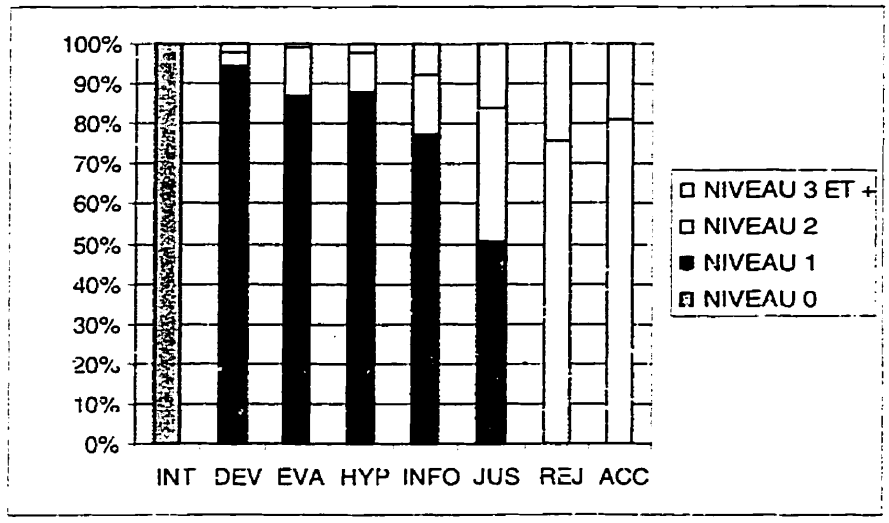



**Figure 5.6 Distribution des interventions par niveau**

Un examen plus approfondi de la nature de ces interventions permet de découvrir (Figure 5.7) que chaque niveau est caractérisé par une différente "signature".

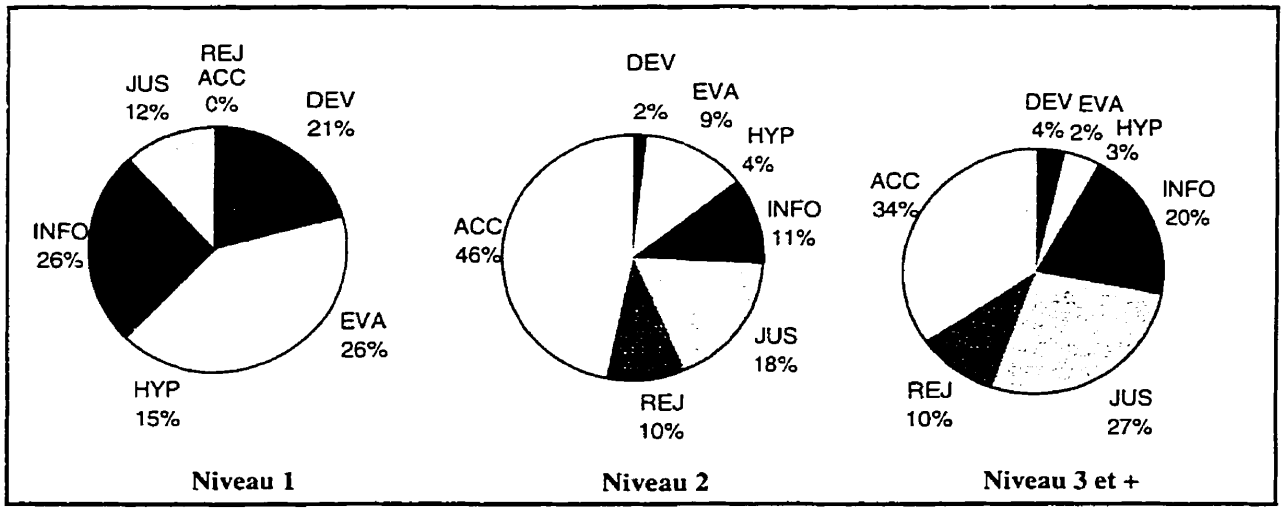



**Figure 5.7 Distribution relative des interventions par niveau**

Au niveau 1, l'artefact agit comme sujet de discussions afin que les participants ajustent (INFO, 26% et HYP, 15%) leur compréhension, évaluent son bien-fondé (EVA, 26%) où développent une nouvelle solution (DEV, 21%). La justification est beaucoup moins

présente, proportionnellement, au niveau 1 que dans les autres niveaux.

Le niveau 2 intègre toutes les discussions générées par le niveau 1. Les interventions n'ont donc pas un sujet de même nature comme au niveau 1 (artefact) mais plutôt plusieurs sujets différents. On voit cependant que l'acceptation et la justification constitue la majorité des interventions de ce niveau (64%), tout comme pour les niveaux 3 et plus (61%). La proportion des interventions de développement et d'évaluation tend à diminuer progressivement au fur et à mesure que le niveau augmente pour laisser sa place à une majorité d'acceptations, justifications, rejets et d'information.

Une analyse des sujets du niveau 2 permet d'identifier un contexte dans lequel les interventions du niveau 2 sont faites. La Figure 5.8 montre que près du tiers des interventions du niveau 2 sont dédiées aux solutions alternatives. Il est aussi important de noter l'apparition des critères comme sujets de discussions au niveau 2.

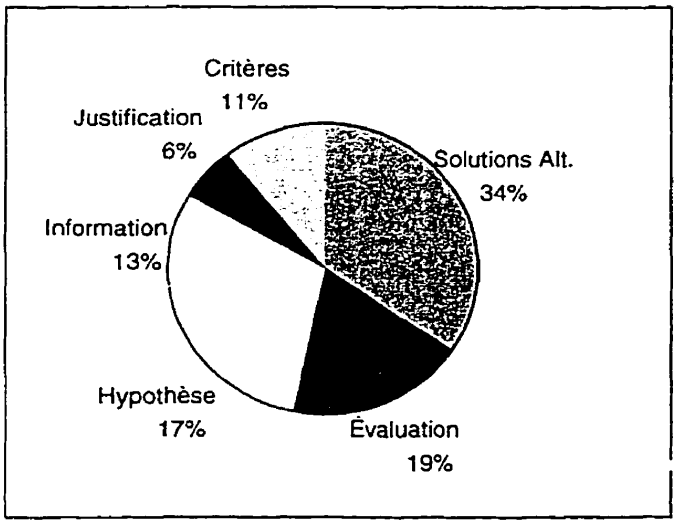



**Figure 5.8 Distribution relative des sujets de deuxième niveau**

Une analyse détaillée des interventions ayant comme sujet un critère (Figure 5.9B) permet de conclure qu'il n'existe pas de représentation partagée des critères utilisés lors de la révision technique. Malgré que tous les participants connaissent les critères devant être satisfaits grâce aux guides méthodologiques qu'ils utilisent, ils continuent de les discuter durant les réunions. Une analyse plus approfondie indique que ces discussions servent à rendre les critères plus explicites (INFO et HYP, 83%) et à ordonner les critères selon leur importance (JUS et EVA, 17%). Cela suggère que le groupe améliore sa connaissance de la procédure de révision durant la réunion en rendant explicite les critères dans le contexte particulier de l'artefact révisé. Ces analyses sont similaires aux résultats de Darses (1997) qui a observé que l'évaluation de solutions mène à l'explicitation des critères utilisés.

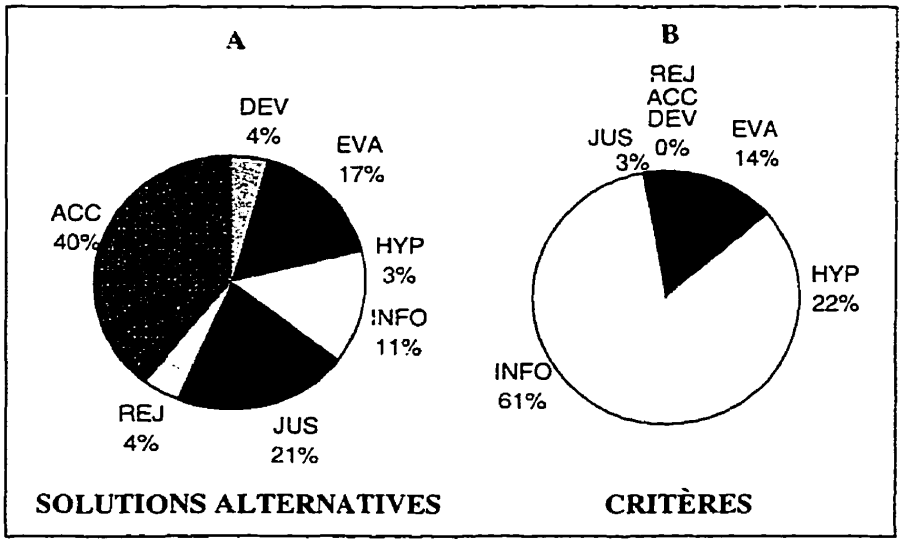



**Figure 5.9 Distribution relative des interventions de niveau 2 ayant pour sujet les solutions alternatives et les critères**

Lorsque le sujet est une solution alternative (Figure 5.9A), c'est-à-dire un développement (DEV) de l'artefact discuté antérieurement dans la séquence, 14% des interventions (INFO et HYP) visent à mieux la comprendre tandis que la majorité (82%) des interventions servent à juger cette nouvelle solution (ACC, EVA, JUS et REJ). Le développement d'une solution durant une réunion de révision est donc habituellement suivi d'une période où les participants tentent de juger la nature de cette solution. Selon la littérature, les activités reliées au développement ne devraient pas avoir lieu lors de la réunion de révision mais être laissées à l'auteur qui modifiera son artefact en fonction des révisions faites. L'analyse des échanges pourra répondre à ce point.

La distribution du propos (mots) tenu lors des réunions par sujets de conversation (Figure 5.10) permet de se rendre compte que, 70% du propos est utilisé pour discuter de l'artefact révisé. Le dernier 30% est-il du temps perdu ? Les critères et les solutions alternatives constituent près de la moitié de ce 30%, il s'agit de juger de la pertinence de leur présence dans la révision d'un artefact.

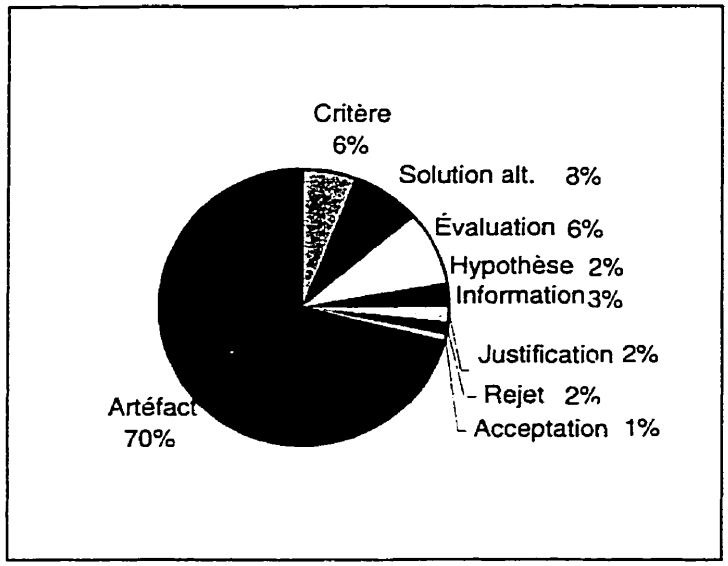



**Figure 5.10 Distribution du propos par sujets**

Une analyse plus en profondeur est nécessaire pour expliquer ces tendances. Les analyses statiques permettent de répondre à plusieurs questions et d'améliorer notre compréhension de ce type de réunions. Elles ne donnent cependant pas beaucoup d'indications sur les facteurs qui influencent l'occurrence des interventions dans ce type de réunion. L'analyse des types d'interventions en fonction du rôle des individus qui les effectuent amène beaucoup d'information très pertinente à l'étude des réunions de révision technique.

## 5.2 Influence du rôle des individus sur les interventions

La réunion de révision technique est constituée d'un ensemble d'interventions reliées dans un réseau dynamique complexe. L'intérêt principal de ce genre de recherche se situe surtout dans la possibilité d'identifier des régularités dans les données recueillies. Le schéma de codage permet d'identifier chacune des interventions individuellement mais

est-il possible d'induire la nature des interventions à partir de paramètres tels que le rôle de l'individu qui fait l'intervention ?

Un individu effectue des interventions à l'intérieur d'un certain contexte. Bien que la nature des interventions du schéma de codage (introduction, évaluation, développement, information, hypothèse, justification, acceptation, demande et gestion) soit relativement générique, il est possible que le type de réunion (révision technique) influence leurs fréquences d'occurrence. L'identification de la dynamique d'une réunion et des facteurs qui l'influencent doit donc passer par la paramétrisation du contexte.

Une caractéristique fondamentale de la réunion de révision technique est la coopération. L'objectif principal de ce genre de réunion est de permettre à plusieurs individus de valider un artefact, les individus ont donc certains rôles à jouer durant la réunion. De plus, ces mêmes individus ont des responsabilités vis-à-vis du déroulement du projet qui influenceront leurs comportements lors de la réunion. On peut supposer qu'il y a interaction entre les participants à la réunion, c'est-à-dire que l'intervention d'un participant amène l'intervention d'un autre. Finalement, il faut mesurer l'influence du rôle des participants sur le déroulement de la réunion. Le Tableau 5.2 présente les rôles identifiés lors d'une réunion de révision technique.

**Tableau 5.2 Caractéristiques des participants lors d'une révision technique**

| | Rôle | Abbr. | Définition |
|---|---|---|---|
| Projet | Expert-procédure | Exp | L'expert en procédure est le gardien de la standardisation de la façon de faire de l'équipe. Il est donc très intéressé par la forme des artefacts et leur adhérence aux standards établis. |
| | Développeurs | Dev | Les autres ingénieurs participants au projet et n'ayant pas de responsabilités spécifiques autres que leur travail individuel. |
| | Chef de projet | Chef | Ce rôle est donné à la personne qui, dans son travail, supervise le travail technique des autres. Il a une vue plus globale du projet. |
| Réunion | Auteur | Aut | Le but de la réunion de révision technique est d'évaluer un artefact produit par un des membres de l'équipe de développement, l'"auteur" du document révisé. |
| | Réviseur | Rev | Tous les participants à la réunion, à l'exception de l'auteur, ont la responsabilité de réviser l'artefact. |
| Tâche | Lien Direct | Dir | Le contenu de l'artefact révisé a-t-il un lien direct avec le travail des réviseurs ? Un lien direct indique que la partie du logiciel décrite dans l'artefact est directement reliée à une autre partie du logiciel développée par un réviseur. |
| | Lien Indirect | Ind | Il n'y a pas de lien direct avec le travail personnel d'un réviseur bien que ce travail fasse partie du même logiciel. |

Les réunions de révision technique observées dans le cadre de cette recherche étaient composées de quatre participants: le chef de projet, l'expert-procédure et deux développeurs. Les rôles décrits dans le Tableau 5.2 ont probablement un impact sur l'occurrence des interventions. Analysées individuellement, les conclusions seraient peut-être difficiles à interpréter sur l'impact des rôles. Il est cependant possible d'utiliser des

plus précises. Une de ces méthodes est la modélisation log-linéaire.

### 5.2.1 Modélisation log-linéaire

#### 5.2.1.1 Théorie

La modélisation log-linéaire (MLL) a pour but de concevoir un modèle théorique à partir de données obtenues empiriquement. La MLL est hiérarchique, c'est-à-dire qu'on construit et vérifie des modèles théoriques additifs construits à partir du logarithme des fréquences de cellules d'un tableau multi-varié de données. Les valeurs attendues de fréquences des cellules sont alors comparées à celles des valeurs obtenues.

La MLL permet d'identifier l'influence mutuelle de variables observées les unes sur les autres à l'intérieur d'un tableau de données. Ce tableau de données est croisé, c'est-à-dire que chaque cellule du tableau contient la fréquence d'occurrence observée d'une situation définie par plusieurs variables. Par exemple, le Tableau 5.3 contient deux variables, chacune de ces variables ayant deux catégories possibles.

**Tableau 5.3 Tableau croisé de données de deux variables**

| | | VARIABLE INDÉPENDANTE (B) | | |
|---|---|---|---|---|
| | | CATÉGORIE 1 | CATÉGORIE 2 | |
| VARIABLE DÉPENDANTE (A) | CATÉGORIE 1 | $t_{11}$ | $t_{12}$ | $t_{1\bullet}$ |
| | CATÉGORIE 2 | $t_{21}$ | $t_{22}$ | $t_{2\bullet}$ |
| | | $t_{\bullet 1}$ | $t_{\bullet 2}$ | N |

On interprète le tableau précédent de la façon suivante: $\mathbf{t_{11}}$ représente la fréquence d'occurrence observée d'une situation ayant comme caractéristique la variable B catégorie 1 et la variable A catégorie 1, $\mathbf{t_{1*}}$ représente la fréquence marginale de la variable A - catégorie 1 tandis que **N** est le nombre totale de données dans le tableau.

L'objectif du MLL est de déterminer un modèle théorique qui explique l'influence d'une ou plusieurs variables indépendantes sur une variable dépendante. Dans l'alternative où les variables sont tout à fait indépendantes l'une de l'autre, la valeur attendue (théorique) de la cellule $\mathbf{t_{ij}}$ (Éqn 5.1) est égale au produit de ses deux fréquences marginales ( $\mathbf{t_{i*}}$ et $\mathbf{t_{*j}}$ ) divisé par le nombre total de données (**N**). La forme logarithmique de cette équation est représentée par l'équation 5.2.

$$t_{ij} = \frac{t_{i*} * t_{*j}}{N} \qquad \text{(Éqn 5.1)}$$

$$\ln t_{ij} = \ln t_{i*} + \ln t_{*j} - \ln N \qquad \text{(Éqn 5.2)}$$

On exprime l'équation 5.2 de façon équivalente en utilisant les paramètres du modèle de l'équation 5.3, qui représente la valeur d'une cellule $t_{ij}$ influencée par les autres variables du modèle.

$$\ln t_{ij} = \mu + \sum_{i}^{A} t_{ij} + \sum_{j}^{B} t_{ij} \qquad \text{(Éqn 5.3)}$$

Dans cette équation, $\mu$ représente l'effet de toutes les données sur $t_{ij}$, $\Sigma(A)$ est l'effet

principal de la variable A et Σ(B) est l'effet principal de la variable B. Un effet principal est la déviation de la fréquence logarithmique moyenne d'une ligne ou d'une colonne du tableau par rapport à la fréquence marginale. Un modèle représentant une certaine association (dépendance entre A et B) serait représenté par:

$$\ln t_{ij} = \mu + \sum_{i}^{A} t_{ij} + \sum_{j}^{B} t_{ij} + \sum_{ij}^{AB} t_{ij} \qquad \text{(Éqn 5.4)}$$

L'équation 5.4 est le modèle saturé des données observées de A et B. La distinction entre les modèles saturés et non-saturés est très importante. Un modèle saturé est parfaitement mais trivialement ajusté aux données puisque les valeurs attendues sont identiques aux valeurs observées. L'objectif du MLL est d'identifier les modèles non-saturés qui sont adéquatement ajustés aux données observées.

Le nombre total de degrés de liberté disponible pour estimer un modèle log-linéaire est égal au nombre de cellules contenues dans un tableau comme le Tableau 5.3. Il faut soustraire à partir de ce total le nombre de paramètres indépendants nécessaires pour vérifier un certain MLL qui est ajusté aux données. La différence équivaut au degré de liberté utilisé pour vérifier un modèle théorique. Le degré de liberté est utilisé lors de l'évaluation des test d'hypothèses.

