Titre: Title:	Étude de la mesure de la fiabilité des analyses de protocole dans le cadre de réunions techniques en génie logiciel
Auteur: Author:	Simon Labelle
Date:	1999
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Labelle, S. (1999). Étude de la mesure de la fiabilité des analyses de protocole dans le cadre de réunions techniques en génie logiciel [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8673/

UNIVERSITÉ DE MONTRÉAL

ÉTUDE DE LA MESURE DE LA FIABILITÉ DES ANALYSES DE PROTOCOLE DANS LE CADRE DE RÉUNIONS TECHNIQUES EN GÉNIE LOGICIEL

SIMON LABELLE
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
SEPTEMBRE 1999

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ÉTUDE DE LA MESURE DE LA FIABILITÉ DES ANALYSES DE PROTOCOLE DANS LE CADRE DE RÉUNIONS TECHNIQUES EN GÉNIE LOGICIEL

présenté par: LABELLE, Simon

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

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

M. BOUDREAULT, Yves, M.Sc.A., président

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

M. PESANT, Gilles, Ph.D., membre

REMERCIEMENTS

J'aimerais d'abord remercier chaleureusement mon directeur de recherche Pierre N. Robillard pour ses précieux conseils, son inspiration et son appui constant, et pour m'avoir dirigé vers les avenues les plus riches et intéressantes du domaine. Je désire également lui exprimer ma reconnaissance pour les ressources matérielles qui ont été mises à ma disposition.

Je remercie profondément Patrick d'Astous, sans qui ce projet n'aurait pu exister, et David Gagnon, pour leur soutien, leur générosité et leurs recommandations tout au long du projet.

Merci également à ma famille et à mes amis pour leurs encouragements et leur oreille compréhensive quand je leur parlais de mon sujet de recherche.

Mais je veux surtout remercier Julie Léveillé pour sa patience sans bornes, ses idées, son soutien moral, sa compréhension, son affection et son amitié inestimable. Je t'aime.

Je tiens aussi à remercier les compagnies suivantes, pour leur support moral indirect: id Software, Activision, Microprose, EA, Nullsoft et Spinner Networks.

RÉSUMÉ

Le génie logiciel est une science qui a tenté de comprendre le processus de développement logiciel et qui cherche maintenant à comprendre l'aspect humain et son impact sur le développement. Pour ce faire, l'utilisation d'études empiriques permet de baser une théorie sur les pratiques existantes réelles.

Les résultats obtenus à partir d'expériences ont un poids et une crédibilité importante. Il est donc important de s'assurer qu'aucune erreur ne s'y est glissée. Une telle erreur peut être introduite facilement dans une expérience et avoir des conséquences importantes sur l'interprétation des résultats.

L'objectif de ce projet est d'étudier la mesure de la fiabilité d'analyses de protocole dans le cadre d'expériences empiriques sur les aspects humains en génie logiciel. Les différents concepts et problématiques concernant la fiabilité et la validité d'expériences qualitatives sont d'abord présentés. La difficulté venant de la nature qualitative des données est inhérente à l'observation d'aspects humains.

De plus, ce projet présente et met en pratique une démarche de validation en validant une méthode d'analyse. Cette méthode a servi dans une expérience d'observation de l'interaction qui survient lors de réunions de révision technique, dans le cadre d'un projet de développement logiciel. Les données ont été obtenues par l'enregistrement vidéo des réunions, leur transcription et leur catégorisation (ou codage). La démarche est analysée pour déterminer si la méthode d'analyse est valide et fiable.

L'objectivité de la méthode d'analyse est d'abord mesurée. Divers outils statistiques sont utilisés à cette fin, puisqu'aucun n'est parfaitement adapté au contexte purement nominal de la méthode. À l'aide de la répétition du codage d'une réunion, l'objectivité de cette

étape de la méthode est calculée. Dans la limite où il y aura toujours une certaine subjectivité inhérente au codage, les analyses effectuées montrent que l'étape est objective. Le reste de la méthode d'analyse étant automatisé, la mesure obtenue représente donc l'objectivité de la méthode en entier.

Un large ensemble de facteurs agit sur l'accord entre deux codeurs. Quelques-uns parmi ceux-ci sont étudiés dans ce projet, pour déterminer leur importance et leur relation avec l'accord. Il est montré que, pour la plupart des codes, les interventions de durées supérieures ont un accord plus important. Des mots-clés peuvent aussi avoir un impact sur le choix de certains codes, mais cet impact est modéré. L'importance de la formation des codeurs sur l'accord entre certains codes est ensuite montrée. Et la distribution des désaccords des codes est abordée, dans l'optique qu'un code peut être sémantiquement proche d'un autre et que cette proximité peut influencer l'accord entre codeurs.

Ce projet examine aussi l'impact qu'ont les désaccords sur les résultats des analyses faites sur les données. Les analyses statiques révèlent d'abord une très grande similitude entre les deux versions du codage. Une analyse séquentielle est également effectuée pour identifier des patrons d'interactions. Ces analyses requièrent une quantité de données importante qui n'est pas disponible directement puisqu'une seule des sept réunions originalement utilisées a été codée plus d'une fois. Une simulation des données, basée sur les différences observées dans la réunion recodée, est donc obtenue. Elle est ensuite utilisée pour comparer la structure des échanges selon les deux versions de codes. Les similitudes observées sont grandes dans cette partie de l'analyse.

Il est donc montré dans ce projet que la validation d'une étude empirique sur les aspects humains est possible. La mesure de la fiabilité sur des mesures nominales est problématique, mais il est possible d'obtenir de bonnes estimations statistiques de l'objectivité. Et l'étude de l'impact révèle que les résultats sont stables, validant ainsi la méthode d'analyse.

La démarche utilisée est une contribution importante du projet. Elle peut être adaptée aux expériences de même nature où l'imprécision réside dans une seule étape du processus, ou aux expériences impliquant des mesures plus précises puisque des outils statistiques plus puissants sont alors disponibles.

Ce projet contribue au génie logiciel en y introduisant les concepts nécessaires à une validation expérimentale sérieuse.

ABSTRACT

Software engineering is a science which has tried to understand the software development process, and which now needs to understand the human aspect involved and its impact on development. To this end, empirical studies are used to build theories based on existing practices.

Results obtained from experimentation have high credibility. It is thus important to make sure they are error-free. An error can easily be introduced in an experiment, and have far reaching consequences on the interpretation of results.

The general objective of this project is to study the measure of reliability of protocol analyses in qualitative empirical studies of software engineering. The different concepts and problems related to the reliability and validity of qualitative studies are first presented. The difficulty coming from the qualitative nature of the collected data is inherent in studies involving observation of humans.

Then, this thesis presents and uses a validation procedure by validating an empirical approach to study human aspects. This approach was used in a study concerning the interaction found in technical review meetings, during a software development project. The data was obtained by videotaping the meetings, transcribing them and coding them. The procedure is used to determine if the approach is valid and reliable.

The objectivity of the approach is first measured. Different statistical tools are used in this step, since no tool is perfectly adapted to the purely nominal measures in the approach. By repeating the coding of a meeting, the objectivity of this step of the approach can be measured. The analyses show that the step is objective, within the limit that a certain

amount of subjectivity will always be present. The rest of the approach being automated, the measure obtained thus represents the objectivity of the whole approach.

A large group of factors influence the agreement between two observers. A few of those are studied in this project, to determine their importance and relation with inter-observer agreement. It is shown that for most codes, longer interventions have a higher observer agreement rate. Key words are also shown to have an impact on the choice of certain codes, but this impact is moderate. The impact of observer training is also analyzed. And the distribution of disagreements is tackled, with the notion that a code can be semantically close to another one, and that this proximity can influence agreement between observers.

In this project, the impact of disagreements on analysis results is also examined. Statistical analysis reveals a great similarity between the two versions of the coding. A sequential analysis is also done to identify exchange patterns. These analyses require a large quantity of data that is not directly available, since only one meeting was coded more than once. A simulation of data is thus obtained, based on differences observed in the meeting coded twice. The new set of data is then used to compare the structure of exchanges found by each version of the coding. The similarities observed in this part of the analysis are important.

It is thus shown in this project that validating an empirical study on human aspects of software engineering is possible. The measure of reliability of nominal measures is problematic, but good statistical estimations of objectivity are possible. And the impact study reveals that the results are stable, hence validating the empirical approach.

The procedure used is an important contribution of this project. It can be adapted to any experience of a similar nature, where imprecision lies in a single step of the experimental

process, or where more precise measures are used, since more powerful statistical tools are available.

This project contributes to software engineering by introducing to it the necessary concepts needed in a serious experimental validation.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ANNEXES

INTRODUCTION

Comme dans toute autre discipline scientifique, l'avancement en génie logiciel se fait par l'expérimentation. Une étude empirique permet d'approfondir les connaissances en partant de l'observation directe d'un phénomène. L'empirisme est le lien entre une théorie et la réalité pratique.

Les expériences sérieuses ne sont pas très nombreuses en génie logiciel, la plupart des articles scientifiques ne font pas d'expérimentation, ou relatent simplement des assertions ou des études de cas (Zelkowitz et Wallace, 1998). Moins de 5 % des articles incluent de la répétition expérimentale. Une telle situation signifie que la majorité des connaissances dans le domaine est basée sur des anecdotes ou sur des opinions (Fenton et al. 1994), et entraîne que plusieurs praticiens croient que le monde académique ne peut rien améliorer au développement de logiciels.

De plus, le grand nombre d'expériences ayant des erreurs de design et le coût important de telles erreurs pour l'industrie baissent le niveau de confiance populaire envers l'empirisme en génie logiciel. Par exemple (tiré de Fenton et al., 1994), l'utilisation de méthodes formelles a été étudiée pour tenter de déterminer si elle augmente la rentabilité et la qualité des logiciels ainsi développés. Une première expérience a étudié sérieusement un projet de 12 ans chez IBM et a montré une réduction des coûts de 9 % grâce à l'utilisation du Z. Cependant, les preuves quantifiables n'ont pas été inclues dans les publications. Une seconde expérience a montré que, même lorsque le Z est utilisé par des esprits très mathématiques, le nombre de défauts augmente avec son utilisation.
Une telle confusion entre des résultats ne peut s'expliquer que par une erreur quelconque dans le design d'au moins une des expériences. Pour éviter de telles erreurs, il est important de valider une expérience avant d'en publier les résultats, et de publier les

résultats de la validation. La validation expérimentale peut se faire de plusieurs manières, elle doit être adaptée au design de l'expérience.

Par ailleurs, l'étude du génie logiciel met graduellement l'accent sur l'observation du travail humain pour comprendre et améliorer les pratiques existantes (Hoc et al., 1990). Une telle orientation est nécessaire puisque le développement de logiciels est une activité essentiellement humaine. Mais étudier l'humain n'est pas aussi facile qu'étudier des lignes de code. Il est au moins aussi facile de faire une erreur dans une expérience observant le côté humain que dans une expérience plus technique puisque les mesures sont souvent moins précises. Ceci explique l'importance qui doit être mise sur la validation expérimentale dans ce genre d'étude. Malheureusement, tout comme il n'y a encore que peu d'études directement sur l'humain en génie logiciel, il y a encore moins d'expériences utilisant une validation solide et bien documentée.

Ce projet veut donc explorer la validation expérimentale d'une étude de l'humain en génie logiciel. Avec un nombre grandissant de telles études, il est important de bien connaître les fondements théoriques et les outils statistiques nécessaires pour montrer qu'une expérience est sans erreurs et fiable.

Pour ce faire, une méthode d'analyse développée récemment sera étudiée. Celle-ci mesure les interactions entre les participants d'une réunion de révision, en assignant un code à chaque intervention de la réunion. Une certaine répétition expérimentale sera faite pour tenter de cerner la fiabilité de la méthode. Ceci entraîne que plusieurs outils parmi les plus importants du domaine seront utilisés, et qu'une réunion sera codée par une autre personne. À partir de la comparaison des deux ensembles de codes, une certaine estimation de la fiabilité de la méthode d'analyse sera possible. Le codage est en effet la seule partie potentiellement subjective de la méthode d'analyse étudiée.

Plus précisément, le premier chapitre de ce mémoire présentera les notions théoriques exposant la problématique de la validation expérimentale dans les sciences sociales, particulièrement dans le contexte d'un système de mesure nominale comme celui utilisé dans la méthode d'analyse étudiée. Cette dernière sera présentée dans ses grandes lignes, tout comme son sujet de recherche, les réunions de revues. Les méthodologies utilisées pour le codage et pour la validation seront aussi présentées dans le premier chapitre.

La mesure des différences entre les deux ensembles de codes sera tentée au deuxième chapitre. Cette mesure sera une estimation de l'objectivité de la méthode d'analyse, composante importante dans l'étude de la fiabilité. La problématique de la mesure sur des données qualitatives sera pleinement visible dans l'exposition des différents outils statistiques disponibles pour mesurer les différences entre les deux codeurs.

Le troisième chapitre approfondira l'analyse de l'objectivité du chapitre 2 en tentant d'identifier certains facteurs pouvant affecter l'accord entre les codeurs. En particulier, la durée des interventions codées sera étudiée, ainsi que la présence de certains mots-clés, l'apport de la formation d'un codeur et la dispersion des désaccords des codes.

Le dernier chapitre s'intéressera à l'impact des différences de codage sur les analyses effectuées sur les données. Certaines manipulations sur les données, nécessaires pour une partie de l'analyse, seront présentées, ainsi que les outils statistiques utilisés.

CHAPITRE I
ÉLÉMENTS THÉORIQUES

L'objectif de ce chapitre est de présenter les diverses théories nécessaires dans le cheminement de ce projet de recherche. Le fondement des études empiriques et leur application en génie logiciel sont d'abord présentés. La validation expérimentale, objectif de ce projet, est également discutée, et ses difficultés théoriques et pratiques sont approfondies. Les réunions de révision, élément du génie logiciel qui sont le sujet de l'expérience étudiée, sont présentées, ainsi que la méthode d'analyse que le projet cherche à valider. Enfin, les différentes méthodologies utilisées pour la validation sont exposées.

1.1 Études empiriques

L'avancement de la science se fait par le biais d'expériences objectives visant à confirmer ou infirmer une hypothèse. Les différentes étapes d'une expérience, résumées par Basili et al. (1986), sont les suivantes: la définition de l'expérience où la théorie est intégrée et les hypothèses formulées, la planification, le design de l'expérience, l'exécution, puis l'analyse et l'interprétation des résultats.

Ce processus, même mené rigoureusement, peut entraîner des erreurs. C'est pourquoi la répétition d'une expérience est nécessaire, pour valider les résultats obtenus par un autre chercheur (Brooks et al., 1994). L'idée ici est qu'une personne peut se tromper à l'une ou l'autre des étapes, faussant les résultats dans un certain sens. Mais plusieurs personnes indépendantes se trompant dans le même sens est plus improbable, et donc des résultats identiques confirment la fiabilité de l'expérience.

Une étude empirique consiste à baser l'expérimentation sur l'observation de la situation réelle à analyser. En général, l'utilisation de l'empirisme est nécessaire pour explorer un sujet peu étudié, ou pour analyser un aspect pour confirmer ou infirmer une hypothèse. La partie empirique d'une expérience permet de recueillir des données afin d'avoir une base solide pour ancrer la théorie.

Et, comme disait Richard Feynman: "The fundamental principle of science, the definition almost, is this: the sole test of the validity of any idea is experiment." (cité par Tichy et al., 1995).

1.1.1 Études empiriques en génie logiciel

Le génie logiciel est défini comme étant l'application pratique de connaissances scientifiques dans le design et le développement de logiciels et de sa documentation associée requise pour développer, opérer et maintenir ces logiciels (Boehm, 1976). Une autre définition est l'application d'une approche systématique, disciplinée et quantifiable au développement, à l'opération et à la maintenance de logiciels, c'est-à-dire l'application de l'ingénierie au logiciel (IEEE, 1990, terme *software engineering*, 1).

Selon Basili et al. (1986), les études empiriques en génie logiciel sont utilisées pour comprendre, contrôler et améliorer le processus de développement actuellement pratiqué. L'important pour le génie logiciel, d'après eux, est de mieux comprendre comment un logiciel est développé, quels effets peuvent avoir telle ou telle technique, et qu'est-ce qui doit le plus être amélioré. Une courte revue des diverses expériences faites aux débuts du génie logiciel a été assemblée par ces auteurs en 1986. Depuis, l'empirisme en génie logiciel a pris de l'expansion (Zelkowitz et Wallace, 1997).

Cependant, Tichy et al. (1995) remarquent qu'il y a toujours une trop faible proportion d'articles basant leur argumentation sur des données empiriques. Dans une étude de 50 articles choisis au hasard dans les principaux périodiques scientifiques en informatique, il a été déterminé qu'à peine 30 % des articles qui incluent une partie d'évaluation sont validés par des données expérimentales. Selon eux, la situation est alarmante et entraîne une perception de faiblesse du domaine. Plus d'empirisme est nécessaire en informatique, et Zelkowitz et Wallace (1997) confirment ceci pour le génie logiciel en particulier.

Myers (1978) résume la raison de la faiblesse simplement: les expériences contrôlées dans le domaine tendent à être longues et très coûteuses, ce qu'appuient plusieurs personnes du domaine (Brilliant et Knight, 1999). Mais cette affirmation est réfutée par Tichy (1998), qui observe que des expériences intéressantes peuvent avoir un petit budget, et que des expériences dispendieuses peuvent valoir beaucoup plus que leur coût. Perry et al (1994) remarquent d'autant plus que toute information empirique, même si l'échantillon est petit, est révélatrice.

Par ailleurs, le développement de logiciels est une activité essentiellement humaine, du design à la programmation (Brooks, 1975; Hoc et al., 1990; Weinberg, 1971). Il est donc logique d'étudier l'humain pour avancer les connaissances du génie logiciel (Curtis et al., 1988). Pour étudier l'humain en génie logiciel, des méthodes hybrides d'étude doivent être utilisés, pour bien représenter les côtés techniques et humains de la question (Shapiro, 1994). L'aspect humain du développement est un champ d'étude relativement récent, prenant son véritable départ avec la publication de Weinberg, en 1971 (Hoc et al., 1990).

Divers aspects de la psychologie humaine sont en relation dans le processus de génie logiciel: la cognition, la psychologie sociale et organisationnelle, ainsi que la science de la gestion (Herbsleb et al., 1995). Même que Robillard (1999) remarque qu'il serait possible de trouver un lien entre la qualité du logiciel, sa complexité et les métriques, si

l'opportunisme humain pouvait être mesuré. L'aspect cognitif correspond au côté individuel de la tâche de la programmation, alors que les aspects sociaux et organisationnels correspondent au travail d'équipe requis dans la majorité des projets de développement logiciel.

La majeure partie des études empiriques en génie logiciel se concentre sur l'étude de la programmation (Myers, 1978), de la compréhension de programmes (Truchon, 1996, résume certains travaux dans ce domaine) et des interfaces humain-ordinateur, champ d'étude faisant l'objet d'intenses travaux depuis les 20 dernières années. Mais la programmation n'est qu'un aspect du problème qui ignore la coopération à l'intérieur d'une équipe. L'étude du fonctionnement des équipes de développement n'est étudiée sérieusement que depuis quelques années. Ceci explique en partie la forte proportion de travaux exploratoires, par rapport aux expériences contrôlées en laboratoire, dans l'étude de cet aspect (Seaman, 1999a).

Curtis et Walz (1990) remarquent que l'organisation des programmeurs en équipe introduit un niveau à caractère social à la tâche cognitive de la programmation. Ils ajoutent qu'en général le développement de logiciels est, entre autres, un processus d'apprentissage, de négociation et de communication. Et les grands projets de développement requièrent une communication considérable, composante qui ne doit pas être négligée dans la recherche.

Certains espèrent élaborer des métriques aidant à prévoir le succès d'un projet à partir des communications entreprises par les membres du projet (Dutoit et Bruegge, 1998). Certains autres espèrent qu'une meilleure compréhension des processus cognitifs des programmeurs pourra améliorer les outils de support au développement (Von Mayrhauser et Vans, 1993), mais d'autres remarquent que cette approche pose problème puisqu'elle se base sur des études sur des étudiants et sur des études utilisant des petits programmes, ce qui est difficile à généraliser aux situations commerciales (Singer et al., 1997).

Potts (1993) a noté certaines limites des études empiriques en génie logiciel. Il commence par remarquer qu'une expérience n'est pas décisive sans qu'il y ait une certaine répétition. Les données quantitatives ne devraient pas être surévaluées par rapport aux données qualitatives; appliquer la méthode GQM peut aider à contourner ce problème (voir entre autres. Basili et Rombach, 1988). Et plusieurs académiciens provenant de l'informatique ne devraient pas tenter de faire une expérience sans consulter des professionnels de la psychologie ou de la sociologie. Mais Potts souligne l'importance de l'implication des informaticiens dans ce genre de recherche, pour apporter la vision globale et les connaissances techniques nécessaires au succès d'une expérience.

Bref, les études empiriques en génie logiciel sont une branche en expansion, qui sera appelée à jouer un rôle important dans les années à venir, à mesure que le domaine maturera. Les aspects humains sont un aspect important à étudier puisque c'est la clef de voûte du processus de développement logiciel. Et le fonctionnement des équipes de développement, qui commence à être étudiée sérieusement, en est une composante importante que l'on doit mieux comprendre pour espérer améliorer le processus de développement.

1.1.2 Modélisation et simulation

Une des approches utilisées pour améliorer la compréhension d'un phénomène est la conception de modèles représentant le phénomène. À l'aide de données empiriques, on identifie les caractéristiques d'intérêt et on construit un modèle. Meister (1990) définit un modèle simplement comme une représentation mathématique d'une situation étudiée. Le but de la conception d'un modèle, selon lui, est de prédire et contrôler les effets résultants de la manipulation des variables du modèle.

Un modèle est nécessaire lorsque la situation physique n'est pas disponible, ou encore lorsque la situation à étudier est trop complexe pour être analysée rapidement. Les aspects humains du développement de logiciels entrent définitivement dans cette deuxième catégorie.

La modélisation d'une situation, en plus d'être un outil important pour la compréhension d'une situation, est à la base de la simulation. Le modèle formalise uniquement les données recueillies, alors que la simulation permet d'explorer les différentes avenues du modèle pour prédire des situations futures (toujours selon Meister, 1990). Évidemment, l'extrapolation est limitée à un certain niveau de déviation des données modélisées, mais ceci ne restreint pas l'intérêt de simuler une situation potentielle. Par exemple, pour évaluer la performance d'une nouvelle situation, on la compare à des parcours simulés et on peut en déduire une idée générale de certains de ses paramètres.

La communication et la simulation commencent à être vues comme reliées et interdépendantes. Crookall et Saunders (1989) remarquent en effet que la communication est inhérente à la compréhension de la simulation, et vice versa. La simulation ne peut être utilisée pleinement sans s'attarder à la communication nécessaire à la participation.

En général, c'est couramment par la modélisation et la simulation que le lien peut être fait entre un ensemble de données empiriques et l'application des analyses conséquentes pour améliorer la pratique.

1.1.3 Validation de modèles empiriques d'analyse

Dans l'utilisation d'expériences scientifiques, la validation se retrouve à deux niveaux. Tout d'abord, on parle de validation de la théorie par le biais d'expériences. Les expériences permettent de tester des hypothèses en les confrontant aux données de la vie

réelle. Puisqu'elles peuvent confirmer ou contredire une théorie, les expériences ont un pouvoir extraordinaire. C'est ce qui justifie l'importance accordée à la validation des expériences elles-mêmes, le deuxième niveau où l'on retrouve la validation.

La validation d'une expérience, et donc des analyses et modèles qui en sont tirés, se fait sur deux fronts. Tout d'abord, une analyse théorique de l'expérience est réalisée pour détecter toute erreur dans le design de l'expérience. Ensuite, une répétition est faite de l'expérience sous des conditions aussi identiques que possible. Cette répétition doit être indépendante de l'originale, c'est-à-dire qu'elle doit être réalisée par d'autres personnes non-liées à l'expérience originale, et doit être complète, c'est-à-dire qu'elle doit partir de la cueillette de données pour aboutir aux mêmes analyses et conclusions finales que l'originale.

Wood et al. (1997) remarquent que la répétition expérimentale, en plus d'être nécessaire dans la validation de résultats, aide à améliorer la recherche. Plus précisément, la répétition d'une expérience peut généraliser un peu plus les résultats ou approfondir un aspect de l'analyse, en variant d'une manière contrôlée certains paramètres.

Les analyses statistiques tirées des répétitions expérimentales permettent d'estimer la validité et la fiabilité, concepts décrits plus loin. La validation d'une expérience se fait par l'étude de sa validité et de sa fiabilité.

La conception d'expériences est difficile et peut être sujette à l'erreur (Brooks et al., 1994). Pflegger (1995) résume les erreurs pouvant survenir dans une expérience. Il peut y avoir une erreur dans le design de l'expérience, rendant une partie ou la totalité de l'expérience invalide. Une erreur peut également se glisser dans les observations, c'est-à-dire la cueillette de données, ou dans les mesures faites sur ces observations, ce qui invalide cette mesure et toutes les analyses qui en dépendent. Aussi, il peut y avoir une variation dans les conditions entourant l'expérience qui pourrait influencer le rendement

de l'expérience. Par exemple, une étude sur la rapidité de programmation serait faussée si, dans le cours de l'expérience, une formation était suivie par certains programmeurs, les rendant plus rapides. D'ailleurs, Myers (1978) observe un aspect important dans les expériences mesurant des aspects humains: la variabilité des résultats est énorme. Ce sont toutes ces erreurs que l'analyse théorique et la répétition cherchent à identifier et mesurer.

1.1.3.1 Fiabilité

Un aspect-clé de la validation expérimentale est la fiabilité (*reliability*). Kerlinger (1973) et Curtis (1980) décrivent la fiabilité d'une mesure comme le degré de stabilité et de précision. On peut s'appuyer sur une mesure fiable. Kerlinger associe la fiabilité au niveau d'erreur de mesure de l'instrument de mesure utilisé. L'erreur peut être systématique ou aléatoire. L'erreur systématique entraîne un biais dans la mesure, alors que l'erreur aléatoire augmente la variance de la mesure, mais ne produit pas de biais précis. Enfin, remarque Kerlinger, la fiabilité est une condition nécessaire mais non suffisante de la valeur d'un résultat et de son interprétation.

Pour Hollenbeck (1978), la fiabilité d'observations peut être influencée par celui qui observe, ce qui est observé, le système de codage utilisé et le lien entre l'observateur et le système de codage. Pour estimer la fiabilité, on doit estimer la stabilité et la précision des mesures utilisées dans l'expérience. La précision est généralement fonction de la variance statistique des mesures, alors que la stabilité est déterminée par la corrélation entre des mesures d'un même objet à divers moments de l'expérience (Johnson et Bolstad, 1973). Une autre définition de la fiabilité est donnée par Dunn (1989), pour qui elle est représentée par la covariance entre les données observée et la vraie valeur théorique, divisée par la variance des données observées. Ces deux définitions supposent la répétition de la mesure dans des conditions similaires pour calculer la précision et la stabilité (Brown et al, 1968).

1.1.3.2 Validité

L'autre aspect-clé de la validation expérimentale est tout aussi important, mais plus difficile à cerner. La validité (*validity*) d'une mesure détermine à quel point la mesure est représentative et adéquate, toujours selon Kerlinger (1973) et Curtis (1980). Plus précisément, elle dit si le contenu de la mesure est représentatif du contenu de la propriété mesurée. Par exemple, une des faiblesses des tests de quotient intellectuel est que leur validité n'est pas démontrée: ils mesurent quelque chose, mais est-ce vraiment l'intelligence? Il est parfois très difficile de déterminer ce type de validité.