### 5.2.2 Tests d'hypothèses

Le processus de recherche d'un MLL approprié aux données requiert qu'il soit possible de

mesurer l'ajustement d'un modèle avec les valeurs observées. Le nombre de modèles théoriques possibles est fonction du nombre de variables observées. Le modèle le plus simple est celui où toutes les variables sont indépendantes (aucun effet) et le plus complexe est celui où il y a interdépendance entre toutes les variables (le modèle saturé). Les modèles intermédiaires sont ceux qui doivent être évalués puisqu'il est possible que l'un d'eux représentent les données significativement.

Chacun des modèles théoriques formulés généreront une liste de valeurs attendues pour chacune des variables. La vérification de l'ajustement de ces valeurs attendues avec les valeurs observées est habituellement effectué en utilisant le test du chi-carré. Le test chi-carré consiste à mesurer l'écart qui existe entre des fréquences observées et des fréquences théoriques et à tester si cet écart est suffisamment faible pour être imputable aux fluctuations d'échantillonnage. Il existe plusieurs façons de calculer le chi-carré mais leur interprétation et leur ordre de grandeur sont relativement similaires. Le chi-carré du ratio de la vraisemblance est défini comme suit (Éqn 5.5):

$$G^2 = 2\sum_{i=1}^{r}\sum_{j=1}^{c} O_{ij} \ln\left(\frac{O_{ij}}{t_{ij}}\right) \qquad \text{(Éqn 5.5)}$$

où: $O_{ij}$ est la donnée observée

$t_{ij}$ est la donnée attendue

le degré de liberté = nombre de variables - 1

r et c sont respectivement le nombre de rangées et de colonnes

On évalue la signification d'un chi-carré en fonction du degré de liberté du modèle et du niveau de confiance requis par l'expérimentateur. L'équation 5.6 illustre la distribution du chi-carré.

$$f(x) = \frac{x^{(\upsilon/2)-1} * e^{-x/2}}{2^{\upsilon/2} * \Gamma(\upsilon/2)} \qquad \text{(Éqn 5.6)}$$

où: ν est le degré de liberté

Γ est la fonction Gamma

### 5.2.3 Outils d'analyse

L'ajustement des modèles théoriques aux données observées est un processus itératif qui requiert plusieurs calculs. Il existe des outils logiciels qui automatisent toutes les étapes d'analyses et qui permettent à l'expérimentateur de vérifier ses hypothèses. Le progiciel *Statistica* est reconnu comme un outil adéquat pour l'étude statistique de données. Ce progiciel a un module dédié à la modélisation log-linéaire. Les résultats démontrés par la suite proviendront de ce progiciel.

### 5.2.4 Ajustement d'un modèle théorique

Cette section a pour but de justifier les conclusions tirées à partir des données empiriques en présentant les étapes suivies lors de l'ajustement du modèle théorique. La Tableau 5.4 présente un échantillon des données à modéliser. Les interventions de niveau 1 seulement ont été analysées puisque ce sont les interventions ayant comme sujet les parties des

artefacts révisés et elles représentent 74% de la réunion. Un nombre 10 $R^2$ de données (R étant le nombre total de catégories) est nécessaire (Fagen et Young, 1978) pour obtenir des résultats significatifs. Le cas de l'analyse des interventions nécessite donc un minimum de 250 données (5 catégories – évaluation, développement, justification, information, hypothèse). Un total de 1399 interventions de niveau 1 ont été analysées.

**Tableau 5.4 Échantillon des données empiriques**

| VALU | 1 INTERV | 2 RÉUNION | 3 PROJET | 4 TÂCHE | 5 CRITERE | 6 FREQ |
|---|---|---|---|---|---|---|
| 1 | EVAL | AUTEUR | CHEF | INDIRECT | CONTENU | 1.000 |
| 2 | EVAL | AUTEUR | CHEF | INDIRECT | FORME | 0.000 |
| 3 | EVAL | AUTEUR | CHEF | DIRECT | CONTENU | 9.000 |
| 4 | EVAL | AUTEUR | CHEF | DIRECT | FORME | 3.000 |
| 5 | EVAL | AUTEUR | EXPERT | INDIRECT | CONTENU | 0.000 |
| 6 | EVAL | AUTEUR | EXPERT | INDIRECT | FORME | 0.000 |
| 7 | EVAL | AUTEUR | EXPERT | DIRECT | CONTENU | 1.000 |
| 8 | EVAL | AUTEUR | EXPERT | DIRECT | FORME | 1.000 |
| 9 | EVAL | AUTEUR | DEVEL | INDIRECT | CONTENU | 0.000 |
| 10 | EVAL | AUTEUR | DEVEL | INDIRECT | FORME | 0.000 |
| 11 | EVAL | AUTEUR | DEVEL | DIRECT | CONTENU | 27.000 |
| 12 | EVAL | AUTEUR | DEVEL | DIRECT | FORME | 4.000 |
| 13 | EVAL | REVISEUR | CHEF | INDIRECT | CONTENU | 50.000 |
| 14 | EVAL | REVISEUR | CHEF | INDIRECT | FORME | 63.000 |
| 15 | EVAL | REVISEUR | CHEF | DIRECT | CONTENU | 0.000 |
| 16 | EVAL | REVISEUR | CHEF | DIRECT | FORME | 0.000 |
| 17 | EVAL | REVISEUR | EXPERT | INDIRECT | CONTENU | 9.000 |
| 18 | EVAL | REVISEUR | EXPERT | INDIRECT | FORME | 38.000 |
| 19 | EVAL | REVISEUR | EXPERT | DIRECT | CONTENU | 6.000 |
| 20 | EVAL | REVISEUR | EXPERT | DIRECT | FORME | 3.000 |

Les différentes colonnes représentent les variables à l'étude:

- Interventions (Interv): On distingue cinq types d'interventions au niveau 1 (directement relié à l'artefact), soient évaluation, développement, justification, information et hypothèse. Il faut noter que les demandes

(DEM), dans le cadre de cette analyse, sont transformées en fonction de l'intervention (ex.: DEM/INFO devient INFO). On ne distingue donc pas entre une demande et une présentation de faits dans cette analyse. On restreint notre analyse aux aspects fonctionnels de la conversation seulement.

- Réunion: Ce sont les rôles plus généraux de la réunion, soient auteur de l'artefact et réviseurs.
- Projet: Les rôles des participants dans le projet (chef de projet, expert en procédure et développeur)
- Tâche: Le lien entre l'artefact révisé et le travail des autres participants (direct ou indirect).
- Critères: On distingue s'il s'agit de la forme ou le contenu de la solution qui est discuté.
- Fréquence (Freq): La fréquence d'occurrence d'un cas décrit par les catégories ci-dessus. Il y a 120 cas (permutations des catégories) différents.

Le Tableau 5.5 donne de l'information quant à la nature du modèle potentiel pouvant être tiré des données empiriques. Le nombre de paramètres indique le type de lien (1-aucun

lien entre les variables jusqu'à 5-modèle saturé) entre les variables. Le chi-carré permet de mesurer l'ajustement d'un modèle avec les données observées.

**Tableau 5.5 Probabilité des interactions**

| Nombre de paramètres | Degrés de Liberté | Chi carré | Probabilité |
|---|---|---|---|
| 1 | 9 | 540.186 | 0 |
| 2 | 29 | 1254.559 | 0 |
| 3 | 43 | 129.4003 | 0 |
| 4 | 30 | 12.29782 | 0.998229 |
| 5 | 8 | 0.001498 | 1 |

Le progiciel utilisé fonctionne en terme d'hypothèse. Dans le cas présent, la dernière colonne indique la probabilité que **le modèle contenant un certain type de lien ne représente pas mieux les données que le type de lien inférieur**. Cette probabilité est calculée en fonction du chi-carré et du degré de liberté possible dans le modèle. On peut donc conclure, en fonction des résultats du Tableau 5.5, que le modèle ajusté contiendra des liens entre trois variables.

On évalue par la suite l'apport possible de chacun des liens sur le modèle théorique. Cette évaluation s'effectue en fonction des associations partielles et marginales. Un lien est associé partiellement si son absence réduit significativement l'ajustement du modèle théorique contenant toutes les interactions. Le même lien est associé marginalement à un modèle théorique si sa présence dans un modèle sans lien améliore significativement ce modèle. Le Tableau 5.6 représente les associations partielles et marginales pour les liens de type 3.

**Tableau 5.6 Associations marginales et partielles**

| Interaction | Degrés de | Ass.Prt. | Ass.Prt. | Ass.Mrg. | Ass.Mrg. |
|---|---|---|---|---|---|
| | Liberté | Chi-carré | p | Chi-carré | p |
| 123 | 8 | 5.10 | 0.75 | 69.89 | 0.00 |
| 124 | 4 | 0.70 | 0.95 | 2.38 | 0.67 |
| 125 | 4 | 7.27 | 0.12 | 18.01 | 0.00 |
| 134 | 8 | 8.46 | 0.39 | 67.34 | 0.00 |
| 135 | 8 | 9.96 | 0.27 | 7.79 | 0.45 |
| 145 | 4 | 6.66 | 0.16 | 11.91 | 0.02 |
| 234 | 2 | 1.97 | 0.37 | 4.59 | 0.00 |
| 235 | 2 | 4.32 | 0.12 | 13.32 | 0.00 |
| 245 | 1 | 0.22 | 0.64 | 0.82 | 0.36 |
| 345 | 2 | 10.00 | 0.01 | 17.36 | 0.00 |

Les probabilités sont reliées à des hypothèses inhérentes au logiciel. L'hypothèse est que **la présence du lien n'aura pas d'effet significatif sur le modèle théorique partiel ou marginal**. Par exemple, le lien 345 (qui représente un lien entre les rôles dans le projet, la tâche et les critères) possède des associations partielles et marginales significatives. Cet lien devrait donc faire partie du modèle théorique.

Les liens 123, 125, 134, 234, 235, 345 ont été choisis pour composer le modèle théorique. Ces choix ont été faits non pas seulement en fonction de leurs associations marginales et partielles mais aussi en fonction des besoins de l'analyse. En effet, les liens ayant de l'influence sur les interventions (donc ayant un 1, voir le Tableau 5.6) sont ceux ayant le plus d'intérêts pour l'instant.

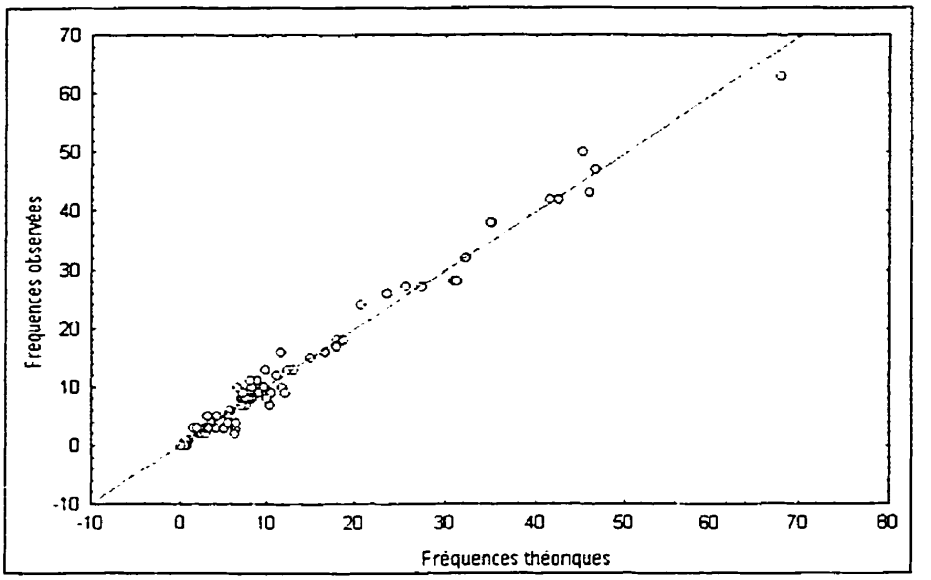



**Figure 5.11 Graphe des fréquences observées en fonction des fréquences théoriques**

La qualité de l'ajustement du modèle théorique peut être évaluée à l'aide de graphes comme celui de la Figure 5.11. Ce graphe représente les fréquences du modèle théorique par rapport aux fréquences observées. Un modèle théorique parfaitement ajusté aurait tous les points sur la droite.

Une autre représentation plus précise de l'ajustement d'un modèle théorique aux données est un graphique illustrant les résiduels (différence entre une fréquence observée et une fréquence théorique) en fonction des fréquences théoriques (Figure 5.12). Un résiduel de 0 indiquerait un ajustement parfait. Enfin, le chi-carré du modèle théorique est calculé ( 26,43) pour un modèle théorique contenant 55 degrés de liberté: cette valeur est significative dans 99% des cas, ce qui nous permet d'affirmer que le modèle utilisé reflète bien les données empiriques.

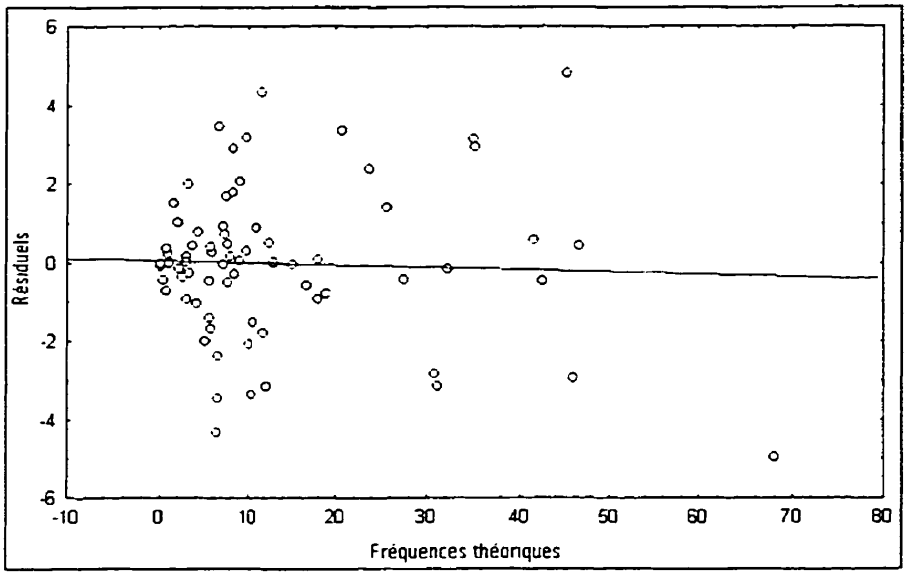



**Figure 5.12 Graphe des résiduels en fonction des fréquences théoriques**

### 5.2.5 Analyse du modèle théorique

Le modèle théorique construit à partir des données et validé grâce à la modélisation log-linéaire peut maintenant être analysé afin d'identifier des facteurs qui influencent l'occurrence des interventions. L'annexe D contient les tableaux marginaux obtenus à l'aide de la MLL. Il est important de comprendre que les valeurs utilisées dans cette section ne proviennent pas des données empiriques mais plutôt du modèle théorique. Un tableau tel que celui présenté ci-dessous (Tableau 5.7) présente des résultats sous une forme qui permet de comparer différentes situations.

**Tableau 5.7 Exemple de valeurs théoriques**

Critère: Forme

| INTÉRÊT | RESPONS CHEF | RESPONS EXPERT | RESPONS DEVEL | Total |
|---|---|---|---|---|
| INDIRECT | 140 | 97 | 36 | 273 |
| DIRECT | 17 | 11 | 64 | 92 |
| Total | 157 | 108 | 100 | 365 |

Critère: Contenu

| INTÉRÊT | RESPONS CHEF | RESPONS EXPERT | RESPONS DEVEL | Total |
|---|---|---|---|---|
| INDIRECT | 155 | 64 | 86 | 305 |
| DIRECT | 95 | 25 | 176 | 296 |
| Total | 250 | 89 | 262 | 601 |

La Figure 5.13B présente les interventions en deux groupes distincts: les interventions ayant trait aux critères de contenu et celles ayant trait aux critères de forme. Cette distinction faite, on peut analyser l'effet des responsabilités dans le projet (chef de projet, expert en procédures, développeur) sur l'occurrence des interventions.

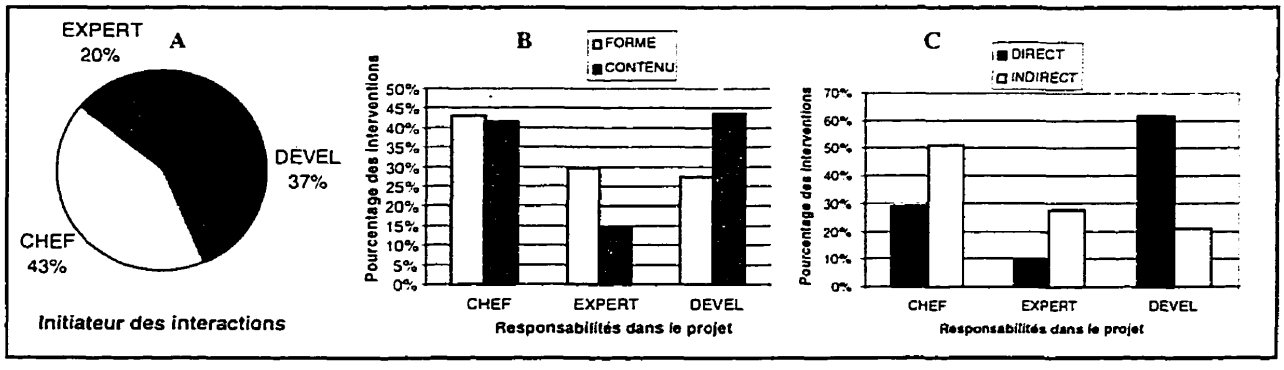



**Figure 5.13 Influence des critères sur les responsabilités de projet**

Une première analyse permet de constater que le chef de projet effectue une grande partie des interventions (43%) de niveau 1 lors des réunions (Figure 5.13A) et ce, malgré le fait qu'il y ait toujours deux développeurs qui participent aux réunions. L'influence du chef de projet sur la révision de l'artefact est considérable. On voit d'ailleurs que la participation relative du chef de projet aux discussions de forme et de contenu de

l'artefact est relativement similaire (Figure 5.13B). L'expert en procédure participe deux fois plus aux discussions de forme que de contenu, ce qui est conforme avec son rôle dans le projet. Les développeurs, quant à eux, semblent être plus intéressés par le contenu que par la forme de l'artefact.

La Figure 5.13C illustre une situation intéressante: il semble que le chef et l'expert soient plus présents lors de discussion sur un artefact n'ayant qu'un lien indirect avec leur propre travail tandis que les développeurs participent beaucoup moins (trois fois moins) à la discussion d'un artefact n'ayant pas de lien direct avec leurs travaux. Bien que le chef participe à 43% de toutes les interventions, sa contribution relative augmente substantiellement lorsqu'il s'agit d'un lien indirect, ce qui semble démontrer que les autres participants sont moins intéressés à réviser lorsqu'il ne s'agit pas d'un artefact important pour eux. La participation relative du chef de projet qui, en tant que superviseur, semble être intéressé par l'ensemble du projet augmente donc significativement.

Les responsabilités vis-à-vis le projet influencent donc le cours d'une discussion, mais le rôle à l'intérieur même de la réunion (auteur vs réviseur) a lui aussi de l'importance. La Figure 5.14 montre bien le clivage important que créent ces rôles sur la discussion de la forme et du contenu d'un artefact.