Ce type de validité n'est qu'une partie de la validité. Herbert et Attridge (1975) et Kerlinger (1973) définissent trois types de validité. La validité de contenu compare ce qui est vraiment mesuré à ce que l'expérimentateur dit mesurer. La validité de construction analyse la signification de la mesure et l'explication théorique derrière la variance observée de cette mesure. La validité reliée au critère étudie les liens entre la mesure effectuée et un critère, c'est-à-dire la possibilité de prédire la valeur du critère à partir des résultats de la mesure. Il est souvent suffisant de mesurer la validité de contenu pour convaincre de la validité de l'expérience (Johnson et Bolstad, 1973).

Pour déterminer la validité d'une mesure, on doit essentiellement se contenter du jugement humain, plus particulièrement du jugement d'experts (Herbert et Attridge, 1975). Une analyse théorique du design de l'expérience est souvent la seule manière de juger cette validité. Une autre manière consiste à mesurer le même phénomène d'une manière la plus différente possible, et à comparer ensuite les résultats obtenus (Campbell et Fiske, 1959).

Notons que la validité d'une mesure a une définition plus stricte que la validation d'une expérience. Dans ce dernier cas, on cherche à savoir si l'expérience produit de bons résultats fiables et valides. Curtis (1980) fait d'ailleurs le lien suivant: les résultats d'une

expérience ne peuvent être plus valides que les mesures inclues. Dans ce projet, le terme *validation* est utilisé dans son sens large, alors que les termes *validité* et *fiabilité* sont utilisés dans leur sens strict.

Ces deux aspects-clés sont repris par Olson et al. (1994) qui les résume en disant que la méthode doit être objective et être une conséquence de notions théoriques. Bien sûr, la question de la validité scientifique ne se limite pas à ces aspects. Il existe d'ailleurs un cheminement philosophique complexe autour de la question. Seuls les deux aspects les plus importants ont été discutés ici. Mitroff et Turoff (1975) ainsi que Trochim (1999), font un rapide survol des principaux points de vue philosophiques existants.

Curtis (1980) ajoute enfin, et c'est le point à retenir de cette discussion, qu'une série d'expériences est plus riche et plus convaincante qu'une seule expérience, qu'elle permet d'améliorer les techniques expérimentales, et donc qu'elle augmente la fiabilité, en plus de l'estimer.

1.2 Problématique de recherche

Le cycle de développement de logiciel étant habituellement long et coûteux, il importe de l'optimiser et d'en augmenter la rentabilité. Puisque le développement repose essentiellement sur l'humain, celui-ci doit être étudié dans son environnement de travail pour comprendre précisément comment il fonctionne et qu'est-ce qui l'affecte, et affecte donc la rentabilité d'un projet de développement. Et puisque la grande majorité des logiciels sont le fruit d'un travail d'équipe, il appert que la collaboration est une facette importante dans ce cheminement. L'amélioration du processus de développement passe donc par une plus grande connaissance de ces aspects, principalement accessible par le biais d'études empiriques sûres.

L'objectif principal de ce projet de recherche consiste à étudier la fiabilité et la validité d'une méthode d'analyse empirique des aspects humains en génie logiciel. On vise une méthode objective et fiable pour assurer la valeur des études des aspects humains. C'est seulement avec des études dignes de confiance que l'analyse des aspects humains sera considérée par la communauté informatique. La validation d'une méthode d'analyse permet d'en cerner les faiblesses inhérentes et de mieux les comprendre.

Plus particulièrement, d'abord, la variabilité externe du mécanisme de recueil des données est mesurée. Une méthode de recherche se basant sur des données trop variables ne donne pas des résultats fiables. Il est donc important de connaître jusqu'à quel point les données sont objectives.

Puis, des facteurs affectant les variations des données sont identifiés. L'étude des aspects humains comporte intrinsèquement une part de jugement humain qu'on ne peut ignorer. Cependant, il est intéressant et pratique de bien saisir quels facteurs influencent le jugement à porter, pour savoir comment en minimiser l'importance.

Enfin, l'impact sur les résultats d'analyses du jugement compris dans les données est étudié. Ceci permet de mettre en contexte les variations mesurées entre les codeurs pour comprendre la relation entre celles-ci et les analyses qui en découlent. Une grande variation des données peut n'entraîner qu'une petite variation des résultats, ce qui laisse supposer que les données n'ont pas d'impact sur les résultats et qu'il y a une erreur dans le design expérimental. De même, une petite variation des données peut entraîner une grande variation des résultats, montrant la fragilité de la méthode et le peu de confiance qu'on peut y accorder.

Ainsi, il sera possible de juger de la robustesse d'une méthode d'analyse des aspects humains en génie logiciel, et d'en confirmer ou non les résultats. Ce projet vise donc à introduire au domaine du génie logiciel les différents concepts et outils nécessaires dans

la validation expérimentale, à fournir un exemple de validation et ultimement à montrer que l'étude des aspects humains est possible en génie logiciel d'une manière rigoureuse et fiable.

1.3 Revues et inspections

En général, un problème bien cerné est plus facile à étudier (Robillard, 1999). Une méthode de recherche vise donc l'étude d'un aspect en particulier du développement. La méthode analysée ici est utilisée dans l'analyse des activités collaboratrices lors de projets logiciels. Les activités collaboratrices se retrouvent principalement concentrées dans des réunions de design ou d'évaluation, et celles-ci sont d'excellents contextes d'étude des interactions (Bange, 1992). La méthode étudiée s'intéresse aux interactions qui surviennent durant les réunions de revues, qui cherchent à évaluer un document et à en identifier les défauts. Le contexte de l'étude sera donc précisé plus en détails.

Tel que mentionné plus haut, le développement de logiciels est un processus complexe et coûteux. Les différentes étapes du cycle de vie sont définies dans le but de maximiser la qualité du logiciel, et donc de minimiser le nombre de défauts tout en s'assurant une productivité maximale. Les défauts sont définies comme étant une étape, une définition ou un procédé incorrect (IEEE, 1990, définition des termes *error* et *fault*). Ils sont généralement très coûteux et lourds de conséquences pour les développeurs ou même pour les utilisateurs (Neumann, 1994). Pour identifier les défauts, il est possible de les chercher à chaque étape du processus, dans une optique de qualité totale, ou de laisser les tests faire cette recherche. Mais, le coût de la correction de défauts dans un logiciel augmente au fur et à mesure de l'avancement du logiciel (Gilb et Graham, 1993). Il est donc souhaitable d'identifier et de corriger les défauts le plus tôt possible dans le processus. Les inspections et revues visent à réduire le nombre de défauts tout au long du processus, et sont donc une activité importante du génie logiciel (Wheeler et al., 1996a).

1.3.1 Définitions

Chaque étape du cycle de vie du logiciel résulte en un ensemble de documents décrivant le produit logiciel. Ces documents sont ensuite passés en revue par une ou plusieurs personnes pour identifier les défauts dans le cadre d'inspections ou de revues.

Une revue est définie comme une réunion durant laquelle un produit est présenté à certaines personnes pour recueillir leurs commentaires ou leur approbation (IEEE, 1990, terme *review*). Une seconde définition provenant du IEEE (1988, terme *review*) est qu'une revue est une évaluation d'éléments du logiciel pour identifier des différences avec le produit prévu et pour recommander des améliorations, en suivant un processus formel. Divers processus sont définis dans la littérature. Chaque organisation adapte généralement ces définitions à ses besoins (Wheeler et al., 1996a).

Une inspection est un processus défini de manière précise et formelle. Concept introduit par Fagan (1976), une inspection comprend une séquence de cinq phases. Tout d'abord, le survol des éléments du logiciel à analyser et la préparation individuelle des participants. Puis, l'inspection comme telle durant laquelle une personne présente les éléments analysés et les autres participants posent des questions et soulèvent des erreurs. Aucune solution alternative ne doit être proposée. Enfin, les erreurs notées sont corrigées et les corrections sont revues, possiblement dans une autre inspection.

Une revue est définie d'une manière plus vaste, englobant les inspections, les walkthroughs, les audits et les révisions techniques (IEEE, 1988). Toutes ces formes de revues sont généralement des variations de l'inspection. Wheeler et al. (1996b) classifient un nombre important de telles variations. Ces auteurs remarquent aussi que le terme *inspection* est souvent utilisé dans son sens large, plus proche de la définition d'une revue que de la définition précise d'une inspection. Par exemple, Seaman et Basili (1998)

appellent *inspection* des revues ne respectant pas les spécifications précises d'une inspection.

1.3.2 État de l'art

L'utilisation des réunions de révision est répandue dans l'industrie du logiciel. Les avantages de ces réunions sont nombreux et alléchants pour les entreprises: réduction du temps de développement (Fagan, 1976), augmentation de la qualité, réduction des coûts de tests, réduction des coûts de maintenance, etc. (Gilb et Graham, 1993). Cependant, les coûts des revues sont aussi importants: le temps perdu par chaque participant durant les réunions, le coût de la production de documents associés, etc. Ces coûts font remarquer à Parnas et Weiss (1987) que la rentabilité des revues et inspections reste à prouver. Il est plus facile de comparer la rentabilité des diverses variations des inspections que de chiffrer la rentabilité d'une inspection.

Porter et al. (1996, 1998) ont étudié divers aspects des réunions de révision technique. Ils ont identifié six aspects qui changent dans les diverses variations des inspections: le nombre de participants, le nombre de sessions de réunions, la coordination entre les sessions, la technique de collection d'erreurs, la méthode de détection d'erreurs et l'utilisation de retour d'information (*feedback*) après la réunion. Selon eux, ces aspects sont ceux responsables des variations de coûts, et donc de rentabilité, des revues.

Par exemple, le nombre de participants idéal est sujet de longs et nombreux débats. Porter et al (1997) ont réalisé une expérience comparant l'efficacité de revues d'une, de deux ou de quatre personnes. Aucune différence notable ne fut observée entre les revues de deux ou de quatre personnes, mais celles-ci étaient plus efficaces que les revues d'un seul évaluateur. Une seconde étude suggère que quatre personnes sont deux fois plus productives que trois (Weller, 1993). Une autre analyse empirique, combinée à des

considérations théoriques et pragmatiques, suggère que la taille idéale d'une équipe de revue technique est de trois à cinq personnes (Bourgeois, 1996). Grady (1992) suggère plutôt de quatre à six personnes. Gilb (1998) propose de quatre à six ou de deux à quatre, en fonction de l'importance sur le nombre d'erreurs trouvées ou la rentabilité. Buck (1981) montre qu'il n'y a aucune différence d'efficacité entre une équipe de trois, quatre ou cinq personnes, alors que d'Astous (1999) suggère que le nombre n'est pas aussi important que la relation des participants avec le document révisé. Cet exemple illustre un des nombreux aspects influant la rentabilité d'une revue, mais montre aussi la tendance à chercher à optimiser chaque aspect pour améliorer l'efficacité et/ou la rentabilité des réunions de revues.

Par ailleurs, Porter et Johnson (1997) ont comparé deux expériences étudiant l'efficacité des réunions de révision. La première étude fut réalisée par Danu Tjahjono en 1996, et la seconde par Lawrence Votta en 1993, les deux comparant l'efficacité de la révision individuelle versus la révision en équipe dans le cadre d'une réunion. Les résultats combinés montrent une forte similitude des résultats individuels et collectifs pour ce qui est du nombre d'erreurs trouvées. Les erreurs trouvées en équipes ont moins tendance à être de fausses erreurs, mais la recherche individuelle d'erreurs étant nettement moins coûteuse, il n'est pas du tout clair quelle méthode est plus avantageuse. Toute étude des revues est donc nécessaire et très importante, en particulier dans la situation actuelle.

Tout ceci revient à dire que les revues et inspections sont une composante importante mais coûteuse du cycle de développement logiciel. Plusieurs tentent d'en optimiser le déroulement pour réduire les coûts qui y sont attachés. Pour améliorer les revues et les inspections, il est nécessaire de très bien comprendre leur fonctionnement. Mais ceux-ci sont des processus déterminés essentiellement par les humains qui y participent. Comprendre leur fonctionnement implique observer des réunions et modéliser les aspects étudiés. Et ainsi faire une expérience empirique valide et fiable. Ceci est donc la justification de l'étude empirique qui est analysée dans ce projet.

1.4 Méthode d'analyse

C'est par l'application d'une méthode d'analyse que l'on peut étudier un phénomène. Et c'est dans la méthode d'analyse qu'est déterminée la validité d'une expérience. Une courte discussion sur d'autres méthodes existantes analysant sensiblement le même problème sera présentée, suivie par la présentation de la méthode d'analyse étudiée dans ce projet.

1.4.1 Autres méthodes d'analyse

Quelques méthodes existent pour étudier les interactions contenues dans une réunion en informatique. Par exemple, une équipe dirigée par Gary Olson (Olson et al., 1992) a développé un système de codage et une méthode d'analyse (Olson et al., 1994) des réunions de design du logiciel. Ils enregistraient sur vidéo des réunions, les transcrivaient et codaient chaque intervention. Par la suite, ils réalisaient des analyses globales ainsi que des analyses séquentielles pour découvrir la structure inhérente aux réunions. Herbsleb et al (1995) ont étudié les mêmes données, mais en analysant l'aspect orienté-objet des discussions. Et Walz et al (1987, 1993) ont étudié le même type de situation, mais à un niveau plus global.

Par ailleurs, Carolyn Seaman (1996; Seaman et Basili, 1997 et 1998) a observé des réunions de revue, mais uniquement à un niveau global. Elle observait directement les réunions et prenait des notes personnelles en même temps qu'elle codait les interventions pour en retirer la durée globale de chaque type. Elle s'intéressait au lien existant entre la communication dans des réunions et la structure organisationnelle du projet de développement logiciel.

Ces deux expériences sont parmi les seules importantes à étudier les aspects humains en génie logiciel. Évidemment, il existe une longue tradition d'observations et d'analyses séquentielles en psychologie sociale et cognitive. Les premières expériences importantes remontent à Robert F. Bales (1951; Bales et Gerbrands, 1948), qui avait construit un appareil permettant de coder les interventions d'une réunion ainsi que l'ensemble de codes génériques permettant de décrire les interventions. Son système de codage fut repris à plusieurs reprises dans de nombreuses expériences. Cependant, le système de codage décrivant les interventions doit être adapté au contexte et à l'intérêt de la recherche (Rosenblum, 1978). Il est donc nécessaire, à chaque fois qu'une expérience est la première à explorer un domaine, qu'elle crée ou adapte un système de codage.

1.4.2 Méthode d'analyse étudiée

La méthode d'analyse étudiée a été conçue par Patrick d'Astous, Pierre N. Robillard, Françoise Détienne et Willemien Visser, dans le cadre du projet ACGL, une association du laboratoire RGL de l'École Polytechnique de Montréal et du groupe EIFFEL de l'INRIA. Elle est décrite dans les écrits de d'Astous (1999), d'Astous et al. (1998) et Robillard et al. (1998). La méthode est exposée ici, dans ses grandes lignes, pour permettre au lecteur de mieux comprendre ce qui est analysé.

Tout d'abord, la terminologie utilisée est présentée:

- Une *intervention* est l'élément de base d'une réunion. Elle est composée d'actes de langage (Winograd, 1987) émis par un seul interlocuteur, sans qu'un autre interlocuteur intervienne. Cette notion provient de la linguistique.
- Une *activité* est une description du contenu de l'intervention. Le schéma de codage définit 11 catégories décrivant les activités.

- Le *sujet* d'une intervention décrit l'objet sur lequel porte l'intervention. Un sujet peut être une intervention précédente, la solution étudiée de l'artefact, une solution alternative ou un critère.
- Un *critère* décrit l'aspect étudié par l'intervention, du côté de la forme ou du contenu. Il existe 19 critères définis dans le schéma de codage.
- Un *code* est une catégorisation d'une intervention relative à un système de codage. Un code peut être entendu comme la description complète de l'intervention, et ainsi comprendre un numéro d'identification de l'intervention, l'activité, le sujet et le critère. On parle aussi d'un code comme d'une activité.
- Un *système* (ou *schéma*) *de codage* est un ensemble cohérent et structuré de codes, élaboré sur des théories reliées, dans le but d'étudier une situation. Un système étudie un problème en particulier, et est donc adapté spécifiquement au problème. De plus, d'Astous (1999, p. 93), "Un schéma de codage est en fait l'élément de base sur lequel dépend tout le reste de la méthode d'analyse."
- Un *codage* est une assignation particulière d'un ensemble de codes à un ensemble d'interventions, où on associe un code par intervention. En quelque sorte, c'est donc une instanciation du système de codage. Aussi, le codage est défini comme l'activité de coder une réunion, c'est-à-dire la réalisation d'un codage.
- Une *séquence* est un ensemble d'interventions portant globalement sur un même sujet.
- Un *codeur* est une personne qui réalise un codage, c'est-à-dire une version du codage.
- Un *artefact* est le document évalué dans la réunion.
- Une *solution* est la partie étudiée de l'artefact, ou peut être une alternative à cette partie.

Chaque intervention est généralement codée en quatre parties: un identificateur numérotant l'intervention de manière unique pour référence future, l'activité caractérisant l'intervention, le sujet de l'intervention et, de manière optionnelle, le critère utilisé dans l'intervention. Les codes ressemblent donc à ceci: (ID) ACTIVITÉ/SUJET[/CRITÈRE].

Les critères ne sont qu'une caractérisation des arguments et sont peu étudiés. Il n'est donc pas jugé important de les présenter ici, sauf pour mentionner qu'ils sont divisés en deux groupes: les critères concernant le contenu (CRIT.C) et les critères sur la forme (CRIT.F).

Les interventions peuvent être classées parmi les 11 activités. Ces activités sont groupées en quatre ensembles:

- La demande (code DEM), où un intervenant exprime le désir d'une activité quelconque.
- La gestion (GES), où la réunion ou le projet sont discutés.
- L'introduction (INTRO ou INT), où un intervenant présente une nouvelle solution au groupe.
- La présentation (PRES), qui regroupe les activités générales de présentation:
    - L'acceptation (ACC), où le sujet est déclaré valable.
    - Le développement (DEV), où une nouvelle idée est expliquée en détails.
    - L'évaluation (EVA), où la valeur du sujet est jugée. Elle peut être positive (EVA+) ou négative (EVA-).
    - L'hypothèse (HYP), où l'intervenant donne ses croyances, ses suppositions en rapport avec le sujet.
    - L'information (INFO), où l'intervenant ajoute des renseignements sur le sujet au reste du groupe. L'intervenant est généralement convaincu de son information.
    - La justification (JUS), où l'intervenant juge qu'une argumentation est nécessaire pour appuyer un point de vue et expliquer le raisonnement derrière un jugement.
    - Le rejet (REJ), où le sujet est déclaré non valable et est écarté.

De plus, il existe un dernier code d'activité possible, autre (AUT), où l'intervention est complètement hors sujet (comme les blagues, par exemple). De plus, l'écoute active (EA)

et les interventions incompréhensibles (INC) sont regroupées dans la catégorie AUT pour les analyses.

L'identificateur des interventions permet ainsi à l'objet de référer à une intervention précédente. Par exemple, "EVAL+/INT95/CRIT.Ff" est une évaluation positive de la solution introduite à l'intervention 95, par rapport à un critère de forme concernant le type des données. Le tableau 1.1 présente un extrait d'une réunion, avec son codage original tel qu'utilisé dans les analyses du projet ACGL.

Tableau 1.1 Extrait de la réunion 02 avec le codage original associé

Auteur	Intervention	ID	Codage original
		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	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/INF77
T	Ben sinon, on n'a pas le choix là!	81	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	EVA+/INT73

Dans cet extrait, on remarque la présence d'une introduction implicite à l'intervention 73, c'est-à-dire que le sujet change sans qu'un intervenant le mentionne explicitement. Aussi, les différentes manières de coder sont bien représentées: l'intervention 76 ne contient que l'activité (une écoute active), qui ne réfère à aucun sujet intéressant, deux interventions

(75 et 81) spécifient le critère utilisé, et une demande d'évaluation est faite à l'intervention 82.

Le lecteur intéressé pourra se référer à la thèse de d'Astous (1999) pour une présentation plus précise du schéma de codage.

1.5 Analyse théorique du système de codage

Tel que mentionné plus haut, c'est par une analyse théorique que la validité d'un système de codage peut être évaluée pour déterminer si les codes représentent correctement ce qui doit être représenté et si le design expérimental ne fausse pas les résultats. Dans la méthode étudiée, c'est au niveau de la définition des codes, sur laquelle les codeurs se basent pour associer les codes aux interventions, qu'est la validité de contenu. Les codes et leur définition furent analysés par Françoise Détienne et Willemien Visser (1997), deux psychologues expertes dans le domaine, au cours du développement du système de codages. Elles ont identifié les liens entre les codes et les diverses théories de la psychologie cognitive et elles ont déterminé que le système est valide par rapport à son contenu. Les autres aspects du système de codage doivent maintenant être étudiés, car ils peuvent avoir un impact sur la fiabilité du système.

Chi (1997) a résumé les problèmes de codages qu'on cherche à évaluer et à éliminer. Il y a tout d'abord l'ambiguïté entre des composantes du système de codage. Pour chaque intervention, on doit associer un et un seul code (Herbert et Attridge, 1975). C'est ce qu'on appelle l'exclusivité mutuelle (*mutual exclusivity*). Idéalement, un seul code est approprié, mais en pratique il arrive que plusieurs codes semblent acceptables (Sackett, 1979).

Deux principales raisons expliquent ce phénomène: il peut y avoir des codes sémantiquement proches ou bien l'intervention à coder peut être vue de plusieurs manières. Par exemple, on comprend intuitivement que la différence entre une évaluation négative et un rejet n'est pas toujours aussi claire qu'on le voudrait et peut parfois entraîner des situations où le choix entre ces deux alternatives pose problème (voir l'intervention 75 du tableau 1.1). Mais il peut aussi arriver que l'intervention comme telle présente une ambiguïté inhérente, auquel cas des codes de nature très différente seraient acceptables. Par exemple, l'intervention 79 pourrait être codée comme une acceptation ou comme de l'écoute active. Ces deux codes sont très différents au niveau sémantique, mais l'intervention comme telle présente une ambiguïté que même le contexte ne peut pas complètement éliminer. Une étude de Patterson (1982) confirme d'ailleurs que l'accord entre deux codages pour des interventions simples est supérieur à celui entre deux codages pour des interventions plus complexes. Un système de codage idéal minimiserait les deux aspects de l'ambiguïté.

Une seconde source de problèmes dans un système de codage est le contexte. Un système idéal n'entraîne pas le besoin de se référer au contexte pour coder une intervention. Le contexte est l'ensemble des interventions précédant ou suivant directement l'intervention à coder. Tenir compte du contexte de chaque intervention lors du codage ralentit le codage et peut induire des erreurs (Chi, 1997). Cependant, tenir compte du contexte augmente la compréhension de la situation par le codeur. Dans le domaine étudié, c'est-à-dire l'analyse de transcriptions de conversations, l'analyse du contexte est habituellement très utile et fournit des informations précieuses qui ne devraient pas être ignorées. Par exemple, associer un code à une intervention ne comportant que "Oui" peut être difficile quand on ne sait pas si l'intervention précédente avait comme contenu "On va faire ceci et cela...", "Es-tu d'accord avec l'idée?" ou "On commence la réunion?". Le code associé à "Oui" pourrait être EA, ACC, GES ou même autre chose. Le besoin de se rabattre sur le contexte dans le codage d'une intervention n'est donc pas nécessairement une faiblesse du système de codage étudié.

Certains auteurs (Herbert et Attridge, 1975) ajoutent un autre critère de qualité d'un système de codage. Selon eux, l'exhaustivité du code est désirable, puisqu'elle assure que le système comprend la situation étudiée (ici, une réunion de révision technique) en en couvrant l'ensemble des possibilités.

Par ailleurs, Bakeman et Gottman (1986) ne trouvent pas essentiel que tous les codes soient mutuellement exclusifs. Des codes orthogonaux ne sont pas mutuellement exclusifs. Par exemple, un système de codage quelque peu étrange composé des codes "Manger", "Boire", "Homme" et "Femme" entraîne un problème quand un homme boit, c'est-à-dire quand deux codes complémentaires sont appropriés. Les auteurs remarquent que l'avantage d'avoir des codes dont la sémantique se recoupe est la diminution du nombre de codes requis. De plus, le codage se fait plus rapidement quand le nombre de codes est plus petit. Mais, au pire cas, tous les codes sont orthogonaux; il faut 2^n codes pour éliminer le recouvrement de n codes. Et, en général, une bonne analyse théorique peut éliminer une bonne partie des ambiguïtés dans un système de codage. Il est à noter qu'une partie de l'ambiguïté entre les codes peut être éliminée par une bonne formation des codeurs dans le schéma de codage.

Un aperçu rapide des codes semble indiquer que le schéma de codage est mutuellement exclusif. La définition précise de chacun des codes ne semble pas laisser beaucoup de place pour un quelconque recoupement entre les codes. Mais l'exclusivité mutuelle est assurée par le fait que les interventions sont découpées en fonction des changements d'activité dans le discours. Une intervention pouvant être codée de plus d'une façon sera divisée en autant d'interventions, aux endroits appropriés du discours. L'ambiguïté restante réside donc dans le contenu des interventions.

Trois conséquences problématiques peuvent survenir en rapport avec le codage. Il peut y avoir des différences entre les différents codeurs, i.e. inter-codeurs. Ce problème

d'objectivité et de précision est étudié dans ce projet à l'aide de comparaisons entre différentes versions de codages d'une même réunion. Il peut y avoir des variations du contenu de la réunion, entre son début et sa fin, affectant la stabilité des résultats. Ceci est débattu à la section 1.7.2. Cependant, il est aussi possible qu'il y ait des différences de codes dans un même codage, en fonction de la position dans la réunion. Plus précisément, un codeur peut changer son interprétation des interventions au courant du codage d'une réunion, résultant en des codes différents pour des interventions similaires. C'est ce qui est appelé la dérive de l'observateur. Il existe des techniques de prévention de telles erreurs (Johnson et Bolstad, 1973), mais pas de méthodes de validation qui permette de vérifier que le codage est pour toutes les interventions d'une réunion.

Le seul moyen de vérifier la dérive de l'observateur est d'avoir un codage qui soit considéré comme étant "idéal"; d'analyser la correspondance entre un nouveau codage et le codage idéal pour des portions de réunions, et de comparer les résultats des diverses portions de réunions. Ce moyen est évidemment fort peu pratique puisqu'il requiert une version "idéale" du codage, qu'il est impossible d'avoir avec certitude (Kazdin, 1977). Dans ce projet, une des hypothèses de travail posées est que la dérive de l'observateur est assez faible et a un impact négligeable sur les analyses subséquentes. Les résultats du projet sont donc dépendants de la valeur de cette hypothèse.

En somme, la validité théorique du système de codage ne semble pas être problématique. Il ne semble y avoir aucun des défauts les plus courants et importants, que ce soit dans la structure, les définitions ou les interrelations entre les codes. Avec la validité de contenu étudiée par Détienne et Visser (1997), il ne reste que la méthode et la fiabilité à étudier.

1.6 Méthodologie de codage

Avant de continuer la discussion sur la validation, il est important de présenter la méthodologie utilisée dans la cueillette des données, c'est-à-dire lors du codage. Le contexte et les particularités de l'expérience sont aussi présentés, et la méthodologie de codage est évaluée rapidement. Globalement, la méthodologie utilisée respecte les règles habituelles telles que prescrites par Ericsson et Simon (1993).

Les données utilisées dans ce projet proviennent de revues qui ont eu lieu en 1996 dans une compagnie de consultation montréalaise. Le projet s'est étendu sur une période de six mois et était formé de quatre informaticiens. Les réunions de revue ont été enregistrées sur vidéo, d'une manière discrète afin de minimiser l'impact de l'observation sur la réunion. Les enregistrements vidéos sont en effet essentiels dans l'analyse des interactions entre les divers membres d'une équipe de développement (Jordan et Henderson, 1995). Sept des quinze réunions ont été transcrites et codées dans le cadre du projet ACGL. La transcription de ces réunions a été réalisée par une personne qui n'était pas familière avec le projet de développement ni certains termes techniques en informatique. Cette personne est donc considérée objective, si bien que la transcription est aussi considérée objective.