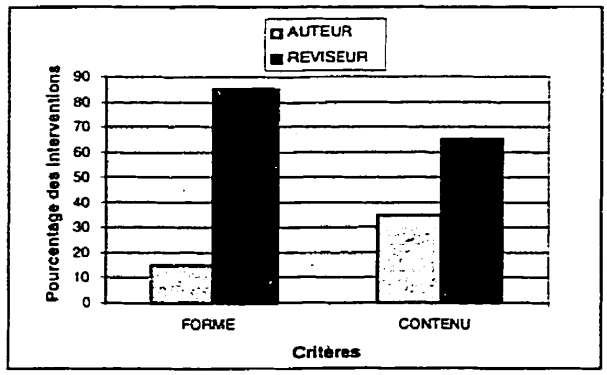
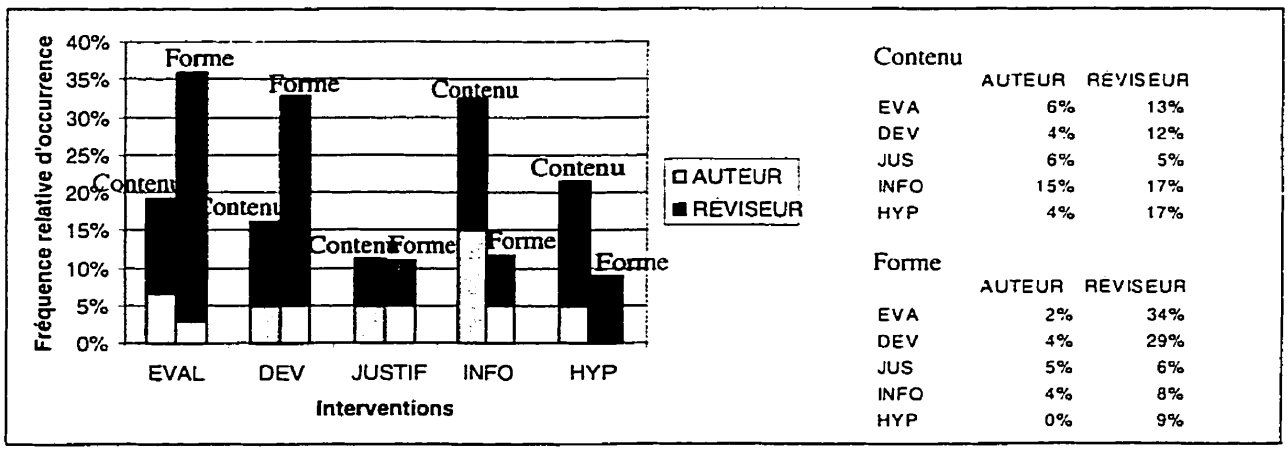



**Figure 5.14 Contribution relative aux discussions de forme et de critère**

La contribution relative de l'auteur à une discussion de la forme de son artefact est seulement de 15% tandis que ce même auteur contribue à 35% des interventions relieés au contenu de ce même artefact. Une analyse des interventions elles-mêmes permet d'expliquer ce phénomène.




Contenu

| | AUTEUR | REVISEUR |
|---|---|---|
| EVA | 6% | 13% |
| DEV | 4% | 12% |
| JUS | 6% | 5% |
| INFO | 15% | 17% |
| HYP | 4% | 17% |

Forme

| | AUTEUR | REVISEUR |
|---|---|---|
| EVA | 2% | 34% |
| DEV | 4% | 29% |
| JUS | 5% | 6% |
| INFO | 4% | 8% |
| HYP | 0% | 9% |

**Figure 5.15 Occurrence relative des interventions**

Plus de 69% des interventions reliées à la forme de l'artefact sont des évaluations et des développements (Figure 5.15) tandis qu'on ne s'informe (information et hypothèse) que très peu (12%) de la forme. Il apparaît donc que la forme de l'artefact est, dans

l'ensemble, bien assimilée par l'équipe de développement. Les discussions ayant trait à la forme de l'artefact sont des corrections (DEV, EVA) apportées à l'artefact par les réviseurs et ne génèrent que très peu de discussions.

Le comportement des participants est très différent dans le cas de discussions du contenu de l'artefact. Près de 53% des interventions permettent aux individus de comprendre le problème (information et hypothèse). La réunion de révision permet donc à l'auteur et aux réviseurs de se mettre d'accord sur le contenu de l'artefact pour ensuite, s'ils le jugent nécessaire, y apporter des corrections (évaluation et développement).

La nature des interventions et les facteurs qui influencent leurs occurrences permettent de comprendre le travail accompli durant une réunion de révision technique. Ces analyses plus statiques ne nous donnent cependant pas de réponse quant au cheminement cognitif suivi par les participants lors de la réunion. Une analyse de l'enchaînement des interventions permet de mieux caractériser ce type de réunion. Ces échanges entre participants nous indiquent la nature du travail coopératif lors d'une réunion de révision technique.

## 5.3 Identification des échanges

L'humain est une créature d'habitudes. On peut émettre l'hypothèse qu'il est peut-être possible de retrouver des comportements organisés de façon séquentielle lors de son travail. La reconnaissance de ces structures séquentielles permet de mieux comprendre

comment une réunion de révision technique s'effectue.

La structure séquentielle d'un comportement interactif peut être décrite autant quantitativement que qualitativement, l'objectif étant d'utiliser ces deux façons pour détecter et identifier des comportements types. Deux méthodes sont utilisées pour caractériser les comportements : l'analyse statistique (quantitative) permet de détecter des patrons de comportements significatifs et l'analyse grammaticale (qualitative) permet de définir des règles d'écriture de ces patrons. Les données analysées grâce à ces méthodes consistent en plusieurs suites d'interventions caractérisant les séquences observées et modélisées à l'aide du schéma de codage développé lors de ce projet de recherche. Le but de ce schéma de codage est de caractériser le contenu (interventions) des séquences survenant lors de la révision d'un artefact produit dans le cadre d'un processus de développement logiciel.

### 5.3.1 Lag Sequential Analysis

Cette méthode permet de déterminer si l'occurrence d'une catégorie de données est influencée par une autre plus souvent que par chance. Il est possible d'utiliser cette technique pour identifier une catégorie qui en suit une autre à plus d'un intervalle de différence. Par exemple, si on fait l'hypothèse que la suite ABC est significative, il est nécessaire de calculer le LSA (un intervalle) de AB et BC ainsi que le LSA (deux intervalles) de AC. Si les trois LSA sont significatifs, les données prouvent l'hypothèse.

Le LSA est donc une mesure de la dépendance d'une catégorie sur une autre. On utilise la cote-z comme mesure de dépendance. Lorsqu'on veut évaluer les probabilités en utilisant une distribution normale, il est nécessaire d'évaluer les régions sous la courbe (normale). Il existe cependant un nombre infini de distributions normales correspondant à $\mu$ (moyenne) et a $\sigma$ (écart-type). La variable "normale" z permet d'unifier ces deux valeurs (équation 5.7), X étant une variable aléatoire appartenant à la courbe normale définie par $\mu_x$ et $\sigma_x$. La cote-z mesure le nombre d'écart-types de distance d'une observation par rapport à la moyenne.

$$z = \frac{X - \mu_x}{\sigma_x} \qquad \text{(Éqn 5.7)}$$

Bien que la puissance du LSA et de la cote-z pour ce genre de travail ne fasse aucun doute, il existe plusieurs différends quant à la façon la plus appropriée de calculer cette statistique. Allison et Liker (1982) ont proposé une telle statistique (équation 5.8) qui est encore aujourd'hui la plus utilisée. Elle se sert de méthodes développées pour l'analyse des tables de contingences. $P_A$ et $P_B$ sont les probabilités d'occurrence de A et B. $P_{B|A}$ représente la probabilité d'occurrence de la catégorie B conditionnelle à la catégorie A. n est le nombre d'échantillons et k le décalage séquentiel entre A et B.

$$z = \frac{p_{B|A} - p_B}{\sqrt{\frac{p_B(1-p_B)(1-p_A)}{(n-k)p_A}}} \qquad \text{(Éqn 5.8)}$$

Le numérateur peut être considéré comme un index de la connexion séquentielle. Ainsi, si $p_{B|A}$-$p_B$ est différent de zéro, cela signifie que la probabilité conditionnelle est significativement différente de la probabilité inconditionnelle. Le numérateur divisé par son écart-type (le dénominateur) est considéré comme une distribution normale. À un niveau de confiance de 95%, si $|z| > 1.96$, il existe alors une connexion séquentielle entre A et B avec un intervalle de k.

### 5.3.2 Analyse grammaticale

Une seconde approche permettant l'analyse des structures séquentielles qui complète les analyses statistiques est la reconnaissance des patrons syntaxiques. Cette approche utilise un formalisme grammatical pour représenter les patrons de comportements (Gonzalez & Thomason, 1978). Les descriptions statistiques (telles le LSA) des patrons se basent sur la contiguïté des événements, ou autrement dit, le calcul de l'influence de l'occurrence d'un événement sur l'occurrence d'un autre. Il reste cependant encore plusieurs séquences qui contiennent une structure enchâssée qui ne peut être détectée par les analyses statistiques.

Les modèles formels (ou approches grammaticales) ont été utilisés, par exemple, pour définir les comportements usager-machine (de Haan et al., 1991; Harrison et Thimbleby, 1990). Les règles syntaxiques sont utilisées pour décrire comment une structure précise

est bâtie. On utilise des règles de réécriture, un formalisme connu en analyse grammaticale comme le *definite-clause grammars* (DCG) développé par Pereira et Shieber (1987) qui utilise des règles très strictes de réécriture. La technique utilisée est appelée *réécriture* plutôt que *transformation* parce que son but est de réécrire les occurrences de patrons dans une séquence et pas nécessairement de fournir une transformation complète de ceux-ci. Une illustration de la différence de résultat est démontrée dans Olson et al. (1994).

### 5.3.3 Utilisation de ces techniques

L'analyse grammaticale est utilisée pour identifier la structure des séquences retrouvées lors de réunions de révision technique. À première vue, une réunion semble très informelle et interactive, ce qui rend l'identification de structures plus difficile. La recherche des structures est guidée par les considérations suivantes:

1. La connaissance des types d'interventions qui, selon la littérature, devraient être présents lors d'une réunion de révision technique (Fagan,1976; d'Astous, 1998; Johnson, 1996);

2. L'analyse statistique des dépendances (avec le LSA) entre les types d'interventions qui permet d'identifier des structures potentielles.

La combinaison d'une méthode statistique à une méthode de réécriture permet d'observer des structures distinctives dans les données. On utilise un processus de raffinement

cyclique dont le but est d'identifier ces structures. Lorsqu'une structure est observée et considérée comme statistiquement significative, elle est remplacée par une nouvelle syntaxe grammaticale et l'analyse statistique est refaite en utilisant les structures réécrites.

L'emphase est mise sur les interventions de niveau 1 parce qu'elles ont la particularité d'avoir le même type de sujet, soit un aspect de l'artefact révisé. Chacune des séquences est réécrite de façon à identifier la suite d'interventions effectuées au niveau 1. Les interventions appartenant aux autres niveaux ont été regroupées dans une seule catégorie (NIV). La Figure 5.16 montre une partie des données.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | INT | INF | INF | HYP | INF | NIV | INF | HYP | INF | HYP | NIV | NIV | JUS | EVA | INF | DEV | NIV | NIV | FIN | |
| 11 | INT | INF | INF | FIN | | | | | | | | | | | | | | | | |
| 12 | INT | INF | EVA | HYP | NIV | HYP | NIV | NIV | DEV | NIV | NIV | NIV | NIV | DEV | DEV | DEV | NIV | INF | INF | FIN |
| 13 | INT | INF | INF | INF | EVA | HYP | DEV | NIV | NIV | FIN | | | | | | | | | | |
| 14 | INT | FIN | | | | | | | | | | | | | | | | | | |
| 15 | INT | INF | EVA | INF | INF | INF | INF | NIV | INF | FIN | | | | | | | | | | |
| 16 | INT | EVA | EVA | EVA | INF | EVA | EVA | FIN | | | | | | | | | | | | |

**Figure 5.16 Exemples de séquences**

Le chiffre identifie la séquence. Par exemple, la séquence 15 provient du codage de la réunion 02 qui a eu lieu au début du projet de développement. Cette séquence contient une intervention d'introduction suivie de (au premier niveau) une intervention d'information, une intervention d'évaluation, quatre interventions d'information, une intervention de niveau 2 et + et une d'information. La séquence 14 représente une séquence où l'artefact a été introduit, mais où aucune discussion n'a suivi.

5.3.3.1 Analyse des données brutes

Le LSA est utilisé pour identifier les liens significatifs sur les données telles qu'illustrées dans la Figure 5.16. Les résultats (Tableau 5.8) peuvent se lire comme suit : en abscisse, nous avons l'intervention précédente d'un couple et en ordonnée, l'intervention suivante. Les liens significatifs ($z > 1,96$) sont en caractères gras.

**Tableau 5.8 Résultats du LSA sur les données brutes**

| INTERVENTION SUIVANTE \ INTERVENTION PRÉCÉDENTE | INT | DEV | EVA | HYP | INF | JUS | NIV |
|---|---|---|---|---|---|---|---|
| INT | -5.78 | -3.76 | -2.21 | -4.33 | -3.88 | -3.37 | 0.61 |
| DEV | 1.80 | **4.64** | **3.57** | -2.05 | -2.43 | -2.32 | **12.48** |
| EVA | **4.28** | -0.71 | **5.31** | -4.15 | -0.98 | -2.33 | **3.46** |
| HYP | 1.17 | -1.44 | -1.16 | **3.16** | **11.72** | -0.18 | **14.73** |
| INF | -0.70 | -4.15 | -2.25 | -0.18 | **9.13** | -3.05 | **6.73** |
| JUS | 0.33 | 1.46 | **8.98** | 0.73 | 0.30 | **6.76** | **19.10** |
| NIV | -10.32 | -4.26 | -6.02 | -4.63 | -6.72 | -5.02 | **7.72** |

Il faut noter qu'un lien ayant un z inférieur à -1,96 permet d'affirmer avec un coefficient de confiance de 95% qu'il n'existe pas de lien entre deux interventions. Par exemple, l'intervention d'introduction ne suit spécifiquement jamais une autre intervention. On peut interpréter ce premier résultat de la façon suivante: l'introduction est présente au début de chacune des séquences, elle permet donc aussi d'indiquer la fin de la séquence précédente. Cela signifie donc qu'on ne peut, à cette étape de l'analyse, savoir si un certain type d'interventions est plus susceptible de terminer une séquence.

Un mot sur l'interprétation de z. Nous savons que la valeur absolue de 1,96 nous permet d'être confiant à 95% du résultat. Une valeur plus élevée (INF-INF, 9.13) indique seulement le niveau de confiance de la mesure et c'est tout. Il n'est pas possible, par

exemple, de comparer ce résultat à un autre (EVA-EVA, 5.31) et affirmer qu'un lien est plus significatif qu'un autre. Z permet seulement d'affirmer avec plus de confiance que le lien DEV-DEV est significatif.

Ces premiers résultats permettent d'affirmer qu'il y a un lien marquant entre deux interventions identiques adjacentes (la diagonale du tableau). Les interventions individuelles des participants à la réunion sont donc souvent suivies par une autre de même nature, effectuée par un autre participant. Il existe des échanges de natures similaires aux interventions. La Figure 5.17 est une représentation graphique des liens significatifs entre les types d'interventions.

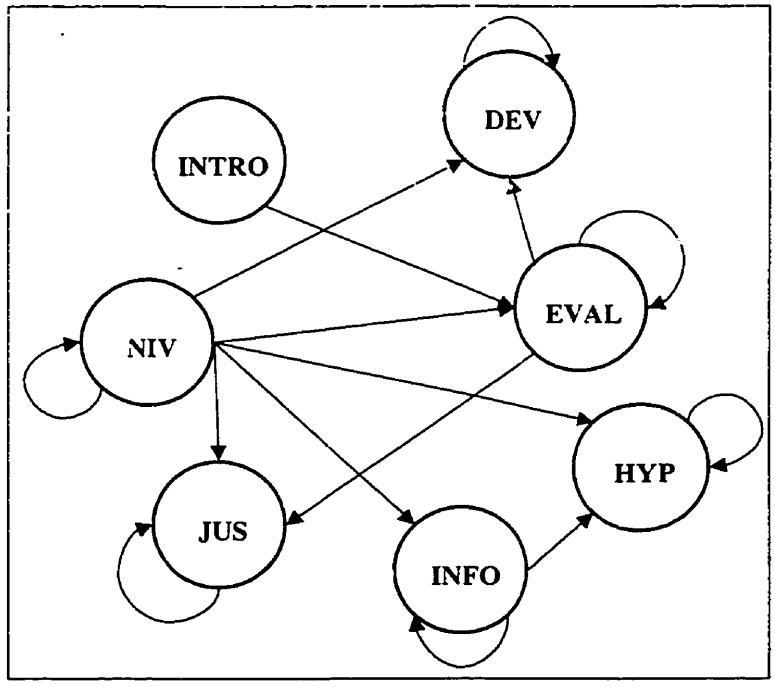



**Figure 5.17 Représentation graphique des liens significatifs entre interventions**

5.3.3.2 Regroupement des interventions en échanges

La première réécriture de nos données consiste à remplacer les suites d'interventions identiques par un seul échange ayant le même nom. Les séquences deviennent donc à ce point-ci une suite d'échanges. Un échange correspond à "la plus petite unité dialogale" (Kerbrat-Orecchioni, 1998). Un dialogue est constitué de différents échanges qui eux-mêmes sont constitués d'interventions.

$$\textbf{INTERVENTION} \; ! \; \textbf{INTERVENTION}^{+} = \textbf{ÉCHANGE} \qquad \text{(Éqn 5.9)}$$

L'équation 5.9 présente la règle de réécriture utilisée pour regrouper les interventions en échanges. Ainsi, une intervention ou une suite d'interventions de même type deviennent un échange unique de ce type. La Figure 5.18 montre le résultat de cette opération sur quelques séquences. Ainsi, la séquence 13 débute avec une introduction suivie d'un échange d'information, une évaluation, une hypothèse, etc. Il faut noter que l'introduction est toujours une intervention.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | INT | INF | HYP | INF | NIV | INF | HYP | INF | HYP | NIV | JUS | EVA | INF | DEV | NIV | FIN |
| 11 | INT | INF | FIN | | | | | | | | | | | | | |
| 12 | INT | INF | EVA | HYP | NIV | HYP | NIV | DEV | NIV | DEV | NIV | INF | FIN | | | |
| 13 | INT | INF | EVA | HYP | DEV | NIV | FIN | | | | | | | | | |
| 14 | INT | FIN | | | | | | | | | | | | | | |
| 15 | INT | INF | EVA | INF | NIV | INF | FIN | | | | | | | | | |
| 16 | INT | EVA | INF | EVA | FIN | | | | | | | | | | | |

**Figure 5.18 Séquences avec les interventions jointes**

Une analyse des séquences réécrites est effectuée de nouveau à l'aide du LSA. La Figure 5.19 représente graphiquement les résultats mathématiques obtenus. Seulement la représentation graphique sera utilisée dans le reste de ce chapitre. Les résultats mathématiques sont disponibles à l'annexe C. Ce processus cyclique continue jusqu'à ce que l'on obtienne une représentation des données qui permet de déterminer les comportements caractéristiques de ce genre de réunions. En fait, on utilise un nouveau cycle d'analyse tant et aussi longtemps qu'il est possible de détecter un patron significatif dans les résultats. La réécriture de ce patron doit permettre d'éclaircir les données et d'en faciliter leur interprétation.