La transcription est à la base du reste du codage. Pour la réaliser, chaque intervention de chaque participant est notée. Une intervention est identifiée comme une prise de parole partielle ou complète d'un participant: s'il est considéré qu'une intervention englobe des actions différentes, on la divisera en autant d'interventions. Dans le cadre de ce projet, la division originale des interventions a été conservée dans toutes les versions de codages. Modifier cette organisation aurait entraîné des complications importantes dans la stratégie de comparaison: on compare le choix du code de chaque intervention. D'ailleurs, dans

une étude de Seaman (1999b) semblable à celle-ci, la division des interventions était très similaire entre les divers codeurs employés.

Pour coder une réunion, plusieurs éléments sont requis. Tout d'abord, la transcription de la réunion doit être disponible. Puis, l'enregistrement vidéo de la réunion est un excellent complément de la transcription. Enfin, il est très utile d'avoir accès à chaque artefact introduit et considéré dans la réunion pour l'identification des objets de la discussion et pour identifier les interventions de lecture.

Le codage d'une réunion se fait généralement en associant un code à chaque intervention à partir de la transcription de la réunion. Si le contenu de l'intervention ne suffit pas pour identifier l'activité qu'on y retrouve, alors le codeur se réfère au contexte, c'est-à-dire à un certain nombre d'interventions précédentes. Cependant, le ton pris lors de l'intervention peut modifier la signification de l'intervention. Dans le doute, le codeur écoute l'enregistrement vidéo de la réunion pour fixer son choix de code (Gero et McNeill, 1997).

Il est à noter que les réunions sont divisées par séquences d'interventions, où une séquence est un ensemble d'interventions sur un même sujet global. Les sujets de discussion sont introduits par l'activité d'introduction (INTRO). Dans le but de pouvoir comparer les sujets des interventions et de conserver la division des séquences de la réunion, les codes INTRO ont été uniformisés. Ceci n'a qu'un faible impact sur l'analyse puisque les introductions ne sont pas une activité qui soit en elle-même intéressante à étudier, mais forment plutôt une activité utile pour l'homogénéité du modèle. C'est grâce aux introductions que le code d'une intervention peut référer au sujet de discussion.

1.6.1 Problèmes potentiels avec le processus de codage

La transcription est la principale source d'information sur la réunion utilisée dans le codage. Il est donc important que cette source soit de bonne qualité. Bien sûr, il est très rare d'avoir une transcription parfaite. Lethbridge (1999) remarque d'ailleurs que le processus de transcription est ardu et qu'il est difficile de trouver quelqu'un capable de faire une bonne transcription d'une réunion. La moindre erreur dans un mot de la transcription peut être très lourde de conséquences et modifier profondément la signification d'un ensemble d'interventions. Dans le cadre de discussions techniques telles celles d'une réunion de révision, il est avantageux que la personne réalisant la transcription ait une certaine connaissance du vocabulaire utilisé pour augmenter le degré de précision de la transcription.

Évidemment, la qualité de la transcription dépend grandement de la qualité du son dans l'enregistrement vidéo. Les paroles d'une personne ne parlant pas en direction du micro, ou ayant une intensité de voix insuffisante, bien que suffisante pour les participants de la réunion, seront plus difficile à comprendre. Dans ce projet, la caméra vidéo était positionnée dans un coin de la salle, pour minimiser son impact sur les discussions. Ceci a pour conséquence que les interventions de faible volume faites par des personnes tournant le dos à la caméra (et donc au micro) sont parfois difficiles à comprendre. Une solution aurait été de placer un micro centré avec la table, mais toujours d'une manière discrète et non encombrante.

Notons que diverses approches sont possibles pour transcrire le contenu d'une réunion. Une transcription peut contenir uniquement la séquence de paroles énoncées, en identifiant les intervenants. À l'autre extrême, une transcription peut contenir tous les codes nécessaires pour inférer le ton de la conversation (Schenkein, 1978). Dans le cadre de ce type de projet, aucun code d'intonation n'est généralement utilisé dans la transcription (Sanderson et Fisher, 1994). Une transcription est un média de

communication moins riche qu'un enregistrement vidéo, suivant la définition de richesse de média de Daft et Lengel (1986). C'est pourquoi il est important de se référer à l'enregistrement vidéo lors du codage.

Donc, pour retrouver le ton du discours, on doit constamment avoir recours à l'enregistrement vidéo. Mais une vérification continue ralentit le processus puisque la bande doit être revue plusieurs fois pour parfaire le codage et valider l'intention derrière une intervention. Il est parfois complexe de déterminer la signification d'un "Eh", d'un "Ben" ou même d'un "Oui", même en ayant accès à l'enregistrement. Sans l'enregistrement, cette tâche devient quasi impossible, car le contexte est parfois pauvre ou insuffisant.

Notons que le contexte culturel doit être pris en compte lors du codage. En effet, un "Non, OK." n'a pas nécessairement la même signification pour des personnes d'origines culturelles différentes. Ce problème est peut-être moins important dans des équipes regroupant des participants de diverses cultures, puisque le langage parlé serait probablement plus international, laissant de côté les régionalismes. Mais, les données qui sont accessibles ne permettent pas de mesurer ce phénomène puisque les participants avaient sensiblement le même bagage culturel.

1.7 Méthodologie de validation

La validation de la méthode d'analyse étudiée se continue par l'analyse de sa fiabilité. La problématique de la fiabilité pour le type d'expérience étudiée est tout d'abord exposée, puis la méthodologie utilisée est décrite et justifiée.

La définition du biais est nécessaire avant de continuer plus loin. Un biais de codage est une erreur systématique d'un codeur qui tend à associer un code *A* à des interventions où

un code *B* serait plus indiqué par les définitions. Un biais peut être une indication d'une mauvaise compréhension des définitions des codes, donc d'un apprentissage insuffisant, ou d'une ambiguïté entre les deux codes. Le biais peut aussi être vu à un niveau plus général, comme étant l'apport de la vision particulière d'un individu au codage.

1.7.1 Analyse de données qualitatives

Suivant la théorie de la mesure (Stevens, 1946), une partie des problèmes de mesure de la fiabilité, discutés plus loin, ne serait pas présente dans le cas d'un système de codage par intervalle, contrairement à un système purement nominal comme c'est le cas ici. Un système de codage est analogue à un système de mesure puisqu'on associe une valeur unique à une situation. Le schéma de codage associe des valeurs nominales aux interventions (Pfleeger et al., 1997). Ce type de données est caractéristique des expériences qualitatives.

Une expérience est considérée comme qualitative si les données recueillies sont dans le format de mots et d'images au lieu de nombres (Seaman, 1999a). Les analyses de données qualitatives sont donc beaucoup plus difficiles et complexes que celle de données quantitatives. Évidemment, les mathématiques appliquées sont bien plus développées pour l'étude quantitative. Seaman remarque aussi qu'il est souvent plus utile de combiner des analyses qualitatives et quantitatives, pour profiter de leur caractère complémentaire.

Le système étudié est de type nominal, par rapport aux autres types existants. Un système où on assigne uniquement un mot ou un nombre non significatif est de type nominal. Il aurait en effet été possible d'avoir un système de codage impliquant l'utilisation de valeurs scalaires, par exemple un système ordinal comme les adresses civiques ou un système par intervalle comme la mesure de la température en degrés Celsius, ce qui aurait permis d'utiliser des outils mathématiques plus puissants (Fenton, 1994). Divers outils,

comme les index KU20 et KU21 de Kuder et Richardson (1937) ou même le coefficient de corrélation, seraient disponibles si le codage n'était pas nominal (Dunn. 1989). Dunn propose des outils mathématiques pour les systèmes nominaux qui nécessitent la transformation de la mesure nominale en une mesure intervalle par l'affectation de distances entre les catégories. Cependant, à moins de justifications théoriques solides, une telle opération est arbitraire et ne respecte pas la théorie de la mesure (Pfleeger et al., 1997).

Mais Dunn reconnaît aussi qu'il ne semble pas exister d'outils de mesure applicables directement aux systèmes nominaux et que l'accord entre codeurs est habituellement la meilleure solution à l'évaluation de la fiabilité (voir aussi Rust et Cooil. 1994). Mais cette absence d'outils directement appropriés cause l'incertitude quand à la meilleure approche pour estimer le mieux possible la fiabilité d'un système nominal.

1.7.2 Fiabilité d'une mesure nominale

Il a été discuté que la fiabilité se mesurait par l'analyse de la stabilité et de la précision d'une mesure. Dans le cas de mesures physiques comme la distance ou le temps, ceci est facilement fait en répétant tout simplement des mesures sur un même phénomène, et en calculant la variance de ces mesures. Mais l'analyse n'est pas aussi simple pour un système de mesure de type nominal. Comment déterminer la variance statistique dans l'assignation de catégories à des interventions?

En général, il est recommandé d'effectuer une analyse en profondeur de la fiabilité d'une expérience. La méthode la plus complète est la théorie de la généralisabilité. Introduite par Cronbach et al. (1972), elle consiste à identifier les facteurs pouvant affecter la fiabilité d'une expérience, à répéter l'expérience d'une manière contrôlée autant de fois que nécessaire en faisant varier un facteur à la fois et à comparer les résultats obtenus à

l'aide des outils statistiques proposés par les auteurs. La philosophie derrière la méthode dit qu'il faut identifier dans quel univers les résultats peuvent se généraliser. La théorie de la généralisabilité est reconnue comme la meilleure méthode d'estimation de la fiabilité (par exemple, Berk, 1979; Kazdin, 1977; Mitchell, 1979).

Une telle démarche donne d'excellents et riches résultats, mais est excessivement lourde pour l'expérimentateur (Johnson et Bolstad, 1973). La littérature montre d'ailleurs fort peu d'expériences utilisant la démarche, les expérimentateurs préférant utiliser des outils qui ont un lien moins direct avec la fiabilité et qui sont moins exacts. La théorie de la généralisabilité est critiquée par certains qui la trouve inutilement complexe. Nelson et al. (1977) suggèrent qu'il n'est pas nécessaire d'être capable de préciser autant de dimensions de généralisation pour qu'une méthodologie soit utile et précise, et que l'absence de généralisabilité pour un univers ne signifie pas que la mesure est imprécise. Mitchell (1979) ajoute qu'il ne vaut pas la peine de faire une étude de généralisabilité uniquement pour estimer la fiabilité d'une mesure, mais plutôt pour l'intérêt scientifique qu'apportent les résultats d'une telle étude. Et Cooil et Rust (1994) mettent en doute la capacité du modèle d'évaluer des situations qualitatives.

Bien plus, Baer (1977) critique les techniques statistiques trop abstraites de mesure de la fiabilité, comme la théorie de la généralisabilité (Hartmann, 1982), en remarquant qu'elles sont simplement des variantes qui augmentent la quantité de résultats sans en améliorer la qualité. Dans le même sens, Hawkins et Fabry (1979) craignent que la sophistication statistique ne remplace le mérite scientifique. Pour Baer, la mesure de la fiabilité d'une étude qui implique l'observation et le codage ne doit pas se faire en mesurant la précision et la stabilité d'un observateur, mais en mesurant la stabilité entre les observateurs. C'est ainsi que la stabilité, et la fiabilité selon lui, sont significatives et importantes. Ceci implique une simple mesure d'accord entre observateurs.

D'autres auteurs affirment dans le même sens que l'accord entre codeurs est une mesure significative et généralement suffisante de la fiabilité. Johnson et Bolstad (1973) remarquent que plus l'accord est important entre les codeurs, moins le biais d'observateur est probable. Cone (1977) préfère mesurer la fiabilité par la théorie de la généralisabilité, mais reconnaît que la priorité doit être mise sur l'accord entre codeurs. Bakeman et Gottman (1986) affirment même que l'accord entre codeurs englobe la fiabilité, puisque celle-ci est la comparaison d'un codage avec la "vérité" (dans l'aspect précision de la fiabilité), et donc qu'un bon accord entre codeurs est probablement suffisant. Kazdin (1977) affirme d'ailleurs qu'il n'y a aucune base solide pour conclure qu'un codage est "vrai". Baer ajoute que la technique elle-même est la "vérité" pour l'expérience et donc qu'il est illogique de tenter de faire un codage parfait pour comparer avec les autres codages. Ces auteurs jugent l'objectivité comme mesure suffisante de la fiabilité par rapport à la théorie de la généralisabilité.

Le problème principal dans l'utilisation de l'accord entre codeurs comme mesure de la fiabilité est qu'il est possible que les codeurs aient tous le même biais, c'est-à-dire qu'ils se trompent tous de la même manière, et ainsi que l'accord soit grand sans que la mesure soit fiable (Hollenbeck, 1978). Brown et al. (1968) remarquent que "A team of observers can be brainwashed to the point of near-perfect agreement", mais que ceci ne garantit pas la fiabilité de leurs observations. Medley et Mitzel (1963) croient qu'il est plus important qu'une mesure soit stable plutôt qu'elle ait un bon accord entre codeurs. Ces auteurs penchent donc tous vers une mesure de la stabilité intra-observateurs comme mesure de fiabilité, par la comparaison d'observations de divers moments de l'expérience faites par un même observateur. Deux aspects sont ainsi mesurés: la dérive de l'observateur et la stabilité du contenu de la réunion. Ils cherchent donc à mesurer la variabilité (ou stabilité) des résultats dans la durée des observations.

Cependant, d'autres auteurs répondent que l'utilisation de l'accord inter-codeurs est meilleure qu'une mesure de la stabilité comme estimation de la fiabilité. Cone (1977)

croit qu'il est plus important de mesurer l'accord inter-codeurs que la mesure de stabilité par l'accord intra-codeurs, dans la mesure de la fiabilité. Birkimer et Brown (1979), entre autres, croient que l'accord entre codeurs est suffisant pour mesurer la fiabilité. Frick et Semmel (1978) remarquent qu'il n'est pas possible de faire des comparaisons de type nominal pour mesurer l'accord intra-observateurs. Et Bakeman et Gottman (1986) reconnaissent le principal défaut de l'accord entre codeurs, mais évaluent qu'en général, lorsque deux observateurs indépendants sont en accords, on considère que leur jugement est correct. Ces auteurs préfèrent donc l'objectivité (ou précision) des résultats, par rapport à la stabilité du contenu ou à la dérive de l'observateur.