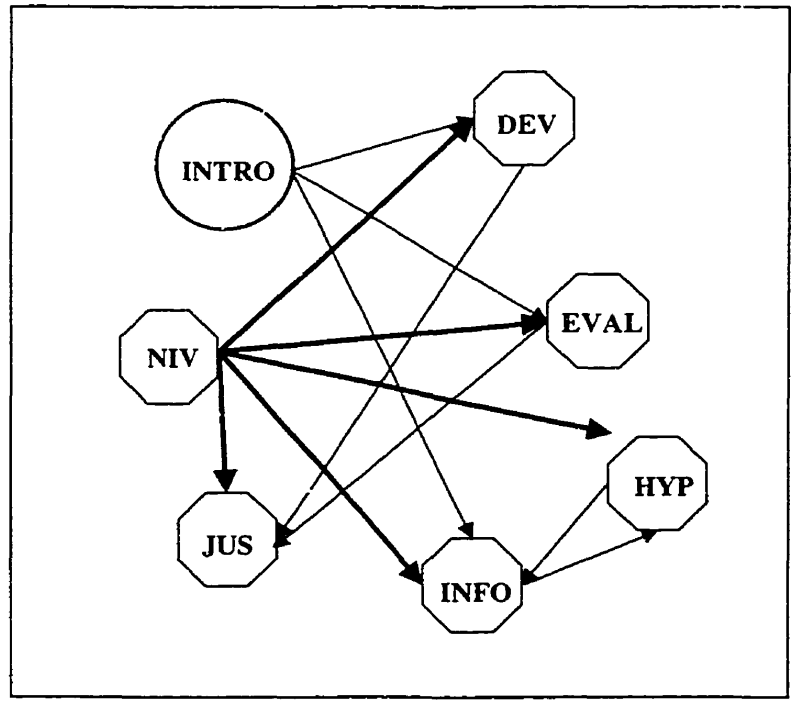



**Figure 5.19 Liens significatifs entre les échanges**

### 5.3.4 Identification des échanges significatifs

Le cycle d'analyse a été reproduit plusieurs fois afin d'obtenir une vision simplifiée et précise d'une réunion de révision technique. Dans un premier temps, les échanges (le résultat du premier cycle d'analyse expliqué dans la section précédente), ont été analysés. Cette section explique en détail chacun des cycles d'analyse nécessaires à la détermination des échanges significatifs lors d'une réunion de révision technique.

#### 5.3.4.1 Les niveaux d'échanges

L'identification de niveaux dans une conversation permet de diviser le propos tenu en fonction des sujets discutés. La révision d'un artefact lors d'une séquence générera probablement des propos sur des sujets reliés à cet artefact. L'ajout d'une catégorie contenant ces propos (NIV) dans l'analyse avait pour but de savoir si ces propos avaient un effet significatif sur les échanges de niveau 1.

Les liens significatifs entre les échanges de niveaux 2 et plus illustrés à la Figure 5.19 font ressortir un patron intéressant: bien que tout type d'échanges de niveau 1 suivent significativement les échanges NIV, il appert qu'il n'y a pas d'échanges qui précèdent significativement les échanges NIV. On peut conclure que, bien que ces échanges fassent partis du propos tenu lors de la réunion, elles ne semblent pas avoir d'impact direct sur la structure des discussions de niveau 1. Cette découverte permet de faire une nouvelle réécriture qui devrait clarifier les résultats: tout échange NIV sera jumelé à l'échange de

niveau 1 suivant (équation 5.10). Ainsi, un échange de niveau 1 peut ou ne peut pas contenir un échange NIV. De plus, afin d'éliminer la redondance d'échanges, les échanges de même type successifs sont regroupés (équation 5.11).

$$\textbf{ÉCHANGE} \mid (\textbf{NIV} \bullet \textbf{ÉCHANGE}) = \textbf{ÉCHANGE} \qquad \text{(Éqn 5.10)}$$

$$\textbf{ÉCHANGE} \mid \textbf{ÉCHANGE}^{+} = \textbf{ÉCHANGE} \qquad \text{(Éqn 5.11)}$$

La Figure 5.20 démontre l'influence de ce regroupement sur la structure de quelques séquences analysées. Ce regroupement permet de réduire le nombre de catégories analysées à six (INT, DEV, EVA, JUS, HYP, INF).

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | INT | INF | HYP | INF | HYP | INF | HYP | JUS | EVA | INF | DEV | FIN |
| 11 | INT | INF | FIN | | | | | | | | | |
| 12 | INT | INF | EVA | HYP | DEV | INF | FIN | | | | | |
| 13 | INT | INF | EVA | HYP | DEV | FIN | | | | | | |
| 14 | INT | FIN | | | | | | | | | | |
| 15 | INT | INF | EVA | INF | FIN | | | | | | | |
| 16 | INT | EVA | INF | EVA | FIN | | | | | | | |

**Figure 5.20 Structure des séquences**

L'analyse des liens significatifs a été de nouveau effectuée sur les séquences modifiées à l'aide du LSA. Les résultats sont illustrés graphiquement dans la Figure 5.21. Cette figure

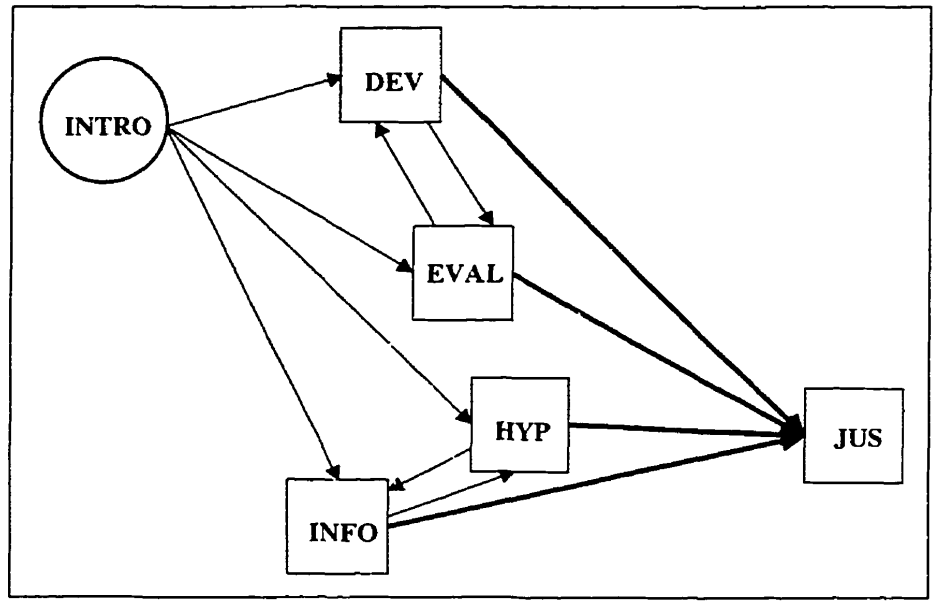



**Figure 5.21 Liens significatifs après regroupement des niveaux**

montre que le regroupement des interventions en échanges permet de discerner des comportements à l'intérieur d'une réunion beaucoup plus facilement.

5.3.4.2 Échanges de justification

Une première constatation peut être effectuée sur les échanges de *justifications* (JUS). L'occurrence d'un tel échange semble être liée à chacun des autres types d'échanges sauf l'*introduction* (INTRO). Inversement, la *justification* ne semble pas avoir d'influence sur l'occurrence des autres échanges. On peut donc tenter de mesurer l'impact de ce type d'échange sur la discussion en cours. L'équation 5.12 montre la réécriture nécessaire pour ce nouveau regroupement. Chacun des types d'échanges (représentés par un carré) peut être jumelé à un échange JUS. Encore une fois, les échanges consécutifs de même type

sont regroupés en un seul échange (équation 5.11).

$$\textbf{ÉCHANGE} \mid (\textbf{ÉCHANGE} \bullet \textbf{JUS}) = \textbf{ÉCHANGE} \qquad \text{(Éqn 5.12)}$$

La Figure 5.22 montre la structure de quelques séquences à la suite de cette réécriture. Une catégorie est éliminée, soit la *justification* (JUS).

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 INT | INF | HYP | INF | HYP | INF | HYP | EVA | INF | DEV | FIN |
| 11 INT | INF | FIN | | | | | | | | |
| 12 INT | INF | EVA | HYP | DEV | INF | FIN | | | | |
| 13 INT | INF | EVA | HYP | DEV | FIN | | | | | |
| 14 INT | FIN | | | | | | | | | |
| 15 INT | INF | EVA | INF | FIN | | | | | | |
| 16 INT | EVA | INF | EVA | FIN | | | | | | |

**Figure 5.22 Structure des séquences après regroupement des échanges JUS**

La Figure 5.23 montre les liens significatifs résultant du regroupement de la *justification* dans les données observées. Il n'y a eu aucun effet sur le résultat. Ceci nous permet de déduire que bien que la *justification* existe, elle n'influence pas directement l'issue finale de la réunion. Il appert toutefois que la *justification* soit inhérente à tout échange, quelque soit son type. Un participant doit souvent justifier les propos qu'il vient de tenir, ce qui semble normal dans le cadre d'une réunion où l'on doit émettre un opinion sur un sujet complexe comme c'est le cas lors d'une réunion de révision technique.

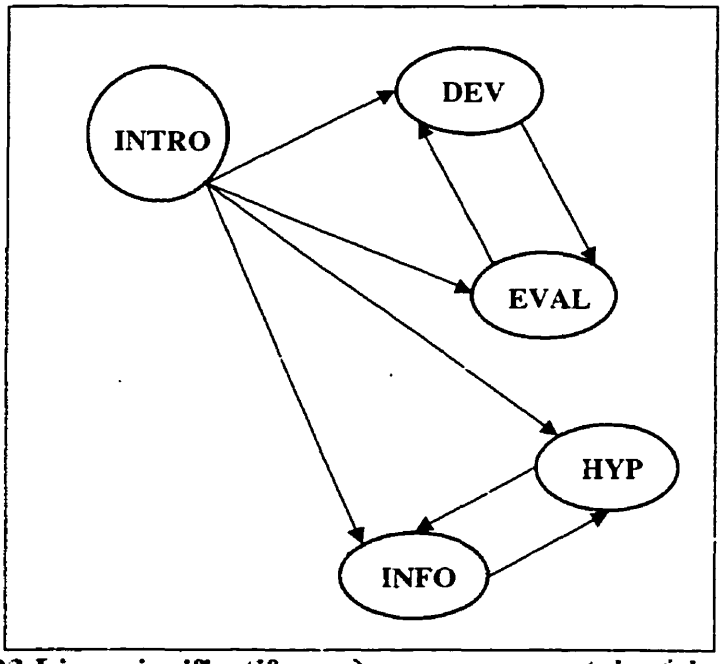



**Figure 5.23 Liens significatifs après regroupement des échanges JUS**

Deux autres résultats ressortent aussi de ce cycle d'analyse. Les échanges *évaluation-développement* et *information-hypothèse* démontrent, à première vue, des comportements similaires. En effet, il semble exister un lien sémantique important entre les deux couples d'échanges. Le LSA ne permet pas d'identifier la nature de ce lien, une autre méthode d'analyse statistique est nécessaire.

5.3.4.3 Analyse de la sémantique des couples d'échanges

Le regroupement hiérarchique (hierarchical clustering) (Johnson, 1967) est une technique statistique utilisée pour étudier les similarités à l'intérieur de structures séquentielles. Une mesure de la similarité est nécessaire: le résiduel standard, c'est-à-dire la différence entre la fréquence observée et attendue divisées par la racine de la fréquence observée, est

utilisé. Un résiduel standard représente une déviation de la chance, la chance étant ce qui est prédit par les fréquences de l'échange précédent et suivant de la transition. Deux échanges ayant des résiduels standards similaires tendent à dévier d'un patron aléatoire de la même façon. La distance euclidienne (équation 5.13) est utilisée pour mesurer la similarité des échanges, c'est-à-dire la distance entre les points dont les coordonnées sont formées des fréquences obtenues.

$$\textbf{Distance}\ (x,y) = [\Sigma_i\ (x_i - y_i)^2]^{1/2} \qquad \text{(Éqn 5.13)}$$

La Figure 5.24 illustre les résultats du regroupement hiérarchique effectué sur les échanges présents après le dernier cycle d'analyse. Le graphique montre très clairement que les échanges *hypothèse* et *information* sont très similaires en tant qu'échange suivant dans une transition. Les autres types d'échanges ne discernent pas une grande différence entre une *hypothèse* et une *information* comme échange suivant. En se basant sur cette analyse, il semblerait possible d'inclure ces deux échanges à l'intérieur d'une seule.

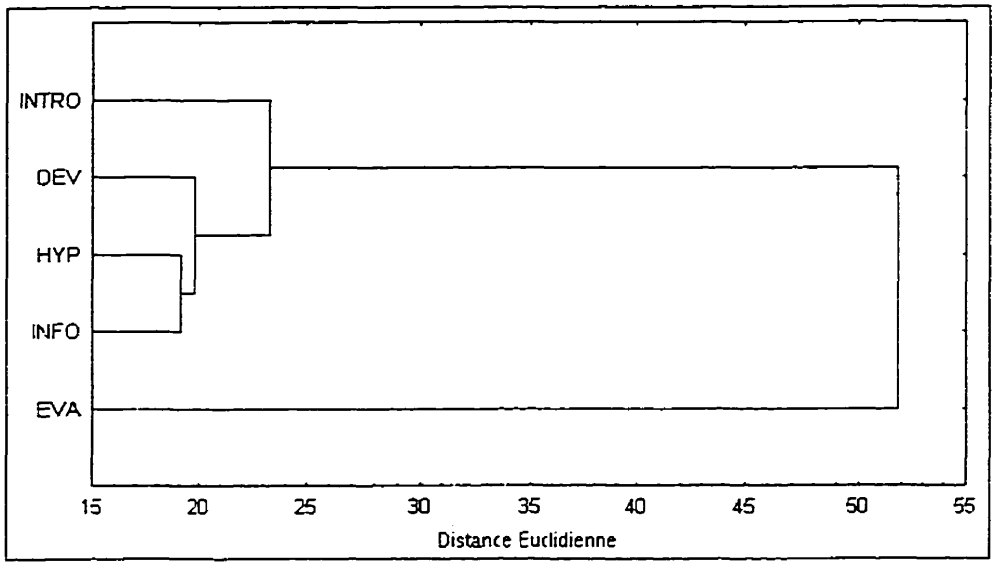



**Figure 5.24 Regroupement hiérarchique des échanges**

Les échanges *évaluation* et *développement* sont cependant très différents l'un de l'autre si l'on en juge la Figure 5.24. D'après ces résultats, il n'est pas possible de les inclure dans un même échange.

Il semble cependant exister une relation entre ces deux échanges qui semblent souvent s'influencer l'un l'autre. Trois réécritures sont nécessaires pour refléter les résultats obtenus à l'aide du regroupement hiérarchique:

1. Tous les échanges *information* et *hypothèse* sont regroupés à l'intérieur du même échange;

$$\mathbf{INF \mid HYP = H_I} \qquad \text{(Éqn 5.14)}$$

2. Les échanges *évaluation* et *développement* restent indépendants, mais forment un nouvel échange lorsqu'ils sont adjacents;

$$\mathbf{EVA \bullet DEV \mid DEV \bullet EVA = E_D} \qquad \text{(Éqn 5.15)}$$

3. Les échanges adjacents de même types sont regroupés pour n'en former qu'un seul (équation 5.11).

La Figure 5.25 montre la structure des séquences de révision après cette dernière réécriture.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10 | INT | H_I | EVA | H_I | DEV | FIN | |
| 11 | INT | H_I | FIN | | | | |
| 12 | INT | H_I | EVA | H_I | DEV | H_I | FIN |
| 13 | INT | H_I | EVA | H_I | DEV | FIN | |
| 14 | INT | FIN | | | | | |
| 15 | INT | H_I | EVA | H_I | FIN | | |
| 16 | INT | EVA | H_I | EVA | FIN | | |

**Figure 5.25 Réécriture des échanges**

La Figure 5.26 montre le résultat de l'analyse statistique (LSA) sur la dernière réécriture des séquences. On peut interpréter ces résultats de deux façons. Les nomenclatures *Hyp-info* et *Éval-dev* sont arbitraires et n'indiquent pas l'ordre de l'échange. Ils indiquent bien sûr les transitions typiques retrouvées lors d'une réunion de révision technique. Ces résultats indiquent aussi ce qui ne semble pas avoir lieu durant une telle réunion. Les comportements déterminants semblent se produire seulement dès qu'un sujet est introduit dans la discussion et que l'échange *Hyp-info* est souvent l'intervention précédente des autres échanges.

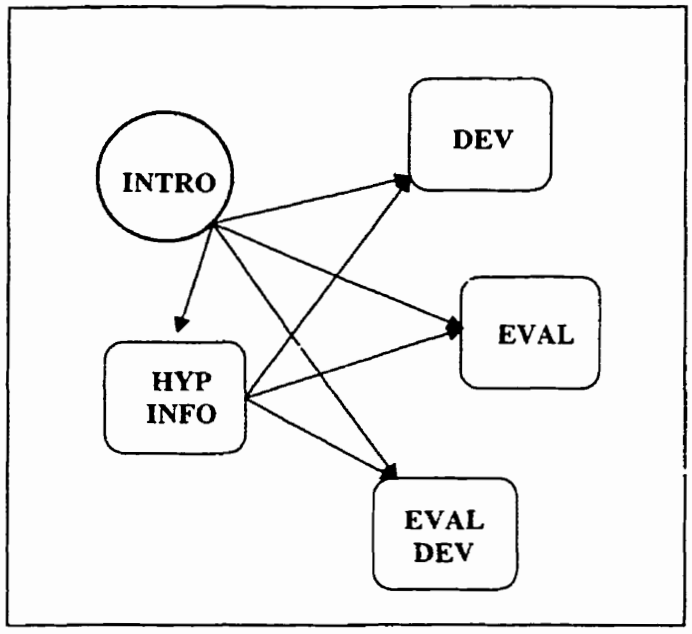



**Figure 5.26 Liens significatifs entre échanges**

L'échange *Éval-dév* démontre une relation entre les activités d'évaluation et de développement. Cela peut être expliqué par le fait que l'évaluation d'une solution amène les participants à tenter d'identifier explicitement une alternative qui améliorerait cette solution. Ce phénomène est intéressant puisque la réunion de révision technique, d'après la plupart des méthodologies de génie logiciel, ne doit pas servir de rencontre de design mais plutôt de vérification d'un design. De plus, il semble que l'effet inverse soit aussi présent: le développement d'une alternative à une solution pourrait permettre de mieux évaluer cette dite solution.

Les résultats obtenus semblent aussi indiquer l'importance relative des échanges *Hyp-info* dans ce genre de discussion. Le lien entre les activités d'évaluation et de clarification a déjà été démontrée par Karsenty (1991) lors d'une étude de réunions de validation de

design de bases de données dont l'objectif était aussi d'évaluer. Falzon (1996) a décrit ce genre d'échange (*Hyp-info*) comme de la synchronisation parce qu'il permet aux participants de s'assurer qu'ils partagent une représentation commune de l'état d'un sujet de discussion. Une réunion de révision technique serait donc aussi une excellente opportunité pour les participants d'échanger l'information et les connaissances nécessaires à la continuation du projet.

### 5.3.5 Sommaire de l'analyse des échanges

Afin de parvenir à la dernière structure des séquences de révision analysées, plusieurs réécritures ont été nécessaires. La Figure 5.27 illustre les différents symboles utilisés pour représenter les réécritures graphiquement.

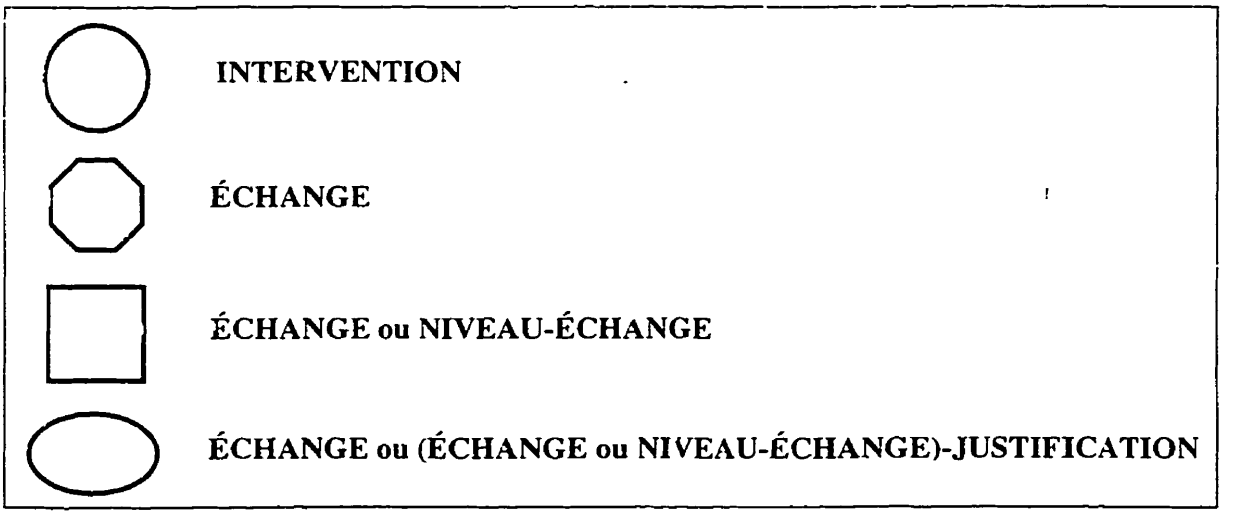



**Figure 5.27 Transformations par réécriture des séquences**

Il est important de comprendre que les réécritures ne retirent pas d'information des structures. Elles sont plutôt une façon de faciliter la compréhension des

enchaînements types en effectuant un éclaircissement des données disponibles. Ainsi, on constate que, bien que les échanges NIV et JUS soient présents dans la plupart des séquences de révision analysées, leurs regroupements avec d'autres types d'échanges permet d'identifier des échanges beaucoup plus significatifs pour la révision technique. Le prochain chapitre présente les déductions tirées des analyses effectuées dans ce chapitre.

# CHAPITRE VI

# MODÉLISATION DE LA PRATIQUE

Les résultats du chapitre précédent démontrent que la méthode exploratoire utilisée pour étudier les réunions de révision technique fonctionne dans le contexte de ce projet de recherche. La discussion de ces résultats se fera sur deux fronts: l'utilisation des résultats pour le génie logiciel et l'apport de la représentation formelle dans l'obtention de ces résultats.

## 6.1 Utilisation des résultats en génie logiciel

Cette recherche exploratoire a permis d'identifier et de mesurer plusieurs caractéristiques des réunions de révision technique. Cette section discute des résultats obtenus et comment ces résultats peuvent améliorer le domaine du génie logiciel et plus spécifiquement le processus de développement logiciel.

### 6.1.1 Nature des échanges caractéristiques d'une réunion de révision technique

La distribution des interventions en niveaux de discussions à l'intérieur des séquences permet d'effectuer un premier filtrage efficace de la réunion de révision technique. L'identification du niveau des interventions montre que 80% des interventions sont reliées à l'artefact révisé. Ce filtrage par sujets facilite les analyses ultérieures en réduisant

le champs d'analyse à sa plus "simple expression". Sachant que toutes les interventions du niveau 1 sont reliées au même type d'artefact, ce paramètre (le sujet) peut être retiré des analyses dynamiques et de l'influence des participants.

La réunion de révision technique est constituée d'un ensemble d'interventions reliées dans un réseau dynamique complexe. L'intérêt principal de ce genre de recherche se situe surtout dans la possibilité d'observer et d'identifier des régularités (ou patrons) dans les données recueillies. Les résultats démontrent qu'il existe des types d'échanges propres à un genre de réunion comme par exemple la réunion de révision technique.

Chacun de ces échanges est composé d'un ou plusieurs types d'interventions faisant parties du schéma de codage. Quatre échanges (d'Astous et als., 1998) ont été identifiés à partir des réunions observées. Le Tableau 6.1 présente les significations données aux différents échanges.

**Tableau 6.1 Signification des échanges**

| ÉCHANGE | SIGNIFICATION |
|---|---|
| **Hypothèse-Information** | **Synchronisation cognitive (Falzon, 1996)**: Permet aux participants de s'assurer qu'ils partagent une représentation commune de l'état d'un objet. |
| **Évaluation** | **Révision**: Les participants jugent la valeur ou donnent leur opinion d'un objet. La révision peut être négative ou positive. |
| **Évaluation-Développement** | **Raffinement**: L'interaction entre l'évaluation et le développement permet aux participants d'émettre des alternatives à la solution du document dans le but de raffiner la solution existante. |
| **Développement** | **Élaboration d'alternatives**: Conception et analyse de solution qui ne font pas partie du document original (artefact). |

Les analyses effectuées démontrent l'importance relative des échanges de synchronisations cognitives (information et hypothèse) sur le contenu de la solution (60%) comparée à la révision ou le raffinement (35%) de celui-ci malgré que, selon la littérature, la révision soit l'objectif principal de ce genre de réunion. La relation entre l'activité d'évaluation et de clarification (synchronisation cognitive) a déjà été démontrée par Karsenty (1991) dans une étude de réunions de validation de bases de données dont l'objectif principal était aussi l'évaluation. Notre analyse va cependant plus loin en identifiant la cause de ce phénomène: une représentation partagée de l'objet de la révision

est un pré-requis à son évaluation puisque la synchronisation cognitive est souvent un précurseur à tout autre type d'échanges.

Un autre résultat est l'occurrence d'activités de développement à l'intérieur d'une réunion de révision. La majorité des méthodes de révision technique retrouvées dans la littérature spécifient clairement que le développement de solutions alternatives ne doit pas avoir lieu durant ce type de réunion. Nos analyses semblent pourtant démontrer que les individus, lorsque laissés à eux-mêmes, effectuent du développement. Cette déviation à la démarche prescrite pourrait être expliquée en utilisant quelques notions de prise de décisions en groupe. Étant donné que le développement de logiciels est un problème mal défini, il existe un grand nombre d'alternatives techniques possibles que l'équipe doit choisir. Un produit logiciel est le résultat d'une démarche décisionnelle où l'artefact révisé représente les décisions techniques prises par un individu sur un problème donné. Lorsque cet artefact est soumis au reste de l'équipe pour révision, ces décisions sont discutées et des alternatives sont proposées.

Les alternatives générées durant l'interaction de groupe (artefact initial et élaborations alternatives) sont les seules disponibles lors de la prise de décision finale. Bien que l'artefact initial soit une proposition bien étoffée puisque son auteur y a passé plusieurs heures de travail, il n'est qu'un sous-ensemble des alternatives qui font l'objet de décisions durant une réunion de révision. Les réviseurs amènent eux aussi des propositions, par le biais du développement, qui doivent faire l'objet d'une décision. L'élimination des

activités de développement signifie le retrait d'alternatives possibles à l'artefact proposé. Puisque les participants à la réunion participent naturellement à ce genre d'activités et qu'elles ont leur utilité, il est proposé d'éliminer cette restriction de méthodes de révision technique.

L'occurrence d'interventions de développement est très spécifique lors d'une réunion de révision technique. Le développement apparaît dans deux types d'échanges:

1. Élaboration alternative: Le développement d'une alternative à l'artefact apparaît souvent dans le cas où un changement à la forme est nécessaire. Dans ces cas là, aucune discussion autre que l'explication du changement à apporter est nécessaire.

2. Raffinement: Le développement interagit avec l'évaluation d'un artefact. Ce résultat est en accord avec Darses et al. (1993) qui ont observé des réunions de design durant lesquelles l'évaluation négative est souvent le précurseur à la proposition de solutions alternatives.

L'intervention de développement ne semble pas être utilisée de la même façon que lors de réunions de design où elle constitue une activité principale. On constate que ce n'est donc pas la nature des interventions individuelles mais bien leurs interrelations dans la formation d'échanges qui caractérisent un certain type de réunion.

### 6.1.2 Rôle des participants à la réunion de révision technique

La participation d'un individu dans une réunion de révision technique semble être influencée par le rôle que celui-ci joue dans l'équipe de développement. Les résultats obtenus démontrent qu'il existe deux types de révision à l'intérieur d'une réunion de révision technique. La révision "de forme", très expéditive, est caractérisée par l'évaluation et le développement d'alternatives simples qui ne sont habituellement pas discutés longuement. Les révisions de forme sont donc la plupart du temps acceptées implicitement par l'auteur. La révision "de contenu" est en majorité une activité de synchronisation cognitive où les participants ajustent leurs visions individuelles à celles des autres. La révision du contenu s'effectue rarement sans un consensus préalable sur la signification d'un artefact.

La forme et le contenu ne sont pas révisés de la même façon. D'une part, la révision est expéditive et est effectuée comme le propose la littérature. D'autre part, la réunion de révision est le cadre d'une vaste opération de synchronisation où les membres du projet s'assurent qu'ils sont sur la même longueur d'onde et qu'ils travaillent tous vers le même but. Bien que les deux ensembles d'activités soient nécessaires, ils n'ont peut-être pas besoin de s'effectuer lors d'une même réunion.

La révision, en tant qu'activité coopérative de prise de décision est cependant très influencée par le chef de projet qui, par ses responsabilités et ses connaissances globales du projet, est capable de convaincre plus facilement les autres de son point de vue. Sa

participation aux réunions de révision semblent donc être nécessaires à leur efficacité. Le besoin de participation des autres individus est cependant plus mitigé. La présence de l'auteur lors de la révision de son artefact permet de faciliter la synchronisation cognitive des autres participants lors de leur travail de révision. La présence des autres individus comme réviseurs n'est peut-être pas nécessaire.

Les développeurs, n'ayant aucun rôle spécifique dans le projet, sont beaucoup moins actifs lors de la révision d'un artefact dont leur travail ne dépend pas directement. Ils tendent d'ailleurs à n'effectuer que des remarques de forme et ne s'étendent que rarement sur le contenu. L'expert-procédure, par sa fonction, se préoccupe de la forme et très peu du contenu. Un lien direct avec l'artefact révisé amène une plus grande implication des participants dans la réunion.

La révision de forme est en général simple et expéditive. Cette révision peut être effectuée à l'extérieur de la réunion par une seule personne (l'expert-procédure) qui soumet par la suite ses corrections à l'auteur. Cette révision terminée, l'équipe se rassemble dans le but de débattre la validité technique de l'artefact en question. Cette réunion serait d'autant plus efficace puisqu'il serait possible de réduire le nombre de participants à ceux qui auraient un intérêt direct dans l'artefact. Plus de temps pourrait être utilisé pour la synchronisation cognitive et le raffinement du contenu de l'artefact original.

La majorité des méthodes de révision technique existantes incluent deux rôles

fondamentaux: l'auteur de l'artefact et les réviseurs. Fagan (1976) est allé plus loin en définissant quelques responsabilités (lecteur, modérateur, etc.) des participants à l'intérieur de la réunion. Les résultats obtenus dans ce travail permettent de confirmer qu'il existe d'autres facteurs qui influencent le déroulement de la réunion.

À l'instar de Seaman et Basili (1998) qui affirment que la structure organisationnelle influence la nature des communications, on peut regrouper l'influence des individus en trois facteurs:

1. Réunion: La littérature définit deux rôles principaux lors d'une réunion de révision technique, soient l'auteur et le réviseur. Les résultats du présent travail permettent d'affirmer que ces deux rôles seuls ne peuvent expliquer totalement l'occurrence des activités coopératives.

2. Projet: Le rôle des individus vis-à-vis le projet global influence la nature de leurs interventions. La réunion de révision n'est donc pas seulement une pratique individuelle avec un seul but mais plutôt une pratique qui influence le développement global d'un logiciel. Il est clair que le chef de projet doit faire partie de la réunion parce qu'il amène une connaissance globale du projet. Il est peut-être possible maintenant de déterminer quels autres individus devraient être présents à la réunion.

3. Tâche: La littérature fait souvent mention du nombre de personnes présents

lors de la réunion de révision. Plusieurs recherches se contredisent quant au nombre optimum de réviseurs à avoir durant la réunion. La réponse ne se situe peut-être pas dans le nombre d'individus mais au contraire dans l'intérêt des individus présents à l'artefact révisé. Nos résultats démontrent qu'il y a un lien sans équivoque entre la tâche des individus et leur participation à la réunion.

### 6.1.3 Comparaison avec un modèle générique existant

Le modèle générique de révision de Tjahjono décrit au chapitre 2 englobe les principales caractéristiques des réunions de révision techniques existantes dans la littérature. L'objectif de cette section est de comparer les résultats obtenus dans ce projet avec le modèle générique afin de déterminer s'il existe des similitudes. Le modèle de Tjahjono permet de comparer des résultats empiriques aux grandes lignes directrices établies dans la littérature quant à la pratique de révision technique. Il n'est cependant pas spécifique à un certain environnement, il ne peut donc prétendre représenter précisément les activités retrouvées lors d'une réunion de révision technique.

Les activités de révision du contenu étudiées dans le cadre de cette recherche s'apparente à celui développé par Tjahjono comme le montre la Figure 6.1. L'échange de synchronisation cognitive est utilisé afin d'assurer une compréhension collective uniforme des participants à la réunion. La synchronisation est souvent un préalable à la révision proprement dite. L'analyse de l'artefact peut correspondre à deux échanges clés identifiés

au cours de ce projet: la révision permet de détecter une anomalie dans l'artefact, tel que définit dans la littérature, tandis que le raffinement va beaucoup plus loin en apportant des correctifs à une anomalie détectée.

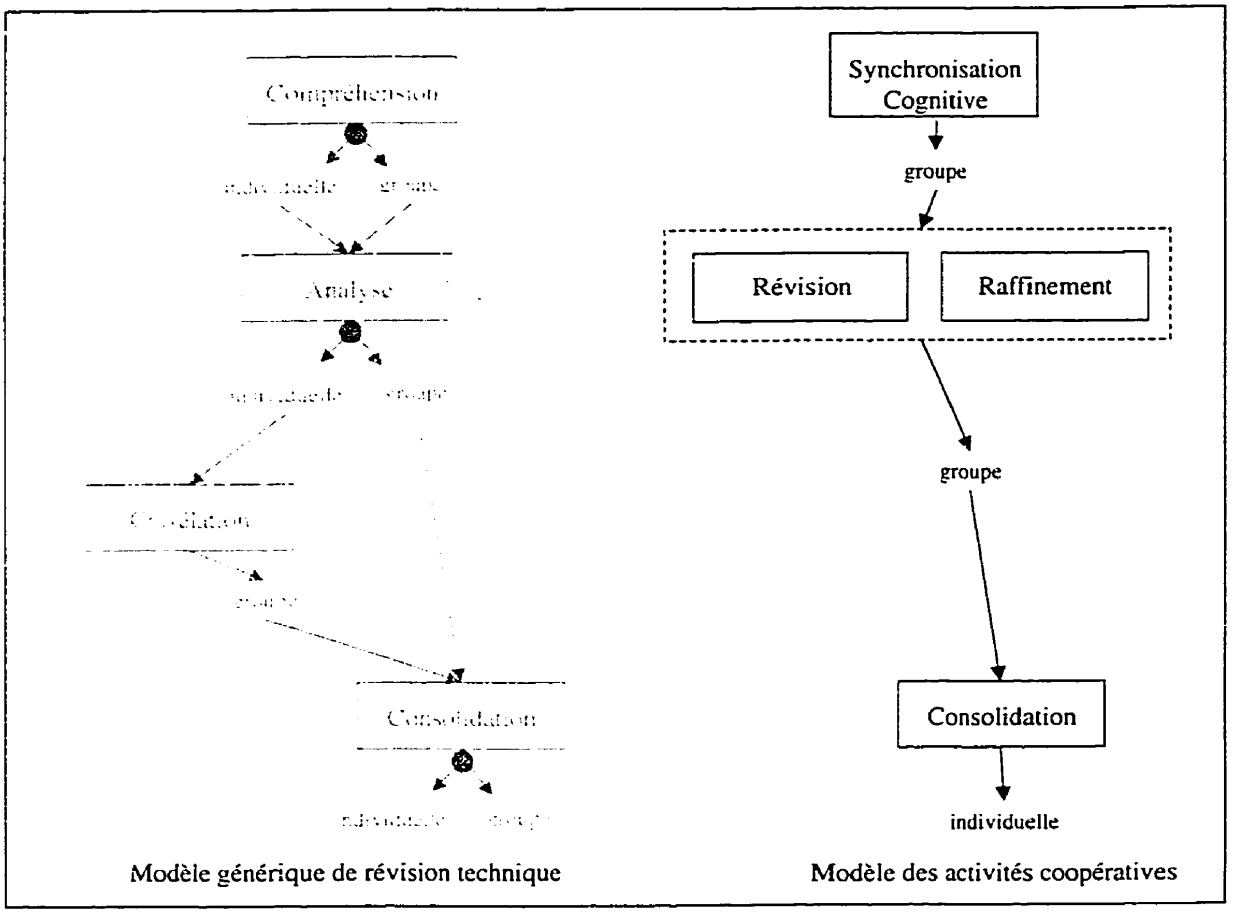



**Figure 6.1 Modélisation de la réunion de révision technique**

La consolidation consiste à s'assurer que les détails discutés durant la réunion ont été répertoriés et classés. Dans son mémoire de maîtrise, Ipperciel (1998) constate, après étude des artefacts modifiés par les réunions de révision analysés dans le présent projet de recherche, que le rapport de consolidation ne semble jamais tenir compte des raffinements discutés lors de la réunion. Pourtant, ce raffinement donne de très bonnes

pistes d'amélioration à l'auteur de l'artefact. On peut argumenter qu'un rôle spécifique de "secrétaire" à la réunion pourrait remédier à la situation en assurant qu'un participant a la responsabilité de noter les conclusions des discussions effectuées.

### 6.1.4 Révision technique à distance

La prolifération des réseaux informatiques et de l'internet ainsi que la tendance à la sous-traitance dans le domaine du génie logiciel aura pour impact de transformer les habitudes organisationnelles. L'une des institutions les plus touchées par ce phénomène sera bien sûr la "réunion face-à-face". Les réseaux permettent maintenant de répartir géographiquement et de façon transparente l'effort de travail. La culture doit maintenant s'adapter à cette nouvelle réalité.

La réunion de révision technique traditionnelle, c'est-à-dire un groupe d'ingénieurs situés dans une même salle pendant quelques heures, doit peut-être maintenant être remplacée par une technique de révision différente, plus évoluée. Un système de révision technique à distance doit pourtant permettre aux individus d'effectuer le même travail significatif que lors de réunions traditionnelles. L'ingénierie d'un système de révision doit tenir compte des facteurs suivants (Johnson, 1996):

1. Le système de révision doit pouvoir s'adapter à des environnements de travail différents;
2. Permettre les révisions asynchrones afin de diminuer le temps de rencontre des réviseurs;

3. L'objectif du système ne doit pas seulement être l'élimination des anomalies mais plutôt l'amélioration de la qualité du processus de développement; et

4. Utilisation d'une technologie d'automatisation pour les révisions techniques.

Un système qui implante le modèle générique de Tjahjono permet-t-il d'obtenir les mêmes résultats qu'une révision face-à-face ? Nos résultats semblent démontrer que ce modèle n'est pas complet et qu'il ne représente pas adéquatement les comportements des réviseurs. Un système de révision doit faciliter l'échange d'information entre les réviseurs sur deux points:

1. Synchronisation cognitive: La révision technique commence dans la majorité des cas par un cycle de mise à niveau de la compréhension de l'artefact. Cette étape est très rapide et expéditive dans un face-à-face, mais qu'en est-il pour une révision asynchrone ?