1.7.3 Méthode générale d'étude

Tel que mentionné, ce projet vise l'étude de la fiabilité et de la validité d'une méthode d'analyse des aspects humains du génie logiciel. La fiabilité de la méthode doit donc être étudiée. L'absence d'une méthodologie statistique bien adaptée permettant de mesurer la fiabilité d'un système nominal et la faible rentabilité de la théorie de la généralisabilité nous forcent à contourner le problème en limitant l'étude sur la précision et l'objectivité de la mesure. Certains auteurs considèrent que ceci est amplement suffisant pour mesurer la fiabilité, tandis que d'autres préfèrent une mesure de stabilité comme mesure de fiabilité. La plupart des expériences similaires se satisfont de l'objectivité (Rust et Cooil, 1994).

L'étude de la fiabilité passe donc par l'analyse de l'objectivité de la méthode d'analyse étudiée. Tel que mentionné dans la problématique de recherche, l'accord inter-codeur est mesuré. En même temps, des biais de codage pourront être identifiés, s'il y en a de présents.

L'ajout d'un codeur, et donc d'une version de codage d'un même ensemble d'interventions, permet d'identifier des situations de biais, et représente la répétition expérimentale discutée plus haut. L'idée est qu'il est peu probable que les codeurs aient le même biais. Malheureusement, cette situation demeure possible. Il ne semble pas exister de moyen absolu pour éliminer totalement le biais, on peut seulement ajouter continuellement des codeurs pour réduire la taille du biais (Kratochwill et Wetzel, 1977). À partir de la comparaison des codages, on peut mesurer et analyser l'accord entre les versions.

1.7.3.1 Conditions expérimentales

L'analyse de l'accord entre les codeurs est faite au niveau atomique, c'est-à-dire au niveau des interventions. Pour chaque intervention, les codes associés par les codeurs sont comparés. Une autre avenue d'analyse serait possible: étudier les différences entre les variables d'analyse résultantes des diverses versions du code. C'est l'approche utilisée par Olson et al. (1992), dans leur étude de la collaboration dans les réunions de design. Mais l'analyse au niveau des interventions permet une plus grande précision et une plus grande richesse de résultats. Et surtout, elle est plus appropriée aux analyses de structure effectuées.

Trois versions du codage d'une réunion sont comparées. Une première version a été produite pendant la construction du schéma de codage, en 1997, par d'Astous en collaboration avec les autres auteurs du schéma de codage. La seconde a été réalisée au début du mois d'avril 1998 par une autre personne, dans le cadre d'un projet pour un cours aux études supérieures. Le modèle n'avait été qu'introduit sommairement à ce moment au codeur. La troisième version a été réalisée au mois d'octobre 1998 par le même codeur, après un apprentissage plus complet du modèle. L'auteur des deuxième et troisième versions est également l'auteur des calculs et analyses d'accord. Ceci aurait pu

introduire un biais supplémentaire, mais Rusch et al. (1975) montrent qu'une telle situation n'est pas problématique.

Seule la réunion 02 a été codée par les deux codeurs, sur les sept réunions transcrites et codées, et sur les quinze réunions du projet. Ceci sera important à considérer dans la section 4.2, lors des analyses séquentielles de l'impact des variations entre les versions.

L'utilisation de deux codeurs différents permettra d'analyser en profondeur les différences existantes dans les deux codages, mais ne donne qu'une dimension de l'accord. Il n'y a aucune analyse longitudinale possible avec seulement deux codeurs et une version chacun. L'impact du degré d'apprentissage d'un codeur ne peut être étudié qu'avec l'utilisation d'un codage supplémentaire réalisé par un même codeur. C'est pourquoi trois versions sont utilisées dans ce projet. Bien sûr, un plus grand nombre de codeurs aurait pu être utilisé, ce qui aurait ajouté de la profondeur à l'analyse en permettant de mesurer la différence du désaccord de divers couples de codeurs, et aurait minimisé les chances de biais particuliers à un codeur dans l'analyse.

1.7.3.2 Analyses effectuées

Cette partie de l'étude nécessite la comparaison de deux ou plusieurs versions de codages d'un même ensemble d'interventions. Aussi, l'ajout de codeurs permet de mesurer l'accord entre eux, et donc de mesurer le degré d'objectivité du codage. On définit l'accord comme la comparaison des codes, pour chaque intervention de chaque version. Un accord existe si les deux activités des codes de chaque version sont identiques. L'accord peut être généralisé en une indication globale de l'étendue de l'accord au niveau des interventions. Remarquons que seul l'accord des activités est mesuré, puisque c'est la partie du code la plus informative sur ce qui se passe dans l'intervention, et est donc la partie la plus importante. Mais il est important de se rappeler que la mesure est

incomplète en ce sens, elle ne tient pas compte de l'accord du sujet de discussion ou du critère utilisé.

Les diverses versions sont ensuite comparées entre elles pour mesurer la différence qui s'y retrouve. Des outils statistiques sont nécessaires pour obtenir une mesure fiable du biais. Puisqu'aucun outil statistique disponible n'est parfaitement adapté à la situation du projet, l'analyse concurrente des divers outils appropriés est nécessaire. Les résultats obtenus seront comparés pour tenter d'en extraire une idée assez précise du biais. Les résultats concernant les codes considérés plus "intéressants" seront étudiés plus en profondeur; les codes tels AUT et GES ne sont pas aussi importants que les autres, dans l'étude des activités collaboratrices d'une réunion.

En étudiant les facteurs du contexte du codage des diverses interventions, il sera possible de caractériser les interventions ayant un accord entre les versions par rapport à celles ayant été codées différemment par les codeurs. L'impact de l'apprentissage sera ainsi mesuré en comparant les versions 2 et 3, réalisés par le même codeur, avec la version 1, réalisée par d'Astous. La durée des interventions sera aussi étudiée ainsi que l'impact de certains mots-clés sur certains codes d'activités et la distribution des désaccords pour chaque code. On note que les versions 1 et 3 seront celles qui seront les plus comparées, puisqu'elles correspondent aux deux dernières versions de chaque codeur, celles où les codeurs avaient une formation importante.

Puis, l'impact des différences observées dans les versions 1 et 3 sera mesuré. Pour ce faire, les analyses effectuées à partir du codage de d'Astous seront répétées avec un codage équivalent provenant de l'autre codeur. Ces analyses sont principalement illustrées dans la thèse de d'Astous (1999) et seront reprises presque intégralement.

Mais ces analyses ne sont pas aussi précises lorsqu'utilisées avec les 345 interventions de la réunion 02, plutôt que les 1950 interventions des sept réunions réunies. Deux solutions

permettent de résoudre cette situation: coder à nouveau les six réunions n'ayant été codées qu'une fois, ou simuler le nouveau codage de ces six réunions en se basant sur les différences observées dans la réunion 02. Le but de cette présentation est ceci: la deuxième solution est explorée, mais elle implique deux nouvelles hypothèses. La réunion 02 est représentative de toutes les réunions codées, et les différences observées dans le codage de la réunion 02 sont représentatives des différences qui auraient été observées si les sept réunions avaient été codées par les deux codeurs. Les analyses utilisant les nouveaux codes simulés seront donc restreintes par la validité de ces hypothèses. Ceci sera discuté plus en profondeur à la section 4.2.

1.8 Résumé du chapitre

Les études empiriques, base du développement scientifique, sont devenues nécessaires en génie logiciel, alors qu'il y a un grand nombre d'articles présentant des idées sans montrer si elles sont en accord avec la situation réelle du développement logiciel. Et les résultats d'une étude empirique qui n'est pas répétée peuvent être simplement le fruit du hasard ou d'une erreur expérimentale. La répétition et la validation d'expériences sont essentielles.

De plus, les aspects humains du développement logiciel sont un domaine du génie logiciel qui requiert de plus en plus d'attention, maintenant que l'on commence à entrevoir les limites des modèles existants. Le développement de logiciels est un processus essentiellement humain et doit être étudié en tant que tel. Un exemple de champs d'étude qui requiert des études empiriques est la réunion de révision, étape coûteuse mais importante dans le développement logiciel. Une bonne compréhension du contenu de ces réunions est nécessaire pour optimiser leur déroulement et augmenter leur efficacité.

Une étude empirique menée par d'Astous et Robillard examine justement les interactions entre les participants d'une réunion de révision. Ce mémoire tente de valider leur approche en examinant la fiabilité de leur méthode d'analyse, en répétant l'expérience pour analyser son objectivité, les facteurs influents de l'accord entre les versions des codes et l'impact des différences entre les versions sur les résultats.

CHAPITRE II
MESURE DU DEGRÉ D'ACCORD

Le chapitre précédent a introduit les notions nécessaires expliquant le cadre de ce projet. Dans la méthodologie présentée par d'Astous (1999), le codage des réunions est l'étape qui a le plus grand potentiel d'introduire un biais dans le processus d'analyse. Ce chapitre proposera différents outils statistiques utilisés pour mesurer l'objectivité de la méthode d'analyse étudiée, puis présentera et discutera des résultats obtenus à l'aide de ces outils.

2.1 Outils d'analyse

Différentes méthodes statistiques ont été élaborées au fil des années, pour tenter d'évaluer l'écart entre deux versions d'un même codage, le but étant de montrer que le biais introduit par le choix des codes n'est pas trop important et ne fausse pas les conclusions trouvées. De nombreux chercheurs surveillent avec attention les diverses propositions dans le domaine, car un grand nombre de projets nécessite l'utilisation de tels outils dans la validation de leur approche. Le débat entre James et al. (1984, 1993), et Schmidt et Hunter (1989) est un exemple éloquent de discussions pouvant survenir dans le domaine.

2.1.1 Outils statistiques conventionnels

Une des mesures les plus directes et intuitives est la proportion d'accord entre deux versions, c'est-à-dire la proportion d'interventions ayant été codées identiquement par les deux codeurs. Cette proportion n'est généralement pas la mesure la plus importante, mais donne habituellement une idée générale de l'accord. Cependant, son utilisation n'est pas encouragée puisque la proportion est erronée, c'est-à-dire qu'elle ne tient pas compte des accords qu'on peut attribuer au hasard. Bakeman et Gottman (1986) remarquent qu'il y a

trop de facteurs pouvant affecter la proportion d'accord, à un point tel qu'on ne peut pas comparer deux proportions de situations différentes entre elles. Deux versions aléatoires auront habituellement une certaine proportion d'accord, simplement due au hasard. Celle-ci est parfois aussi importante que 84 %, dans un exemple de Bakeman et Gottman. Et Hollenbeck (1978) recense cinq faiblesses de l'outil qui le rendent impropre comme mesure de l'accord. Si possible, on veut utiliser un outil plus approprié.

Il serait aussi tentant d'utiliser le coefficient de corrélation pour étudier la correspondance entre deux versions. Celui-ci mesure précisément le degré de lien entre deux variables. Cependant, cet outil n'est pas approprié dans le cas de codes nominaux, comme c'est le cas dans le schéma de codage utilisé, parce que le coefficient de corrélation requiert deux variables d'échelle intervalle. Les outils statistiques conventionnels ne sont donc pas suffisants pour la comparaison de codages.

2.1.2 Coefficient d'accord kappa de Cohen

Une des mesures d'accord qui corrige la situation du simple pourcentage est le coefficient kappa de Jacob Cohen (1960). Ce coefficient élimine de la proportion d'accord simple la partie due au hasard. Le kappa de Cohen est probablement l'outil le plus répandu pour étudier l'accord entre deux codeurs. Il est dérivé des travaux de Scott (1955). Les valeurs obtenues varient entre 0 si deux codeurs ne s'entendent que par hasard, et 1 dans le cas d'un accord parfait.

Les hypothèses de départ sont l'indépendance des interventions, l'indépendance du codage des divers codeurs, ainsi que l'indépendance, l'exhaustivité et l'exclusion mutuelle des catégories de codage (Brennan et Prediger, 1981). Toutes ces hypothèses sont remplies dans le système de codage étudié: les interventions sont considérées indépendantes puisqu'un code d'une intervention ne détermine pas directement le code

d'une autre intervention, le codage des différents codeurs est indépendant, les codes sont totalement indépendants, ils couvrent toutes les interventions possibles à l'aide du code AUT, et les catégories sont mutuellement exclusives, suivant leur définition précise dans la thèse de d'Astous (1999).

Pour calculer le coefficient kappa, on divise la différence entre la proportion d'accord observée P_O et la proportion due à la chance P_C, par 1 - P_C. C'est-à-dire que P_O - P_C est la proportion d'accord brute P_O d'où on retire la proportion d'accord due à la chance P_C. Cette proportion d'accord nette est divisée par la proportion d'accord maximale quand P_C est due à la chance, c'est-à-dire par 1 - P_C. Donc, on fait simplement retirer la chance de la proportion d'accord. L'équation 2.1 est la formule ainsi obtenue.

$$\kappa = \frac{P_O - P_C}{1 - P_C} \tag{2.1}$$

Soit N éléments à coder à l'aide d'une des n catégories. Ici, il y a 345 interventions à coder dans la réunion 02 et 11 codes possibles pour chacune. On compte les interventions ayant chaque couple de codes (e. g. DEV dans une version et DEV dans l'autre version, DEV et EVA, EVA et DEV, etc.). On obtient donc un tableau n par n, comme le tableau 2.1, où sont inscrites les fréquences de chaque couple de codes. Les diverses versions sont identifiées à la section 1.7.3.1.