2. Médiation: Une réunion de révision technique est composé de plusieurs échanges où les participants tentent d'améliorer l'artefact révisé. Nos résultats démontrent que ce n'est pas l'occurrence d'une intervention qui est significative mais plutôt sa relation avec les autres interventions. La possibilité pour les individus d'échanger est donc très importante lors de la révision technique.

Un système doit permettre la médiation entre les réviseurs pour avoir du succès. Un module de médiation permettra de faciliter les deux points ci-dessus. L'analyse des

rôles permet aussi de déterminer la façon dont l'information doit être échangée par les participants. Par exemple, le chef de projet n'est pas intéressé par les mêmes caractéristiques de l'artefact que les développeurs. Le système doit permettre à la bonne personne d'obtenir l'information désirée et pas plus. Une analyse du domaine de la révision plus poussée en utilisant un démarche exploratoire comme celle-ci devrait permettre de répondre à beaucoup de questions.

## 6.2 Efficacité de la représentation formelle

Le succès de ce projet repose entièrement sur la qualité de la représentation formelle des données. Deux caractéristiques fondamentales de cette représentation ont permis d'obtenir les résultats présentés dans ce document: son formalisme et sa syntaxe.

### 6.2.1 Apport du formalisme

Un représentation formelle facilite la description des données et les analyses subséquentes. L'utilisation de la linguistique comme formalisme de représentation des conversations permet l'identification d'unités fondamentales et de leurs liens sémantiques. L'intervention, l'échange et la séquence forment un ensemble structuré d'entités descriptives des conversations.

L'utilisation du paradigme objet comme formalisme pour la représentation conceptuelle des conversations a ajouté de la précision à l'analyse en facilitant la description de

persistance des interventions individuelles. Cette persistance est à la base de l'identification des niveaux de discussion.

Le schéma de codage, par sa structure, regroupe les paramètres d'activité et de sujet de l'activité dans une même description. Ce regroupement permet de définir la notion de niveau de discussion, qui, en tant que filtrage des données, permet de focaliser les analyses sur une partie des discussions seulement.

### 6.2.2 Apport de la syntaxe

La représentation formelle ne peut être efficace sans la définition d'une syntaxe descriptive efficiente. Sa simplicité inhérente a beaucoup contribué au succès de ce projet de recherche. Une syntaxe efficace permet de réduire le travail manuel dans l'analyse, celui-ci se concentrant seulement sur le codage des interventions à partir des données observées. On peut circonscrire l'ajout de biais dans la recherche à cette première étape seulement, ce qui permet de mesurer la fiabilité des analyses à partir de la mesure de fiabilité du schéma de codage.

La syntaxe facilite la conception de procédures d'analyses automatiques. On identifie précisément la nature de l'information nécessaire pour une certaine analyse et on l'obtient rapidement. Le chercheur peut donc se concentrer sur la sémantique des analyses et non pas sur la méthode de lecture des données.

L'utilisation de procédures d'analyses automatiques permet de réduire considérablement le

temps d'analyse (TA) du projet. Ce dernier avantage est considérable parce qu'une démarche exploratoire comme celle utilisée dans le cadre de ce projet est habituellement longue et coûteuse. On peut affirmer que l'utilisation simultanée d'une représentation formelle et d'une syntaxe descriptive efficace permet de réduire de beaucoup les coûts d'un projet comme celui-ci.

# CONCLUSION

Un processus logiciel structure le développement d'un logiciel en instaurant des objectifs très précis de qualité et de productivité ainsi qu'en prescrivant un ensemble de pratiques à mettre en oeuvre. Le succès d'un projet de développement logiciel dépend donc énormément de l'habileté des ingénieurs logiciels à pouvoir accomplir efficacement les pratiques prescrites.

Il est raisonnable de penser qu'il existe une relation entre la qualité du processus utilisé et la qualité du produit logiciel développé par ce processus. La mesure du produit logiciel lui-même a mené à l'amélioration du génie logiciel. La mesure des pratiques du processus logiciel devrait elle aussi contribuer grandement à l'amélioration du domaine. Le génie logiciel ne peut donc que bénéficier de recherches spécifiquement dédiées à l'amélioration des pratiques. La réunion de révision est l'une de ces pratiques.

Le développement de pratiques plus ergonomiques en génie logiciel est la prochaine étape dans l'amélioration continuelle du processus. Cette amélioration est cependant basée sur l'observation et la mesure d'une situation existante. Une démarche exploratoire, bien que son coût soit élevé, permet de faire le pont plus facilement entre la description théorique d'une pratique et son utilisation. Les principes de *l'Exploratory Sequential Data Analysis* (ESDA) ont grandement contribué à organiser l'approche présentée dans ce travail.

Une approche plus structurée de l'observation des réunions de révision technique permet de desceller des comportements déviants de la pratique considérée comme usuelle. L'occurrence de ces "nouveaux comportements" est cependant tout à fait plausible si l'on connaît les principales caractéristiques du travail coopératif en général. Ce travail a identifié et mesuré les activités individuelles et coopératives inhérentes à la réunion de révision technique afin de mieux comprendre son fonctionnement. Une meilleure compréhension de la situation existante permet d'y apporter des correctifs ou de développer une toute nouvelle méthode de révision technique.

L'approche de recherche ainsi que la représentation formelle proposées dans ce travail fournissent une procédure systématique qui simplifie l'observation et l'analyse du travail des ingénieurs logiciels lors des réunions de révision technique. La représentation formelle des données conçue dans le cadre de ce projet de recherche peut réduire considérablement le temps d'analyse des données. De plus elle permet de circonscrire l'ajout de biais à la première étape de représentation des données seulement puisque les étapes subséquentes sont effectuées automatiquement à l'aide d'outils logiciels. Les résultats présentés démontrent la facilité avec laquelle la structure du schéma de codage permet d'extraire de l'information à partir des données. L'utilisation efficace de méthodes statistiques pour identifier et modéliser les comportements valide encore plus l'utilité de la représentation formelle.

La réunion de révision technique est une pratique clé du processus de développement

logiciel. Sa contribution au développement d'un logiciel n'est cependant pas restreinte seulement à la détection rapide d'anomalies mais propose aussi un cadre facilitant la synchronisation cognitive des participants au projet de développement. On ne peut donc pas utiliser seulement le taux de détection des anomalies comme mesure d'efficacité de ce type de réunions.

La littérature contient un grand nombre de méthodes de révision technique, dont la conception est justifiée par l'observation de comportements déviants lors de l'utilisation d'une autre méthode. Cela résulte en l'existence de plusieurs méthodes de révision plus ou moins semblables. La description d'une méthode se restreint souvent à expliquer les activités à accomplir et les résultats attendus, sans vraiment expliciter sur son caractère distinctif.

On peut présumer que l'existence de tant de méthodes est due au manque de compréhension de tous les facteurs qui influencent le déroulement de ce type de réunions. Il est clair que par sa forte occurrence dans le processus de développement, la réunion de révision technique fait partie du processus décisionnel inhérent au développement d'un logiciel complexe. La prise de décision et la solution de problèmes coopératives sont des activités importantes lors de réunions de révision technique que l'on doit tenir compte dans la description d'une pratique.

La réunion de révision technique est la cadre d'un travail coopératif qui a un impact important sur le succès d'un projet de développement. En premier lieu, 60% (Boehm,

1987) des anomalies créées au cours du processus de développement sont détectés lors des réunions de révision technique. Cette réunion permet aussi aux membres de l'équipe de développement d'améliorer les solutions proposées ou d'y apporter des modifications importantes. Finalement, la réunion de révision technique est une plate-forme privilégiée pour la dissémination des connaissances nécessaires au projet. Un résultat du projet de recherche est l'importance relative des échanges de synchronisation cognitive comparés à ceux de révision, qui sont pourtant l'objectif principal d'une réunion de révision technique. Cette activité constitue d'ailleurs la majorité de l'effort dépensé au cours d'une réunion de révision technique.

Une réunion de révision technique devrait être constituée d'au moins deux participants: l'auteur de l'artefact et le chef de projet. La présence de ces deux personnes garantit déjà une certaine efficacité puisque le chef de projet a une connaissance globale du projet qui lui permet de proposer des alternatives intéressantes. Cet embryon original peut être complété par tous les concepteurs qui ont un intérêt direct avec l'artefact révisé. Il y a deux avantages à ce que ces personnes soient présentes: un plus grand nombre d'alternatives à l'artefact original seront présentées au cours de la réunion, ce qui résultera dans un plus grand choix lors des prises de décisions et les participants auront obtenu toute l'information nécessaire pour uniformiser leur compréhension collective d'un artefact.

La révision de forme devrait être effectuée avant la tenue de la réunion de révision

proprement dite. Cette révision permet d'éliminer les anomalies de forme "bénignes" qui ne nécessitent pas vraiment une prise de décision pour laisser plus de temps à la révision des parties importantes (contenu ou forme plus complexe) lors de la réunion. Cette modification à la pratique pourrait grandement améliorer l'efficacité des réunions.

L'identification de la dynamique des échanges, et plus particulièrement l'interdépendance entre les différents types d'échanges fait ressortir plusieurs contraintes pour le design d'un logiciel d'aide au travail coopératif lors des réunions de révision technique. Le support de révisions techniques sur internet ou de tout autre type de réunion est une problématique de l'heure en génie logiciel. Johnson (1996) a développé un outil de révision à distance qui structure la communication en fonction des parties de l'artefact à réviser. Les résultats obtenus dans ce travail semblent démontrer qu'il existe des façons plus efficaces de structurer la communication entre participants d'une équipe de développement "distribuée". Une structure implantée à partir de la modélisation de l'interdépendance des échanges pourraient rendre plus efficace le travail coopératif "électronique".

Les résultats obtenus ont illustré la grande capacité de l'approche exploratoire utilisée et démontré la pertinence de ce genre de travail de recherche pour le domaine du génie logiciel. Bien que la provenance très spécifique des données ne permette pas totalement d'affirmer de la généralité des résultats obtenus, cette recherche a permis de valider la démarche et prouver son intérêt tant au point de vue théorique que pratique.

Afin d'étendre la portée de ces travaux, les suggestions de recherche suivantes sont

présentées comme voies potentielles d'exploration dans ce domaine de recherche en génie logiciel. On distingue deux avenues de recherche possibles: l'utilisation de la représentation formelle dans l'étude d'autres pratiques coopératives du génie logiciel et la conception de pratiques plus "ergonomiques" de révision technique.

Utilisation de la représentation formelle:

- Utiliser la démarche exploratoire pour observer et mesurer les réunions de design logiciel;
- Utiliser le schéma de codage comme structure de codage d'activités individuelles (cognitives); et
- Vérifier la validité de la démarche exploratoire dans un autre domaine que le génie logiciel.

Recherche sur la réunion de révision technique:

- Approfondir les analyses afin de déterminer si d'autres facteurs influencent la réunion de révision technique (par exemple, la nature de l'artefact révisé);
- Répéter les analyses avec des données empiriques provenant d'un autre environnement similaire;
- Répéter les analyses avec des données empiriques provenant d'un environnement avec

des caractéristiques différentes (par exemple, un développement procédural plutôt qu'orienté-objet);

- Utiliser la démarche pour analyser la nature du travail coopératif lors de la révision sur internet; et

- Développer une méthode de révision qui réponde autant aux attentes du génie logiciel qu'à ceux des ingénieurs logiciels.

Il est suggéré que la représentation formelle utilisée dans cette recherche et plus particulièrement le schéma de codage des interventions, sont dissociables de l'approche exploratoire. Ils pourraient être utilisés comme base pour d'autres types de recherche similaires où il existe le besoin de représenter les données brutes dans une syntaxe plus structurée.

Cette recherche démontre qu'il est possible de "mesurer le non mesurable" en génie logiciel et d'obtenir des résultats aptes à améliorer notre compréhension du domaine.

# RÉFÉRENCES

ACKERMANN, F., BUSHWALD, L.S., LEWSKI, F.H. (1989). Inspections: An Effective Verification Process. IEEE Software, mai, 31-36.

ADRION, R.W., BRANDSTAD, M.A., CHERNIAVSKY, J.C. (1993). Validation, Verification and Testing of Computer Software. Computing Surveys, vol. 14, no. 2. 159-192.

ALLISON, P.D., LIKER, J.K. (1982). Analyzing Sequential Categorical Data on Dyadic Interaction: A comment on Gottman. Psychology Bulletin, 91, 393-403.

BALES, R.F. (1950). Interaction Process Analysis: A method for the Study of small Groups. New York, Holt.Rinehart and Winston.

BANGE, P. (1992). Analyse conversationnelle et théorie de l'action. Hatier-Didier.

BELBIN, R.M. (1981). Management Teams. London, Henimann.

BELL, D., MORREY, I., PUGH, J. (1987). Software Engineering: A Programming Approach. Prentice-Hall International, chapitre 14.

BERGER, M.A. (1985). The Technical Approach to Teamwork. Training and Development Journal, mars, 53-55.

BISANT, D.B., LYLE, J.R. (1989). A two-person Inspection Method to Improve Programming Productivity. IEEE Transactions on Software Engineering, octobre, 1294-1304.

BOEHM, B. (1981). Software Engineering Economics. Prentice-Hall.

BOEHM, B. (1987). Industrial Software Metrics Top 10 List. IEEE Software, septembre, 84-85.

BOOCH, G. (1991). Object oriented design with applications. Cambridge, MA, Harvard University Press.

BOSTROM, R.P., KAISER, K.M. (1981). Personality Differences within Project Teams, Proceedings of the 18th Annual Computer Personnel Research Conference, ACM No. 443810.

BROOKS, F.P. (1975). The mythical man-month: Essays on software engineering. Reading, MA, Addison-Wesley.

BROOKS, F.P. (1987). No silver bullet. Computer, avril, 10-19.

BRYKCZYNSKI, B. (1994). Why isn't Inspection used ?, Technical Report at University of Hawaii at Manoa, mars.

BUCK, F.O. (1981). Indicators of Quality Inspections. TR-21.802, IBM Systems Products Division, septembre.

BUSH, C.M., SCHKADE, L.L. (1985). In Search of the Perfect Programmer, Datamation, 985, 128.

COLLOFELLO, J.S. (1988). The Software Technical Review Process, SEI Curriculum Module SEI-CM-3-1.5, Software Engineering Institute.

COMPTROLLER GENERAL. (1979). Contracting for Computer Software Development. FGMSD-80-4, Government Accounting Office, septembre.

DARSES, F. (1997). L'ingénierie concourante: Un modèle en meilleure adéquation avec les processus cognitifs en conception. Ingénierie Concourante. De la technique au social, Paris, Economica.

DARSES, F., FALZON, P, ROBERT, J.M. (1993). Cooperating partners: investigating natural assistance. Proceedings of Human-Computer Interactions, 1993, Orlando.

D'ASTOUS, P. (1997). La mesure des activités collaboratives retrouvées lors d'une réunion de révision technique du processus de génie logiciel. EPM/RT-97/30, Montréal.

D'ASTOUS, P., DÉTIENNE, F., ROBILLARD, P.N., VISSER, W. (1998). Types of dialogs in evaluation meetings: an analysis of technical-review meetings in software development. COOP'98, Cannes, France, 26-29 mai.

DE HAAN, G., VAN DER VEER, G.C., VAN VLIET, J.C. (1991). Formal Modelling Techniques in Human-Computer Interaction. Acta Psychologica, 78, 27-67.

DEWEY, J. (1910). The influence of Darwin on Philosophy, and other essays on contemporary thoughts. Bloomington, Indiana University Press.

DOYLE, M., STRAUSS, D. (1976). How to Make Meetings Work. New York, Jove Books, Chapitre 16.

EMERY, F.E., EMMERY, M. (1975). Participative Design-work and Community. ACT, Australian Centre for Continuing Education, Australian National University, Canberra.

FAGAN, M.E. (1976). Design and Code Inspections to Reduce Errors in Program Development, IBM Systems Journal, vol. 15, no. 3, 182-211.

FAGAN, M.E. (1986). Advances in Software Inspections. IEEE Transactions on Software Engineering, juillet, 744-751.

FAGEN, R.M., YOUNG, D.Y. (1978). Temporal Patterns of Behavior: Durations, Intervals, Latencies and Sequences. Quantitative Ethology, Wiley.

FALZON, P. DARSES, F., BÉGUIN, P. (1996). Collective Design Processes. Proceedings of the Second International Conference on the Design of Cooperative Systems, Juan-les-Pins, France.

FISHER, A.B. (1974). Small Group Decision Making: Communication and Group Process. New York, McGraw Hill, Chapitre 7.

GONZALEZ, R.C., THOMASON, M.G. (1978). Syntactic Pattern Recognition: An Introduction. Addison-Wesley.

GILB, T., GRAHAM, D. (1993). Software Inspection, Addison-Wesley.

HACKMAN, J.R., OLDHAM, G.R. (1980). Work Redesign. Reading, MA. Addison-Wesley.

HARRISON, M., THIMBLEBY, H. (1990). Formal Methods in Human-Computer Interaction, Cambridge University Press.

HARTWIG, F., DEARING, B.E. (1970). Exploratory data analysis. Sage University Paper Series on Quantitative Applications in the Social Sciences, Newbury Park, CA: Sage.

HERBSLEB, J., KLEIN, H., OLSON, G.M., BRUNNER, H., OLSON, J.S., HARDING, J. (1994). Object-Oriented Analysis and Design in Software Project Teams. Human-Computer Interaction, 10, 249-292.

HOEIM, D., SULLIVAN, K. (1994). Designing and using integrated data collection and analysis tools: Challenges and considerations. Behavior and Information Technology, 13, 160-170.

HUMPHREY, W.S.(1989). Managing the Software Process. Addison-Wesley, Reading, Massachussetts, chap. 10, 171-190.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. (1993). IEEE Standards Collection for Software Engineering - Std 1028. IEEE Publishing.

IPPERCIEL, S. (1998). Caractérisation des réunions de révision technique dans un projet de génie logiciel. Projet de maîtrise, mars.

JOHNSON, S.C. (1967). Hierarchical Clustering Schemes. Psychometrika, 32, 241-254.

JOHNSON, P.M. (1996). Reengineering Inspection: The Future of Formal Technical Review. Technical Report at University of Hawaii at Manoa, ICS-TR-95-24, janvier.

KANTER, J., SCHIFFMAN, S. et HORN, J.F. (1990). Let the customer do it; from grocery robots to photo kiosks, computerized self-service is in., Computerworld, Août, 24-35.

KARSENTY, L. (1991). The Validation Dialog for a Database Conceptual Schema. INRIA, Rocquencourt, Framce.

KERBRAT-ORRECCHIONI, C. (1998). Les interactions verbales , Armand Colin.

KNIGHT, J.C., MYERS, E.A (1993). An Improved Inspection Technique. Communications of the ACM, vol. 11, 51-61, novembre.

LETOVSKY, S., PINTO, J., LAMPERT, R., SOLOWAY, E. (1987). A Cognitive Analysis of Code Inspection. Empirical Studies of Programmers - 2nd Workshop, 231-247.

LYONS, M.L. (1985). The DP Psyche. Datamation, août, 103.

MARTIN, J., TSAI, W.T. (1990). N-Fold Inspection: A Requirement Analysis Technique. Communications of the ACM, vol. 33, 225-232, février.

MCGRATH, J.E. (1984). Groups: Interaction and performance, Englewood Cliffs, NJ, Prentice-Hall.

MYERS, G. (1978). A Controlled Experiment in Program Testing and Code Walkthroughs/Inspections. Communications of the ACM, septembre, 760-768.