Tableau 2.1 Fréquences d'associations des codes entre la version 1 et la version 3

	Version 3												
Version 1	DEV	EVA	DEM	JUSTIF	ACC	REJ	HYP	INF	INT	GES	AUT		$P_{i\bullet}$
DEV	12	1	1	1	2		2	3				22	0.064
EVA	1	17	4	1	2	3	3	3		1	1	36	0.104
DEM		2	27									29	0.084
JUSTIF		5		17	2		1	7				32	0.093
ACC		1			25	1		1			5	33	0.096
REJ		1			1	5		1			1	9	0.026
HYP	5	1	1				27	2			1	37	0.107
INF	1			3			2	43		1		50	0.145
INT									27			27	0.078
GES										20	2	22	0.064
AUT	1	2	2	1	3		2	4		7	26	48	0.139
	20	30	35	23	35	9	37	64	27	29	36	345	
$P_{\bullet j}$	0.06	0.09	0.1	0.066	0.1	0.03	0.1	0.2	0.08	0.08	0.1		

Un exemple aidera à comprendre ce tableau. Il y a 4 interventions pour lesquelles la version 1 assigne le code EVA et la version 3 assigne le code DEM. Ceci est représenté par la case à la deuxième ligne et troisième colonne. De même, il y a 2 interventions codées DEM dans la version 1 et EVA dans la version 3. La diagonale représente donc le nombre d'interventions en accord pour chaque code: 12 interventions en accord pour DEV, 17 pour EVA, etc. De plus, les sous-totaux marginaux sont inscrits pour chaque code et chaque version. Par exemple, il y a 20 interventions codées DEV dans la version 3, alors qu'il y en a 22 dans la version 1. Ceci donne une proportion d'accord de 0.6 par rapport à la version 3 (12 sur 20) et 0.55 par rapport à la version 1 (12 sur 22).

Dans le calcul de P_C, on doit calculer les probabilités marginales $P_{i\bullet}$ et $P_{\bullet j}$, où $P_{i\bullet}$ est la probabilité d'être dans la i-ème ligne et $P_{\bullet j}$ est la probabilité d'être dans la j-ème colonne. Pour obtenir P_C, on fait la somme des produits des probabilités marginales pour chaque code, comme le montre l'équation 2.2; cette somme représente la chance attendue de la situation. Évidemment, le calcul de P_O est simplement la somme des éléments a_{ii} sur la diagonale divisée par N, comme dans l'équation 2.3.

$$P_C = \sum_i P_{i\bullet} P_{\bullet i} \tag{2.2}$$

$$P_O = \frac{\sum_i a_{ii}}{N} \tag{2.3}$$

Il est aussi possible de calculer l'écart type de cette mesure. L'équation 2.4 est tirée d'un article de Fleiss et al. (1969), auquel le lecteur intéressé peut se référer pour une explication de l'équation.

$$\sigma_\kappa = \sqrt{\frac{\left(\sum_i P_{i\bullet} P_{\bullet i}\left(1-\left(P_{\bullet i}+P_{i\bullet}\right)\right)^2\right)+\left(\sum_i \sum_{i\neq j} \sum_j P_{i\bullet} P_{\bullet j}\left(P_{\bullet i}+P_{j\bullet}\right)^2\right)-P_C{}^2}{N(1-P_C)^2}} \tag{2.4}$$

Pour plus de précautions, plusieurs auteurs préfèrent se référer à la borne inférieure de la mesure. À un niveau de confiance de c, la borne inférieure se décrit comme:

$$Borne_{Inf} = \kappa - z_c \sigma_\kappa \tag{2.5}$$

où z_c est la valeur critique associée à c, suivant une courbe normale centrée réduite (puisque kappa tend à suivre une courbe normale: voir Fleiss et al, 1969). Par exemple, pour un niveau de confiance de 95 %, $z_c = 1.96$.

2.1.2.1 Limite du kappa et solution

Une des conséquences de la définition de P_C est que le hasard en est ainsi défini. Cependant, cette définition ne tient que dans un cas en particulier: quand les probabilités marginales sont fixées. Ceci revient à dire que les probabilités de chaque catégorie sont déterminées à l'avance et que les codeurs en sont informés. Ils doivent alors s'assurer d'être conforme à ces valeurs. S'il y a une discordance entre les probabilités marginales,

alors le kappa devient plus conservateur puisque la valeur maximale devient inférieure à 1. Perreault et Leigh (1989) remarquent justement que, dans plusieurs cas, utiliser le kappa de Cohen est non seulement conservateur, mais peut induire en erreur. Dans ce projet, comme dans plusieurs études exploratoires, les probabilités marginales ne sont pas fixées. Le kappa n'est donc pas parfaitement approprié à la situation de ce projet.

Il existe une famille de coefficients d'accord entre codages qui sont dérivés du kappa de Cohen (Brennan et Prediger, 1981; Cohen, 1968; Dewey, 1983; Landis et Koch, 1977), auquel s'ajoute l'index phi (Haggard, 1958). Tous ont la même structure et visent à éliminer la contribution de la chance de la proportion d'accord brute. Ils diffèrent dans leurs hypothèses de départ ou dans leur définition du hasard. Par exemple, le kappa de Brennan et Prediger κ_n suppose une équiprobabilité des catégories. Les résultats obtenus à l'aide de ces différents kappas peuvent être grandement différents de ceux obtenus grâce au kappa de Cohen.

Différents coefficients d'accord basés sur kappa ont été calculés dans le cadre de ce projet, et les différences ont été notées seulement à partir de la troisième décimale. Les différences entre les variantes sont donc faibles pour ce projet. Ceci est logique puisqu'il n'y a pas de définition parfaite de la chance, aucun index n'est parfaitement approprié au projet, il y a toujours une déviation d'un côté ou de l'autre. Seul le kappa de Cohen sera donc discuté, représentant les variantes assez fidèlement pour les besoins de ce projet.

Pour contourner le problème des probabilités marginales fixées, il est courant de calculer le kappa maximal qui peut être atteint par un accord parfait, mais avec une distribution asymétrique (Brennan et Prediger, 1981; Cohen, 1960). La mesure corrigée est obtenue en divisant le kappa standard par le kappa maximal. Cette mesure se comporte de la même manière que le kappa, mais peut atteindre la valeur maximale de 1 pour toute distribution des codes. Le kappa maximal est identique au kappa standard, sauf pour la définition de

la proportion d'accord brute P_O. On remplace celle-ci par un accord parfait, décrit à l'équation 2.6:

$$P_O = \sum_i \min(P_{i\bullet}, P_{\bullet i}) \tag{2.6}$$

2.1.3 Index de fiabilité

Une approche tout à fait différente a été prise par Perreault et Leigh (1989) pour mesurer l'accord entre deux codeurs. Leur index de fiabilité ne fait pas les hypothèses nécessaires aux kappas, quand à la distribution des probabilités marginales. L'index présenté par les auteurs est basé sur celui de Bennett et al. (1954). Il est uniquement fonction de la proportion d'accord et du nombre de catégories n. Plus précisément, si P est la proportion d'accord et I_r est l'index de fiabilité, alors:

$$I_r = \sqrt{\left(P - \frac{1}{n}\right)\left(\frac{n}{n-1}\right)}, \text{ pour } P \geq \frac{1}{n} \tag{2.7}$$

On remarque que la formule n'est pas valable pour P inférieur à $1/n$. En effet, ceci représente une situation où la proportion d'accord est inférieure à celle attendue de codeurs choisissant des codes au hasard. Dans ce cas, I_r est fixé à 0.

Il est également possible de calculer l'écart type de l'index de fiabilité. L'équation est plus simple que celle de l'écart type du kappa de Cohen. Évidemment, on peut aussi calculer la borne inférieure d'une mesure, de la même manière que pour le kappa de Cohen.

$$\sigma_{I_r} = \sqrt{\frac{I_r(1-I_r)}{N}}, \text{ pour } I_r N > 5 \text{ et } N > 30 \tag{2.8}$$

Cette mesure de fiabilité corrige l'erreur des kappas dans le cas de distributions sérieusement inégales de codes. Les kappas ont tendance à sous-évaluer de tels cas, erreur que ne ferait pas l'index de fiabilité, selon les auteurs. La mesure n'est pas aussi répandue que le kappa, mais est utilisée dans plusieurs recherches sérieuses, par exemple dans un article du Gallup Organization (Ha, 1998).

2.1.4 Plus de deux codeurs

Tous les outils d'analyse présentés jusqu'ici font la comparaison entre deux versions de codage d'un même événement. Cependant, comment comparer trois versions différentes, comme c'est le cas ici si on tient compte de la version 2, ou un nombre indéterminé de versions? Il existe quelques outils qui permettent justement une analyse de l'accord entre plus de deux codeurs.

Pour déterminer le codage final à utiliser dans les analyses prévues par l'expérience, quand deux codeurs réalisent un codage, on s'assure de leur accord à l'aide des outils présentés et on choisit une des deux versions en considérant qu'elle est bonne, ou bien on divise les éléments à coder entre les deux codeurs, ou encore mieux, on tente d'arriver à un consensus entre les deux codeurs. Dans le cas de trois codeurs ou plus, pour déterminer le codage, une pratique courante veut que le codage soit simplement "voté" par les codeurs: le code choisi par le plus grand nombre de codeurs sera utilisé dans le codage final (Bakeman et Gottman, 1986). Les éléments où tous les codeurs sont en désaccord sont analysés plus en profondeur et un compromis est atteint.

Pour simplement mesurer l'accord entre trois codeurs ou plus, on peut utiliser certaines versions du kappa. En effet, Landis et Koch (1977) ont adapté le kappa de Cohen à une situation très générale. Cohen (1968) avait déjà permis d'associer des poids pour préciser l'importance relative de chaque erreur. Le kappa de Landis et Koch permet, en plus de

prioriser certains désaccords, de considérer plusieurs sous-ensemble de codeurs sur plusieurs sous-ensembles d'éléments à coder. D'autres kappas généralisés ont été documentés, entre autres celui de Uebersax (1982), qui généralise le kappa de Conger (1980). Conger avait en effet généralisé des versions différentes de kappas d'accord entre plus de deux codeurs. Il avait montré que ceux-ci sont généralement équivalents à la moyenne du kappa de Cohen pour toutes les paires de codeurs. Donc, à moins que la situation force l'utilisation de particularités de certains kappas, on peut simplement utiliser la moyenne des kappas pour déterminer le kappa commun de plusieurs codeurs.

Récemment, Rust et Cooil (1994) ont proposé une nouvelle mesure, qui tente de faire le lien entre l'index de fiabilité de Perreault et Leigh et le kappa de Brennan et Prediger (1981; voir aussi Cooil et Rust, 1994 et 1995). Leur mesure a la forme générale du kappa, mais utilise un calcul tout à fait différent de la proportion d'accord observée P_O. Comme toutes les autres mesures d'accord, le PRL a une valeur de 1 lors d'un accord parfait et de 0 lors de l'absence d'accord. C'est au niveau des hypothèses et de la base théorique que sont les différences. Le PRL est une fonction de l'accord global entre tous les codeurs, du nombre de codeurs et du nombre de catégories disponibles. Il est effectivement une généralisation de l'index de fiabilité.

Dans ce projet, il n'est pas approprié d'utiliser des outils statistiques analysant l'accord commun entre 3 codeurs puisque seuls deux codeurs ont été utilisés. Trois versions sont disponibles puisqu'un des codeurs a codé deux fois, mais ceci ne justifie pas l'utilisation de tels outils puisqu'une des hypothèses de départ (trois versions indépendantes) n'est pas respectée. Il est en effet facile de démontrer la dépendance entre les versions 2 et 3, même si ces versions sont différentes. De plus, l'analyse de l'accord de deux codeurs est plus intéressante, car elle donne de l'information plus détaillée concernant l'accord ou le désaccord.

2.1.5 Interprétation des mesures

Les mesures présentées jusqu'ici ont un avantage commun: leur plage de valeurs significatives est entre 0 et 1. Ceci est vrai pour la proportion d'accord autant que pour les kappas ou l'index de fiabilité. Cependant, leur interprétation peut être plus problématique.

On cherche généralement en statistiques à tester si une valeur est significative ou non. Pour les mesures décrites, ces tests ne sont pas intéressants puisqu'ils ne testent que si un accord est significativement supérieur à l'accord attendu du hasard. Cohen (1960) reconnaissait lui-même qu'un test de signification n'apporte rien d'intéressant puisqu'il est attendu que l'accord sera de beaucoup supérieur à la chance.

Il y a tout d'abord le problème de fixer des qualificatifs aux valeurs obtenues. Quel est le seuil d'un "bon" accord ou d'un accord trop faible? En général, on peut mettre tous les index sur le même pied. Cependant, les différences de bornes et le conservatisme changent l'interprétation. En effet, une proportion d'accord simple a une limite inférieure à 0. Ce cas représente un désaccord complet entre les deux codeurs. Par contre, pour le kappa de Cohen et pour l'index de fiabilité, 0 représente un certain accord dû au hasard, c'est-à-dire environ $1/n$ pour la mesure de la proportion.

De plus, certaines mesures interprètent différemment certains accords. Par exemple, pour une situation comme celle décrite dans le tableau 2.2, la proportion d'accord est de 82 %, le kappa de Cohen $\kappa = 0$, et donc $\kappa / \kappa_{max} = 0$ aussi ($\kappa_{max} = 1$), la variante de Brennan et Prediger $\kappa_n = 0.64$ et l'index de fiabilité $I_r = 0.8$ (exemple tiré de Perreault et Leigh, 1989). Donc, qu'est-ce qui est plus représentatif? Et, surtout, est-ce que l'accord est effectivement bon ou non?

Tableau 2.2 Fréquences d'association de codes, exemple fictif

	A	B
A	81	9
B	9	1

La réponse à ces questions vient des hypothèses de codage. Si les probabilités marginales étaient fixées, alors κ est approprié et l'accord est nul. Sinon, on pourrait utiliser soit κ_n ou I_r en fonction des hypothèses, et l'accord serait presque bon.

Ceci amène un deuxième problème: quelles valeurs sont associées à des "bons" accords, à des accords "excellents", etc. Dunn (1989) remarque avec raison que toute délimitation sera nécessairement subjective. Il n'existe pas de moyens pour déterminer précisément les bornes d'un "bon" accord.

Ceci étant dit, plusieurs auteurs ont présenté des valeurs d'interprétation différentes. Ils reconnaissent généralement tous que ce ne sont que des divisions arbitraires (Landis et Koch, 1977), des "rule of thumb" (Bakeman et Gottman, 1986). Le tableau 2.3 compare les valeurs selon certains auteurs.