MYERS, G. (1979). The Art of Software Testing, Wiley and Sons.

MYERS, I.B. (1980). Gifts Differing, Palo Alto, CA, Consulting Psychologists Press.

NAUR, P., RANDALL, B. (1969). Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee, OTAN.

OLSON, G.M., HERBSLEB. J.D.. RUETER. H.H. (1994). Characterizing the sequential structure of interaction behaviors through statistical and grammatical techniques. Human-Computer Interaction, 9, 427-472.

OLSON, G.M., OLSON, J.S., CARTER, M.R., STORROSTEN, M. (1992). Small Group Design Meetings: An Analysis of Collaboration. Human-Computer Interaction, 7, 347-374.

PARNASS, D.L., WEISS, D.M. (1987). Active Design Review: Principles and Practices. Journal of Systems and Software, vol. 7, 259-265.

PAULK, M.C., CURTIS, B., CHRISSIS, M.B., WEBER, C.V. (1993). Capability Maturity Model for Software, Version 1.1, SEI-93-TR-24, Software Engineering Institute.

PEREIRA, F.C.N., SHIEBER, S.M. (1987). Prolog and Natural-Language Analysis (CLSI Lecture Note 10). Menlo Park, Californie.

PMI STANDRARD COMMITTEE (THE). (1987). Project Management Body of Knowledge, Project Management Institute.

PORTER, A., VOTTA, L., BASILI, V.R. (1994). Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. Technical Report at University of Maryland, juin.

PORTER, A., SIY, H., VOTTA, L. (1995). A Review of Software Inspections. Technical Report at University of Maryland, UMCP-CSD:CS-TR-3552, octobre.

PORTER, A., JOHNSON, P.M. (1997). Assessing Software Review Meetings: Results of a Comparative Analysis of Two Experimental Studies. IEEE Transactions on Software Engineering, mars.

PRESSMAN, R.S. (1997). Software Engineering: A Practioner's Approach. McGraw-Hill.

RAGLAND, B. (1992) Inspections are Needed Now More than Ever, Crosstalk, novembre.

REITMAN, W.R. (1965). Cognition and thought. New York, Wiley.

RITTER, F.E., LARKIN, J.H. (1994). Developing process models as summaries of HCI action sequences. Human-Computer Interaction, 9, 345-383.

ROBILLARD, P.N., D'ASTOUS, P., DÉTIENNE, F., VISSER, W. (1998). Measuring Cognitive Activities in Software Engineering. ICSE'98, Kyoto, Japon, 19-25 avril.

ROBILLARD, P.N., D'ASTOUS, P., DÉTIENNE, F., VISSER, W. (1998). An Empirical Method Based on Protocol Analysis to Analyze Technical Review Meetings. CASCON'98, Toronto, 7 décembre.

ROULET, E. (1987). L'Articulation du discours en français contemporain, École de Genève.

RUSSEL, G.W. (1991). Experience with Inspection in Ultralarge-scale Developments. IEEE Software, 25-31, janvier.

SANDERSON, P.M., FISHER, C. (1994). Exploratory Sequential Analysis: Foundations. Human-Computer Interaction, janvier, 251-317.

SEAMAN, C.B., BASILI, V.R. (1998). Communication and Organization: An Empirical Study of Discussion in Inspection Meetings. IEEE Transactions on Software Engineering, juillet, 559-572.

SCHACH, S.R. (1990). Software Engineering, Aksen Associates.

SCHEIDEL, J.M. (1976). Speech communication and Human interaction. Glenview, Il, Scott. Foresman.

SCHNEIDERMAN, B. (1980). Software Psychology, Winthorp Publishers.

THOMSETT, R. (1990). Effective Project Teams. American Programmer, juillet/août, 25-35.

TJAHJONO, D. (1994). Evaluating the Cost-Effectiveness of Formal Technical Review Factors. Thèse de doctorat, mars.

TJAHJONO, D. (1996). Exploring the Effectiveness of Formal Technical Review Factors with CSRS: a Collaborative Software Review System, ICS-TR-95-08, University of Hawaii at Manoa.

TUCKEY, J.W. (1977). Exploratory data analysis. Reading, MA : Addison-Wesley.

VAN SCOY, R.L. (1992). Software Development Risk: Opportunity, Not Problem . Software Engineering Institute, Carnegie Mellon University.

VOTTA, L.G. (1993). Does Every Inspection Need a Meeting ?. Proceedings of the ACM SIGSOFT 1993 Symposium on Foundations of Software Engineering, décembre.

WELLER, E.F. (1993). Lessons from Three Years of Inspection Data. IEEE Software, 38-45, septembre.

WIENBERG, G.M. (1971). The Psychology of Computer Programming, Van Norstrand Reinhold Company.

WOOLNOUGH, R. (1991). TI drives at Euro autos; makes 'PACT' to win microcontroller business., Electronic Engineering Times, Août .

YOURDON, E. (1989). Structured Walkthroughs, Prentice-Hall, 4è édition.

# ANNEXE A

## Protocole type d'une réunion de révision technique

| P | INTERVENTION | # | NOTE | CODE |
|---|---|---|---|---|
| B | Ok, walkthrough sur quoi? | 1 | | GER/RT |
| B | Réunion du 7 avril. | 2 | | AUT |
| M | 7 avril, y neige dehors. | 3 | | AUT |
| B | Y neige dehors. | 4 | | GER/RT |
| B | Ok, on commence avec que, quoi? | 5 | | GER/RT |
| M | L'analyse des gestions des messages. | 6 | | GER/RT |
| B | L'analyse des gestions des messages. | 7 | | GER/RT |
| B | C'est partie. | 8 | | GER/RT |
| M | Page 5. | 9 | | GER/RT |
| B | Attends, faut regarder pour voir si y a des/ | 10 | | GER/RT |
| B | Ok, là, ... comment faire ça. | 11 | | AUT |
| C | Ok. | 12 | | AUT |
| B | Là, on parle tu/ | 13 | | GER/RT |
| M | J'peux ben. | 14 | | GER/RT |
| B | Tu notes tu les erreurs toé là. | 15 | | GER/RT |
| M | Ouais, ouais j'vas les noter les erreurs là moé. | 16 | | GER/RT |
| B | Ok. | 17 | | GER/RT |
| M | Ok, un mécanisme des gestions des messages est nécessaire pour transmettre toute l'information concernant le déroulement normal de la simulation. La tâche consiste à implanter le gestionnaire des messages capable de supporter plusieurs langues ainsi que plusieurs types de messages!!! et mess, | 18 | | INTRO/SOLb |
| M | des types de données créées pour l'implantation du gestionnaire de messages. | 19 | | INTRO/SOLc |
| M | D'abord langues des messages, elangue, elangue des messages acceptera les valeurs suivantes. Des langues français ou des langues anglais pour valeurs pour aller chercher au besoin son désir de supporter d'autres langues/ | 20 | | INTRO/SOLd |
| B | Eh/ | 21 | NOMENCL. VAR. | EVAL-/INT20/CRIT.Fa |
| M | Oui/ | 22 | | EA |
| B | Une langue, c'est une langue. | 23 | NOMENCL. VAR. | EVAL-/INT20/CRIT.Fa |

| | | | | |
|---|---|---|---|---|
| M | Oui/ | 24 | | EA |
| B | Donc, on peut voir le deuxième mot s'accorder avec le premie. | 25 | NOMENCL. VAR. | EVAL-/INT20/CRIT.Fa |
| M | Ah, oui, langue française et langue anglaise | 26 | | DEV/INT20 |
| M | , ça va pour tout le monde ça. | 27 | | DEM/EVA/DEV26 |
| B | Next/ | 28 | | GER/RT |
| M | Les identificateurs de messages, elong.a, identificateurs de messages exécutera par défaut les 4 valeurs suivantes : le message de déverminage, le message d'information, les messages d'avertissements, les messages d'erreurs !!! Le message selon son type. Selon le type. On ajoutera une valeur pour chaque message qu'on voudra utiliser. | 29 | | INTRO/SOLea |
| M | Si on veut avoir un message eh, mémoire insuffisante, ben on ajoutera une valeur!!! | 30 | EXEMPLE | INFO/INT29 |
| B | Ok, c'est ça l'affaire ok, | 31 | | ACC/INF30 |
| B | c'est une question qu'on a plus tard mais comme tu viens de la dire là. | 32 | | GER/RT |
| | | 33 | | INTRO/SOLeb |
| B | Eh, à chaque fois qu'on va vouloir faire un message d'erreur/ | 34 | | HYP/INT33 |
| M | Euh/ | 35 | | EA |
| B | Faut faire un ENOM, faut faire, faut faire, faut faire un espèce d'exemple/ | 36 | | HYP/INT33 |
| M | Ouais, ouais/ | 37 | | ACC/HYP34 |
| B | C'est ça, pour, on est tu obliger d'avoir tout le temps eh, comment je pourrais dire ça, parce que là, on fait toujours, catégoriser notre message là, si on est capable là, t'sais capable de le catégoriser. | 38 | | DEM/INF/INT33 |
| M | Ben, ça va peut-être nous aider eh, ça va peut-être nous forcer, a avoir des messages qui sont des plus eh, sans être trop généraux, mais eh, réutiliser des messages qui existent. Pas avoir des messages qui disent la même affaire. | 39 | | INFO/INT33 |
| B | Mais si y a cent cinquante messages de même, y va y avoir cent cinquante similitudes/ | 40 | FONCTIONNALITE | EVAL-/INT33/CRIT.Ca |
| M | Ah, y va en avoir cent cinquante mais j'pense pas que ça tombe, j'pense pas que ça tombe/ | 41 | FONCTIONNALITE | JUSTIF/INT33/CRIT.Ca |

| | | | | |
|---|---|---|---|---|
| | | 42 | | INTRO/SOLec |
| C | Moi, j'pensais qu'on allait passer une string/ | 43 | | HYP/INT42 |
| M | C'est parce qu'à ce moment, ouais mais/ | 44 | | INC |
| C | Moi, j'pensais appeler, caller une fonction pis écrire ma string dans ma fonction/ | 45 | | HYP/INT42 |
| M | Ouais mais c'est pas ça/ | 46 | | REJ/HYP43/CRIT.C |
| C | T'as fait passer pis tu l'affiches à l'écran/ | 47 | | HYP/INT42 |
| M | Ouais mais c'est ça, c'est, si t'arrives, ce qui va arriver, c'est quand on va avoir les messages qui seront pas là, qui seront pas dûs avant nous autres.On va dire soit qu'on les met en anglais soit qu'on en français. | 48 | FONCTIONNALITE | JUSTIF/REJ46/CRIT.Ca |
| C | Ok, ouais/ | 49 | | ACC/REJ46 |
| B | Non, non mais/ | 50 | | EA |
| C | Ouais, faut qui soit déjà eh, non, excuse/ | 51 | | JUSTIF/ACC49 |
| M | On pourra pas être, on pourra pas être, on pourra pas être bilingue, faut que, pis eh/ | 52 | FONCTIONNALITE | JUSTIF/REJ46/CRIT.Ca |
| C | Faut qui soit déjà en quequ'part/ | 53 | | JUSTIF/ACC49 |
| | | 54 | | INTRO/SOLed |
| B | Pouquoi tu l'as mis là ma cent cinquante. | 55 | | DEM/JUS/INT54 |
| M | Pourquoi faire? | 56 | | EA |
| M | C'est parce que j'crois pas plus au eh, defined, le cent cinquante defined. Ça va revenir au même, mais ça au moins c'est parce que le compilateur va les checker. Tandis que les defined là, le compilateur les checke pas | 57 | FONCTIONNALITE | JUSTIF/INT54/CRIT.Ca |
| B | Ouais/ | 58 | | EA |
| M | Y fait juste remplacer/ | 59 | FONCTIONNALITE | JUSTIF/INT54/CRIT.Ca |
| C | Des message d'erreurs, y va peut-être en avoir cinquante différents t'sais/ | 60 | | HYP/INT54 |
| M | En/ | 61 | | EA |
| C | Des messages d'erreurs, y va peut-être en avoir cinquante différents/ | 62 | | HYP/INT54 |
| M | Ah, non, mais ça c'est juste un type, comme le type du message comme tel. | 63 | FONCTIONNALITE | REJ/HYP60/CRIT.Ca |
| C | Euh, euh. | 64 | | ACC/REJ63 |
| | | 65 | | INTRO/SOLee |

| | | | | |
|---|---|---|---|---|
| M | Ça c'est juste parce que ça me prend, ça me prend des, des champs, ok, les quatres qui sont là, là, c'est parce que ça me prend des chaînes qui sont déjà fixées au départ pour, pour eh, qui sont comme, fixes au message. | 66 | FONCTIONNALITE | JUSTIF/INT65/CRIT.C |
| M | Fait que ceux-là je les fixe, j'mets les quatres, j'mets comme les quatres premiers, les messages qu'on va ajouter, ça va être, ça va être les suivants.Fait que là on va en mettre eh, on mettra ce qu'on voudra là, un message erreur, mémoire insuffisante eh, pis j'sais pas trop quoi eh, bon, pis on les rajoutera. Mais j'penserais pas qui en ait cent cinquante. | 67 | FONCTIONNALITE | INFO/INT65/CRIT.Ca |
| T | Ah, mais comme le temps passe vite, c'est amusant non. | 68 | | AUT |
| B | Pourquoi faire dans le fond, si on pourrait les utiliser quand même. | 69 | | DEM/INF/INT65 |
| M | Ah/ | 70 | | EA |
| B | Ouais, ok, ouais de toute façon on n'a pas le choix. | 71 | FONCTIONNALITE | ACC/INT65/CRIT.Ca |
| M | On peux en faire là. | 72 | | INFO/INT65 |
| | | 73 | | INTRO/SOLef |
| B | J'pensais que tu voulais mettre des constantes là, | 74 | | HYP/INT73 |
| B | sauf que eh, c'est pas facile à gérer. | 75 | FONCTIONNALITE | REJ/HYP74/CRIT.Ca |
| C | Bon, ouais/ | 76 | | EA |
| T | C'est des constantes/ | 77 | | INFO/INT73 |
| M | C'est des constantes de toute façon. | 78 | | INFO/INT73 |
| T | C'est justement/ | 79 | | EA |
| B | Ouais c'est ça là, ben non. | 80 | | ACC/INF73 |
| T | Ben sinon, on n'a pas le choix là! | 81 | FONCTIONNALITE | JUSTIF/INT73/CRIT.Ca |
| M | On va mettre ça en quequ'part de toute façon. Sais pas, a moins que, voyez-vous un gros problème avec ça là?. | 82 | | DEM/EVA/INT73 |
| M | On peut trouver un autre façon de le faire là. Parce que j'trouvais que c'était une façon, c'était une façon simple là, elle aurait pas été méchante | 83 | | JUSTIF/INT73 |
| B | Non c'est correct/ | 84 | | ACC/INT73 |

| | | | | |
|---|---|---|---|---|
| C | Ben y avait eu plein, y avait eu des bonnes façons, si on avait été juste sur Windows mais là si on est sur Unix là, t'sais c'est plate là. | 85 | | JUSTIF/ACC84/SOLe |
| T | Ouais, mais c'est ça, j'vois pas le rapport. | 86 | | REJ/JUS85 |
| M | Parce que avant, avant le groupe avant l'autre mécanisme, c'était semblable à ça, sauf que c'était généré automatiquement/ | 87 | Analogie | JUSTIF/INT73 |
| B | Euh, euh/ | 88 | | ACC/JUS85 |
| M | C'est ça l'affaire/ | 89 | | INC |
| B | C'est ça/ | 90 | | EA |
| M | Pis là on sait pas automatiquement/ | 91 | | INC |
| | | 92 | | INTRO/SOLeg |
| C | Pis là faut regarder où les messages. | 93 | | DEM/INF/INT92 |
| M | C'est dans un fichier, ben c'est parce qu'on les écrit dans un fichier, mais eh, tu, tu eh, tu programmes en C, pis tu pognes le fichier/ | 94 | Analogie | INFO/INT92 |
| T | Qui génère une autre fonction. | 95 | | HYP/INT92 |
| M | Qui génère une autre fonction, pis les ENOM avec | 96 | Analogie | INFO/INT92 |
| C | Ok/ | 97 | | ACC/INF94 |
| M | Fait que la mémoire en/ | 98 | Analogie | INFO/INT92 |
| C | Y le met en mémoire en le loadant. | 99 | | HYP/INT92 |
| M | Ben, c'est une fonction, c'est une fonction avec un ENOM, pis c'est que tu links avec ça, tu compiles ça, pis tu links avec ça. | 100 | Analogie | INFO/INT92 |

# ANNEXE B

Codage des réunions utilisé pour la recherche

| P | # | CODE | RÉUNION | MOTS |
|---|---|---|---|---|
| B | 1 | GER/RT | re02 | 4 |
| B | 2 | AUT | | 4 |
| M | 3 | AUT | | 5 |
| B | 4 | GER/RT | | 3 |
| B | 5 | GER/RT | | 6 |
| M | 6 | GER/RT | | 5 |
| B | 7 | GER/RT | | 5 |
| B | 8 | GER/RT | | 2 |
| M | 9 | GER/RT | | 2 |
| B | 10 | GER/RT | | 9 |
| B | 11 | AUT | | 6 |
| C | 12 | AUT | | 1 |
| B | 13 | GER/RT | | 4 |
| M | 14 | GER/RT | | 2 |
| B | 15 | GER/RT | | 7 |
| M | 16 | GER/RT | | 9 |
| B | 17 | GER/RT | | 1 |
| M | 18 | INTRO/SOLb | | 42 |
| M | 19 | INTRO/SOLc | | 11 |
| M | 20 | INTRO/SOLd | | 33 |
| B | 21 | EVAL-/INT20/CRIT.Fa | | 1 |
| M | 22 | EA | | 1 |
| B | 23 | EVAL-/INT20/CRIT.Fa | | 5 |
| M | 24 | EA | | 1 |
| B | 25 | EVAL-/INT20/CRIT.Fa | | 11 |
| M | 26 | DEV/SOLd | | 7 |
| M | 27 | DEM/EVA/DEV26 | | 8 |
| B | 28 | GER/RT | | 1 |
| M | 29 | INTRO/SOLea | | 50 |
| M | 30 | INFO/INT29 | | 14 |
| B | 31 | ACCC/INF30 | | 7 |
| B | 32 | GER/RT | | 16 |
| | 33 | INTRO/SOLeb | | 0 |
| B | 34 | HYP/INT33 | | 11 |
| M | 35 | EA | | 1 |

| | | | | |
|---|---|---|---|---|
| B | 36 | HYP/INT33 | | 14 |
| M | 37 | ACC/HYP34 | | 2 |
| B | 38 | DEM/INF/INT33 | | 37 |
| M | 39 | INFO/INT33 | | 41 |
| B | 40 | EVAL-/INT33/CRIT.Ca | | 16 |
| M | 41 | JUSTIF/INT33/CRIT.Ca | | 18 |
| | 42 | INTRO/SOLec | | 0 |
| C | 43 | HYP/INT42 | | 7 |
| M | 44 | INC | | 7 |
| C | 45 | HYP/INT42 | | 13 |
| M | 46 | REJ/HYP43/CRIT.C | | 5 |
| C | 47 | HYP/INT42 | | 8 |
| M | 48 | JUSTIF/REJ46/CRIT.Ca | | 43 |
| C | 49 | ACC/REJ46 | | 2 |
| B | 50 | EA | | 3 |
| C | 51 | JUSTIF/ACC49 | | 8 |
| M | 52 | JUSTIF/REJ46/CRIT.Ca | | 17 |
| C | 53 | JUSTIF/ACC49 | | 6 |
| | 53 | INTRO/SOLed | | 0 |
| B | 54 | DEM/JUS/INT53 | | 8 |
| M | 55 | EA | | 2 |
| M | 56 | JUSTIF/INT53/CRIT.Ca | | 40 |
| B | 57 | EA | | 1 |
| M | 58 | JUSTIF/INT53/CRIT.Ca | | 4 |
| C | 59 | HYP/INT53 | | 11 |
| M | 60 | EA | | 1 |
| C | 61 | HYP/INT53 | | 10 |
| M | 62 | REJ/HYP59/CRIT.Ca | | 16 |
| C | 63 | ACC/REJ62 | | 2 |
| | 64 | INTRO/SOLee | | 0 |
| M | 65 | JUSTIF/INT64/CRIT.C | | 45 |
| M | 66 | INFO/INT64/CRIT.Ca | | 68 |
| T | 67 | AUT | | 10 |
| B | 68 | DEM/INF/INT64 | | 13 |
| M | 69 | EA | | 1 |
| B | 70 | ACC/INT64/CRIT.Ca | | 11 |
| M | 71 | INFO/INT64 | | 5 |

| | | | | |
|---|---|---|---|---|
| | 72 | INTRO/SOLef | | 0 |
| B | 73 | HYP/INT72 | | 8 |
| B | 74 | REJ/HYP73/CRIT.Ca | | 8 |
| C | 75 | EA | | 2 |
| T | 76 | INFO/INT72 | | 3 |
| M | 77 | INFO/INT72 | | 6 |
| T | 78 | EA | | 2 |
| B | 79 | ACC/INF72 | | 6 |
| T | 80 | JUSTIF/INT72/CRIT.Ca | | 8 |
| M | 81 | DEM/EVA/INT72 | | 22 |
| M | 82 | JUSTIF/INT72 | | 28 |
| B | 83 | ACC/INT72 | | 3 |
| C | 84 | JUSTIF/ACC83/SOLe | | 30 |
| T | 85 | REJ/JUS84 | | 8 |
| M | 86 | JUSTIF/INT72 | | 19 |
| B | 87 | ACC/JUS86 | | 2 |
| M | 88 | INC | | 3 |
| B | 89 | EA | | 2 |
| M | 90 | INC | | 6 |
| | 91 | INTRO/SOLeg | | 0 |
| C | 92 | DEM/INF/INT91 | | 7 |
| M | 93 | INFO/INT91 | | 29 |
| T | 94 | HYP/INT91 | | 5 |
| M | 95 | INFO/INT91 | | 9 |
| C | 96 | ACC/INF93 | | 1 |
| M | 97 | INFO/INT91 | | 5 |
| C | 98 | HYP/INT91 | | 8 |
| M | 99 | INFO/INT91 | | 25 |
| M | 100 | INFO/INT91 | | 24 |

# ANNEXE C

## Séquences de révision

| REU | A | E | CODE | SUJET | OBJET | CRIT | N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | SÉQUENCE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RE02 | M | 18 | INTRO/SOLb | SOLb | INT18 | | 0 | | | | | | | | 1 |
| RE02 | M | 19 | INTRO/SOLc | SOLc | INT19 | | 0 | | | | | | | | 2 |
| RE02 | M | 20 | INTRO/SOLd | SOLd | INT20 | | 0 | | | | | | | | 3 |
| RE02 | B | 21 | EVAL-/INT20/CRIT.Fa | INT20 | EVA21 | CRIT.Fa | 1 | | | | | | | | |
| RE02 | M | 26 | DEV/INT20 | INT20 | DEV26 | | 1 | | | | | | | | |
| RE02 | M | 27 | DEM/EVA/DEV26 | DEV26 | EVA27 | | 2 | | | | | | | | |
| RE02 | M | 29 | INTRO/SOLea | SOLea | INT29 | | 0 | | | | | | | | 4 |
| RE02 | M | 30 | INFO/INT29 | INT29 | INF30 | | 1 | | | | | | | | |
| RE02 | B | 31 | ACC/INF30 | INF30 | ACC31 | | 2 | | | | | | | | |
| RE02 | | 33 | INTRO/SOLeb | SOLeb | INT33 | | 0 | | | | | | | | 5 |
| RE02 | B | 34 | HYP/INT33 | INT33 | HYP34 | | 1 | | | | | | | | |
| RE02 | M | 37 | ACC/HYP34 | HYP34 | ACC37 | | 2 | | | | | | | | |
| RE02 | B | 38 | DEM/INF/INT33 | INT33 | INF38 | | 1 | | | | | | | | |
| RE02 | M | 39 | INFO/INT33 | INT33 | INF39 | | 1 | | | | | | | | |
| RE02 | B | 40 | EVAL-/INT33/CRIT.Ca | INT33 | EVA40 | CRIT.Ca | 1 | | | | | | | | |
| RE02 | M | 41 | JUSTIF/INT33/CRIT.Ca | INT33 | JUS41 | CRIT.Ca | 1 | | | | | | | | |
| RE02 | | 42 | INTRO/SOLec | SOLec | INT42 | | 0 | | | | | | | | 6 |
| RE02 | C | 43 | HYP/INT42 | INT42 | HYP43 | | 1 | | | | | | | | |
| RE02 | M | 46 | REJ/HYP43/CRIT.C | HYP43 | REJ46 | CRIT.C | 2 | | | | | | | | |
| RE02 | C | 47 | HYP/INT42 | INT42 | HYP47 | | 1 | | | | | | | | |
| RE02 | M | 48 | JUSTIF/REJ46/CRIT.Ca | REJ46 | JUS48 | CRIT.Ca | 3 | | | | | | | | |
| RE02 | C | 49 | ACC/REJ46 | REJ46 | ACC49 | | 3 | | | | | | | | |
| RE02 | C | 51 | JUSTIF/ACC49 | ACC49 | JUS51 | | 4 | | | | | | | | |
| RE02 | M | 52 | JUSTIF/REJ46/CRIT.Ca | REJ46 | JUS52 | CRIT.Ca | 3 | | | | | | | | |
| RE02 | C | 53 | JUSTIF/ACC49 | ACC49 | JUS53 | | 4 | | | | | | | | |
| RE02 | | 53 | INTRO/SOLed | SOLed | INT53 | | 0 | | | | | | | | 7 |
| RE02 | B | 54 | DEM/JUS/INT53 | INT53 | JUS54 | | 1 | | | | | | | | |
| RE02 | M | 56 | JUSTIF/INT53/CRIT.Ca | INT53 | JUS56 | CRIT.Ca | 1 | | | | | | | | |
| RE02 | C | 59 | HYP/INT53 | INT53 | HYP59 | | 1 | | | | | | | | |
| RE02 | M | 62 | REJ/HYP59/CRIT.Ca | HYP59 | REJ62 | CRIT.Ca | 2 | | | | | | | | |
| RE02 | C | 63 | ACC/REJ62 | REJ62 | ACC63 | | 3 | | | | | | | | |
| RE02 | | 64 | INTRO/SOLee | SOLee | INT64 | | 0 | | | | | | | | 8 |
| RE02 | M | 65 | JUSTIF/INT64/CRIT.C | INT64 | JUS65 | CRIT.C | 1 | | | | | | | | |
| RE02 | M | 66 | INFO/INT64/CRIT.Ca | INT64 | INF66 | CRIT.Ca | 1 | | | | | | | | |
| RE02 | B | 68 | DEM/INF/INT64 | INT64 | INF68 | | 1 | | | | | | | | |
| RE02 | B | 70 | EVAL+/INT64/CRIT.Ca | INT64 | EVA70 | CRIT.Ca | 1 | | | | | | | | |
| RE02 | M | 71 | INFO/INT64 | INT64 | INF71 | | 1 | | | | | | | | |
| RE02 | | 72 | INTRO/SOLef | SOLef | INT72 | | 0 | | | | | | | | 9 |
| RE02 | B | 73 | HYP/INT72 | INT72 | HYP73 | | 1 | | | | | | | | |
| RE02 | B | 74 | REJ/HYP73/CRIT.Ca | HYP73 | REJ74 | CRIT.Ca | 2 | | | | | | | | |
| RE02 | T | 76 | INFO/INT72 | INT72 | INF76 | | 1 | | | | | | | | |
| RE02 | M | 77 | INFO/INT72 | INT72 | INF77 | | 1 | | | | | | | | |
| RE02 | B | 79 | ACC/INF76 | INF76 | ACC79 | | 2 | | | | | | | | |
| RE02 | T | 80 | JUSTIF/INT72/CRIT.Ca | INT72 | JUS80 | CRIT.Ca | 1 | | | | | | | | |
| RE02 | M | 81 | DEM/EVA/INT72 | INT72 | EVA81 | | 1 | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| RE02 | M | 82 | JUSTIF/INT72 | INT72 | JUS82 | | 1 | |
| RE02 | B | 83 | EVAL+/INT72 | INT72 | EVA83 | | 1 | |
| RE02 | C | 84 | JUSTIF/EVA83 | EVA83 | JUS84 | | 2 | |
| RE02 | T | 85 | REJ/JUS84 | JUS84 | REJ85 | | 3 | |
| RE02 | M | 86 | JUSTIF/INT72 | INT72 | JUS86 | | 1 | |
| RE02 | B | 87 | ACC/JUS86 | JUS86 | ACC87 | | 2 | |
| RE02 | | 91 | INTRO/SOLeg | SOLeg | INT91 | | 0 | 10 |
| RE02 | C | 92 | DEM/INF/INT91 | INT91 | INF92 | | 1 | |
| RE02 | M | 93 | INFO/INT91 | INT91 | INF93 | | 1 | |
| RE02 | T | 94 | HYP/INT91 | INT91 | HYP94 | | 1 | |
| RE02 | M | 95 | INFO/INT91 | INT91 | INF95 | | 1 | |
| RE02 | C | 96 | ACC/INF93 | INF93 | ACC96 | | 2 | |
| RE02 | M | 97 | INFO/INT91 | INT91 | INF97 | | 1 | |
| RE02 | C | 98 | HYP/INT91 | INT91 | HYP98 | | 1 | |
| RE02 | M | 99 | INFO/INT91 | INT91 | INF99 | | 1 | |
| RE02 | C | 101 | HYP/INT91 | INT91 | HYP101 | | 1 | |
| RE02 | M | 102 | ACC/HYP101 | HYP101 | ACC102 | | 2 | |
| RE02 | B | 103 | ACC/HYP101 | HYP101 | ACC103 | | 2 | |
| RE02 | M | 104 | JUSTIF/INT91 | INT91 | JUS104 | | 1 | |
| RE02 | C | 105 | EVAL-/INT91/CRIT.Ca | INT91 | EVA105 | CRIT.Ca | 1 | |
| RE02 | C | 106 | DEM/INF/INT91 | INT91 | INF106 | | 1 | |
| RE02 | M | 107 | DEV/INT91/CRIT.Ca | INT91 | DEV107 | CRIT.Ca | 1 | |
| RE02 | B | 108 | HYP/DEV107 | DEV107 | HYP108 | | 2 | |
| RE02 | M | 109 | ACC/HYP108 | HYP108 | ACC109 | | 3 | |

# ANNEXE D

Résultats détaillés du LSA et MLL

**LSA pour les interventions brutes**

| INTERVENTION SUIVANTE \ INTERVENTION PRÉCÉDENTE | INT | DEV | EVA | HYP | INF | JUS | NIV |
|---|---|---|---|---|---|---|---|
| INT | -5.78 | -3.76 | -2.21 | -4.33 | -3.88 | -3.37 | 0.61 |
| DEV | 1.80 | 4.64 | 3.57 | -2.05 | -2.43 | -2.32 | 12.48 |
| EVA | 4.28 | -0.71 | 5.31 | -4.15 | -0.98 | -2.33 | 3.46 |
| HYP | 1.17 | -1.44 | -1.16 | 3.16 | 11.72 | -0.18 | 14.73 |
| INF | -0.70 | -4.15 | -2.25 | -0.18 | 9.13 | -3.05 | 6.73 |
| JUS | 0.33 | 1.46 | 8.98 | 0.73 | 0.30 | 6.76 | 19.10 |
| NIV | -10.52 | -4.26 | -6.02 | -4.63 | -6.72 | -5.02 | 7.72 |

**LSA pour les interventions regroupées en échanges**

| INTERVENTION SUIVANTE \ INTERVENTION PRÉCÉDENTE | INT | DEV | EVA | HYP | INF | JUS | NIV |
|---|---|---|---|---|---|---|---|
| INT | -7.549 | -4.69 | -3.6 | -5.07 | -4.54 | -3.91 | -2.01 |
| DEV | 3.388 | -4.25 | 1.77 | -1.62 | -1.59 | -1.94 | 12.46 |
| EVA | 7.986 | 0.74 | -5.12 | -3.76 | 0.541 | -1.6 | 5.577 |
| HYP | 1.358 | -1.35 | -1.05 | -3.23 | 10.44 | -0.14 | 11.93 |
| INF | 3.255 | -2.78 | 0.32 | 2.51 | -4.38 | -2.04 | 12.3 |
| JUS | 1.285 | 2.24 | 10.3 | 1.39 | 0.941 | -2.21 | 18.32 |
| NIV | -8.56 | 1.78 | -0.26 | -0.4 | -1.49 | -2.14 | -8.42 |

**LSA pour les échanges avec regroupement des niveaux**

| INTERVENTION SUIVANTE \ INTERVENTION PRÉCÉDENTE | INT | DEV | EVA | HYP | INF | JUS |
|---|---|---|---|---|---|---|
| INT | -10.29 | -4.31 | -4.62 | -5.7 | -5.7 | -4.65 |
| DEV | 4.547 | -4.12 | 9.1 | 0.34 | -0.18 | -1.64 |
| EVA | 7.757 | 2.83 | -5.47 | -3.73 | 1.126 | -1.57 |
| HYP | 1.923 | 0.03 | 0.49 | -3.16 | 13.53 | 0.33 |
| INF | 3.103 | -1.68 | 1.16 | 5.01 | -4.59 | -0.34 |
| JUS | 1.02 | 3.73 | 12.7 | 3.84 | 2.63 | -2.32 |

**LSA pour les échanges avec regroupement des justifications**

| INTERVENTION SUIVANTE \ INTERVENTION PRÉCÉDENTE | INT | DEV | EVA | HYP | INF |
|---|---|---|---|---|---|
| INT | -11.25 | -4.74 | -4.72 | -5.98 | -6.03 |
| DEV | 4.691 | -4.24 | 9.34 | 0.43 | -0.37 |
| EVA | 8.266 | 3.48 | -5.58 | -3.69 | 1.399 |
| HYP | 2.651 | 0.14 | 1.76 | -3.3 | 13.11 |
| INF | 3.125 | -1.62 | 1.9 | 6.04 | -4.74 |

**LSA avec les échanges E_D et H_I**

| INTERVENTION SUIVANTE \ INTERVENTION PRÉCÉDENTE | INT | DEV | EVA | H_I | E_D |
|---|---|---|---|---|---|
| INT | -15.05 | -4.64 | -6.24 | -7.26 | -5.45 |
| DEV | 19.44 | -1.42 | -2.03 | 24.9 | -1.83 |
| EVA | 19.96 | -2.1 | -3 | 10.4 | -2.7 |
| H_I | 7.141 | -1.68 | 1.69 | -5.25 | -0.31 |
| E_D | 28.82 | -1.87 | -2.66 | 8.55 | -2.39 |

## Tables marginales obtenues à partir

## de l'analyse log-linéaire des données

### 1. Responsabilités dans le projet en fonction de l'intérêt pour la solution.

Fréquences théoriques d'occurrence des interventions traitant de la forme ou du contenu.

Critere: Forme

| SOLUTION | PROJET CHEF | PROJET EXPERT | PROJET DEVEL | Total |
|---|---|---|---|---|
| INDIRECT | 140 | 97 | 36 | 273 |
| DIRECT | 17 | 11 | 64 | 92 |
| Total | 157 | 108 | 100 | 365 |

Critere: Contenu

| SOLUTION | PROJET CHEF | PROJET EXPERT | PROJET DEVEL | Total |
|---|---|---|---|---|
| INDIRECT | 155 | 64 | 86 | 305 |
| DIRECT | 95 | 25 | 176 | 296 |
| Total | 250 | 89 | 262 | 601 |

### 2. Nature des interventions en fonction du rôle dans la réunion.

Fréquences théoriques d'occurrence des interventions effectuées par les différentes responsabilités.

Projet: Expert

| GENERIQ | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| AUTEUR | 2 | 0 | 4 | 0 | 0 | 6 |
| REVISEUR | 56 | 64 | 8 | 39 | 24 | 191 |
| Total | 58 | 64 | 12 | 39 | 24 | 197 |

Projet: Chef

| GENERIQ | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| AUTEUR | 13 | 10 | 25 | 52 | 13 | 113 |
| REVISEUR | 113 | 81 | 20 | 39 | 41 | 294 |
| Total | 126 | 91 | 45 | 91 | 54 | 407 |

Projet: Devel

| GENERIQ | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| AUTEUR | 31 | 29 | 24 | 50 | 13 | 147 |
| REVISEUR | 31 | 32 | 26 | 56 | 70 | 215 |
| Total | 62 | 61 | 50 | 106 | 83 | 362 |

## 3. Nature des interventions en fonction des responsabilités dans le projet.

Fréquences théoriques d'occurrence des interventions effectuées selon l'intérêt envers la solution révisée.

Solution: Indirect

| PROJET | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| CHEF | 114 | 81 | 20 | 39 | 41 | 295 |
| EXPERT | 47 | 58 | 5 | 31 | 20 | 161 |
| DEVEL | 18 | 13 | 18 | 33 | 40 | 122 |
| Total | 179 | 152 | 43 | 103 | 101 | 578 |

Solution: Direct

| PROJET | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| CHEF | 12 | 10 | 25 | 52 | 13 | 112 |
| EXPERT | 11 | 6 | 7 | 8 | 4 | 36 |
| DEVEL | 44 | 48 | 32 | 73 | 43 | 240 |
| Total | 67 | 64 | 64 | 133 | 60 | 388 |

## 4. Nature des interventions en fonction du rôle dans la réunion.

Fréquences théoriques d'occurrence des interventions traitant de la forme ou du contenu.

Critere: Contenu

| GENERIQ | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| AUTEUR | 38 | 25 | 34 | 89 | 26 | 212 |
| REVISEUR | 77 | 71 | 33 | 105 | 103 | 389 |
| Total | 115 | 96 | 67 | 194 | 129 | 601 |

Critere: Forme

| GENERIQ | INTERV EVAL | INTERV DEV | INTERV JUS | INTERV INFO | INTERV HYP | Total |
|---|---|---|---|---|---|---|
| AUTEUR | 8 | 14 | 19 | 13 | 0 | 54 |
| REVISEUR | 123 | 106 | 21 | 29 | 32 | 311 |
| Total | 131 | 120 | 40 | 42 | 32 | 365 |

## 5. Rôles dans la réunion en fonction des responsabilités dans le projet.

Fréquences théoriques d'occurrence des interventions traitant de la forme ou du contenu.

Critere: Contenu

| PROJET | GENERIQ AUTEUR | GENERIQ REVISEUR | Total |
|---|---|---|---|
| CHEF | 96 | 154 | 250 |
| EXPERT | 2 | 87 | 89 |
| DEVEL | 114 | 148 | 262 |
| Total | 212 | 389 | 601 |

Critere: Forme

| PROJET | GENERIQ AUTEUR | GENERIQ REVISEUR | Total |
|---|---|---|---|
| CHEF | 17 | 140 | 157 |
| EXPERT | 4 | 104 | 108 |
| DEVEL | 33 | 67 | 100 |
| Total | 54 | 311 | 365 |