Tableau 2.3 Comparaison des valeurs d'interprétation des mesures

Accord	Bakeman et Gottman	Fleiss	Hartmann	Landis et Koch	Perreault et Leigh	Rust et Cooil
excellent		0.75		0.8	0.9	0.9
important	0.7	0.6	0.6	0.6	0.7	0.7
moyen		0.4		0.4		
faible				0.2		

Pour le lecteur intéressé, ces valeurs sont tirées de Bakeman et Gottman (1986), Fleiss (1981), Hartmann (1977), Landis et Koch (1977), Perreault et Leigh (1989), et Rust et Cooil (1994). Yelton (1979) a également contribué au débat, remarquant qu'un accord

significativement supérieur à ce que le hasard aurait donné est suffisant, ce à quoi s'objecte plusieurs auteurs (dont Bakeman et Gottman, 1986, et Cohen, 1960, 1968).

On voit donc qu'il y a une grande variation entre les auteurs. Par exemple, Fleiss (1981) considère qu'une valeur aussi faible que 0.5 peut être acceptable, contrairement à Bakeman et Gottman, pour qui une valeur inférieure à 0.7 représente un accord trop faible. D'ailleurs, les auteurs spécifient habituellement que leur division des interprétations est uniquement arbitraire, donc nullement basée sur une étude empirique. De plus, le 0.7 de Perreault et Leigh, pour leur index de fiabilité, vaut-il autant que le 0.7 de Bakeman et Gottman pour le kappa de Cohen? Il faut donc retenir de cette discussion que les valeurs obtenues ne peuvent être interprétées facilement, peu importe la statistique utilisée.

Un fait ressort de ce tableau: à partir de 0.75, l'accord est suffisant pour tous les auteurs commentant kappa, Perreault et Leigh fixent des bornes pour leur index de fiabilité. Dans la limite, bien entendu, que certains préfèrent un accord plus élevé dans le cadre d'expériences importantes, c'est-à-dire pas uniquement exploratoires. La majorité des auteurs s'entendent pour dire que 0.7 est une valeur suffisante pour être "bonne". Dunn (1989) note qu'un accord suffisamment "bon" est recherché pour qu'un seul des codeurs comparé soit utilisé pour coder les situations à venir, donc pour être en mesure de changer arbitrairement les codeurs.

Il est à noter que l'interprétation de la valeur d'un kappa est similaire à l'interprétation d'un coefficient de corrélation. Les deux mesures ont une valeur de 0 lorsqu'il n'y a aucun lien entre les données comparées, et de 1 lorsque tout est parfaitement lié. Aussi, dans son utilisation, un coefficient de corrélation ne fixe pas de limite absolue d'accord, de la même manière qu'aucun outil statistique d'accord ne fixe de seuil absolu de qualité d'une valeur obtenue. On dit plutôt que certaines données ont une corrélation plus ou moins forte.

Même s'ils ne s'entendent pas sur une valeur précise, les auteurs sont d'accord pour dire qu'une telle valeur dépend du contexte de l'étude. Une étude exploratoire permet, en général, un accord moins fort comme acceptable; d'Astous (1999) spécifie d'ailleurs que son expérience était exploratoire. Entrent aussi en ligne de compte la formation des codeurs, l'environnement et la méthodologie de codage, ainsi que la complexité du système de codage.

2.2 Analyse des résultats

L'analyse des codages s'est faite à l'aide de plusieurs outils statistiques, ce qui entraîne une quantité importante de données à analyser. Il est important de rappeler, avant d'aller plus loin, qu'il y a trois versions à comparer. La version 1 est celle réalisée par l'auteur principal du système de codage, en même temps que l'élaboration du système lui-même, en 1997. La version 2 a été réalisée par une seconde personne, avec un minimum de formation, en avril 1998. Enfin, la version 3 a aussi été réalisée par cette même personne, mais après une utilisation intensive du système de codage, en octobre 1998.

2.2.1 Proportions brutes

Il a été montré que les proportions brutes ne sont pas de très bonnes valeurs pour décrire l'accord. Cependant, elles demeurent intéressantes à considérer, car elles donnent une certaine indication de l'ordre de grandeur de l'accord. Une proportion d'accord de 1 entraîne généralement que tous les autres index auront une valeur de 1, par exemple. Aussi, pour un même contexte expérimental, si un accord est supérieur à un autre d'après la proportion brute, alors il sera aussi supérieur à l'autre d'après un autre index d'accord; les mesures d'accord conservent l'ordre entre les accords, dans une même expérience.

Trois proportions sont principalement considérées: l'accord entre la version 1 et la version 2, entre la version 2 et la version 3, et entre la version 1 et la version 3. De plus, la proportion d'accord total entre les trois versions a été calculée, ainsi que la proportion d'accord partiel, où deux des trois versions sont en accord, peu importe lesquelles. Les résultats sont résumés dans le tableau 2.4. Un rapide coup d'oeil indique que l'accord n'est pas parfait, mais semble quand même intéressant. Bien sûr, une étude plus poussée est nécessaire.

Tableau 2.4 Proportion d'accord par sous-ensembles de versions

Domaine étudié	Proportion d'accord
Version 1 et version 2	0.58
Version 1 et version 3	0.71
Version 2 et version 3	0.66
Accord total	0.51
Accord partiel	0.93

Un fait important peut être relevé à ce point-ci. On remarque que dans 51 % des cas, les trois versions sont en parfait accord, mais que les proportions d'accord, quand on compare les versions deux par deux, ne sont pas toujours très fortes. Ceci entraîne la conclusion que les divergences entre versions sont sur les mêmes 49 % d'interventions. Ce qui revient à dire que, même s'il y a 58 %, 71 % et 66 % d'accord entre deux versions, les interventions codées différemment sont généralement les mêmes. De plus, dans 93 % des cas, on peut décider du codage par un processus majoritaire, donc en prenant le code le plus populaire, choisi dans deux des trois versions.

Ces constatations sont riches en conséquences. On peut en effet déduire de ces résultats que le système de codage est très fiable et facilement utilisable pour 51 % des interventions. Pour l'autre moitié des cas, il y a des ambiguïtés à un niveau quelconque qui rendent le choix du code plus difficile. La difficulté n'est cependant pas énorme puisque dans 93 % des cas, deux versions sur trois sont en accord.

Bref, les proportions ne donnent peut-être pas d'index d'accord directement utilisable et fiable, mais elles fournissent beaucoup d'informations importantes et riches en interprétations.

2.2.2 Distribution des codes

Il est important, avant de continuer, de présenter la distribution comparée des codes choisis dans chaque version du codage. On remarque, dans la figure 2.1, que la distribution est plutôt similaire entre les versions, particulièrement entre les versions 1 et 3. Aussi, les codes sont sensiblement également répartis, mais il y a en même temps des différences importantes. Par exemple, il y a presque trois fois plus d'occurrences du code INFO que du code REJ. Ceci est une variation importante qui peut fausser les données. En effet, les kappas sont inférieurs quand la distribution des codes n'est pas uniforme et que les codeurs n'utilisent pas une distribution fixée d'avance.

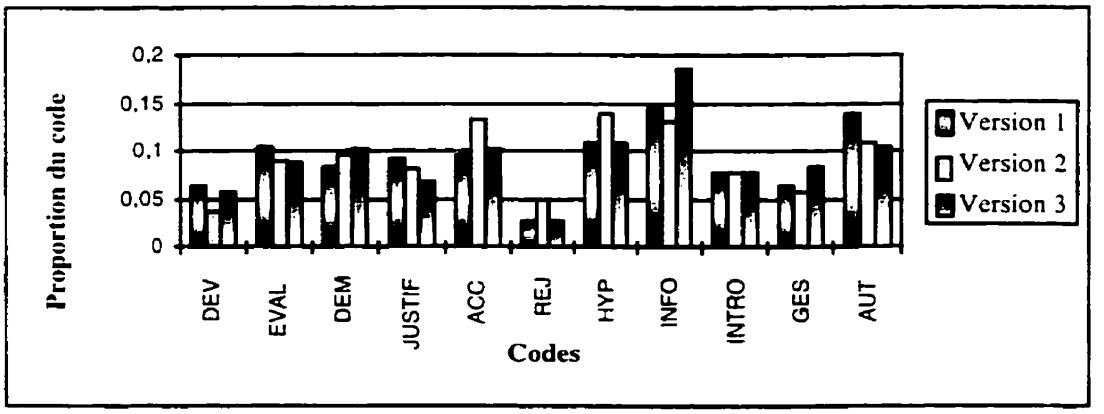

Figure 2.1 Distribution des codes dans les trois versions

Notons surtout que, pour un même code, les variations sont plutôt faibles. Toutes ont une certaine variation (sauf les INTRO) qui semble plutôt proportionnelle à leur fréquence. En somme, il n'y a pas d'écarts qui semblent trop significatifs.

2.2.3 Kappa

Le kappa de Cohen a été calculé pour chaque paire de versions. Les résultats sont résumés dans le tableau 2.5.

Tableau 2.5 Résultat du calcul de kappa par couple de versions

Domaine étudié	Kappa (κ)	Écart type	κ / κ_{max}
Version 1 et version 2	0.53	0.02	0.60
Version 1 et version 3	0.68	0.02	0.75
Version 2 et version 3	0.62	0.02	0.71

Les valeurs obtenues par le calcul de kappa standard sont intéressantes à plus d'un point de vue. Tout d'abord, elles semblent indiquer que l'accord entre les versions n'est pas extrêmement fort, mais sans être faible non plus. Évidemment, on préférera attendre d'avoir tous les résultats avant de poser un jugement définitif à ce sujet.

Aussi, la correction avec le kappa maximal donne des résultats tout d'abord assez différents, plus élevés dans tous les cas. Et les résultats deviennent plus intéressants, ils semblent indiquer un assez bon accord entre les versions 1 et 3.

De plus, on note que l'ordre d'accord est conservé entre la proportion brute d'accord et le kappa, ainsi qu'avec la correction du kappa.

Enfin, l'accord entre les versions 1 et 3 est d'intérêt, pour ce couple de versions le plus important, puisqu'il correspond à deux codeurs différents et expérimentés. Une valeur de 0.68 pour le kappa est intéressante parce qu'elle est suffisante selon certains auteurs, mais pas selon d'autres (voir plus haut). Mais le kappa corrigé de 0.75 égale ou dépasse les limites d'un "bon" accord pour tous les auteurs. Et il dépasse la limite populaire chez la majorité des auteurs de 0.7.

L'écart type donne des informations supplémentaires intéressantes. La borne inférieure, à un degré de confiance de 95 %, est 0.75 - 1.96 × 0.02 = 0.71, pour le kappa corrigé des versions 1 et 3. Ceci est toujours supérieur à la majorité des limites de "bons" accords.

2.2.4 Index de fiabilité

L'index de Perreault et Leigh a été présenté comme étant plus approprié que les kappas, dans la mesure où il ne diminue pas de valeur simplement parce que la distribution des codes est inégale. Les codes ne sont pas équiprobables d'une manière similaire pour chaque version. Les valeurs de l'index pour chaque couple de versions sont dans le tableau 2.6.

Tableau 2.6 Index de fiabilité de Perreault et Leigh par couple de versions

Domaine étudié	Index de fiabilité	Écart type
Version 1 et version 2	0.73	0.024
Version 1 et version 3	0.83	0.020
Version 2 et version 3	0.79	0.022

Ces valeurs sont très importantes, car elles confirment les données obtenues par le kappa. L'accord entre tous les couples de versions dépasse la limite des auteurs pour un "bon" accord. La borne inférieure de l'accord entre les versions 1 et 3, à un niveau de confiance de 95 %, est 0.83 - 1.96 × 0.020 = 0.79. Celle-ci est encore supérieure à la limite et est d'ailleurs amplement suffisante, selon Perreault et Leigh, pour une étude exploratoire.

De plus, il est à noter que les trois mesures comparant les couples de versions conservent toutes l'ordre entre les mesures d'accord. L'accord entre les versions 1 et 3 est le plus important pour la proportion brute, le kappa de Cohen, le kappa corrigé et l'index de fiabilité.

2.2.5 Comparaison des résultats

En observant tous les résultats, certains résultats semblent plus positifs que d'autres, c'est-à-dire que l'accord entre les versions semble plus grand selon l'index de Perreault et Leigh que selon le kappa de Cohen. Comme il a été discuté dans la section sur l'interprétation des résultats, les valeurs de chaque mesure ont une signification différente. Et certaines mesures sont plus appropriées que d'autres, en fonction de leurs hypothèses initiales. Une mesure d'accord doit être utilisée en fonction de sa signification dans le contexte expérimental (Goodman et Kruskal, 1954).

Les résultats de l'accord entre la version 1 et la version 3 sont "bons" d'après toutes les mesures. L'accord entre les outils de mesure est intéressant, et pourrait être considéré comme significatif de la force de l'accord, par rapport à une mesure qualifiant l'accord de "bon" et une autre le qualifiant de "moyen" ou de "faible". Dans un tel cas, plus de prudence aurait été nécessaire dans l'interprétation des résultats. Les résultats actuels signifient donc, dans les limites de la prudence raisonnable habituelle, que l'accord entre les versions 1 et 3 est "bon".

En comparant les deux versions du même codeur, les versions 2 et 3, il est possible d'observer une évolution dans leur accord avec la version de l'auteur du système de codage, la version 1. Ceci laisse entendre qu'il y aurait eu un apprentissage, entre le moment du codage des deux versions, qui aurait rapproché le codage des deux codeurs. Les divergences entre la version 1 et la version 2 peuvent être attribuées, du moins en bonne partie, au faible apprentissage du codeur de la version 2.

Les versions 2 et 3 sont en accord d'une manière plutôt importante. On peut expliquer ceci en rappelant que le même codeur est impliqué dans ces deux versions, mais à des instants différents. Il y aurait donc certaines différences entre ce codeur et le codeur de la version 1 qui seraient constantes dans les versions 2 et 3. Cette part d'interprétation, aussi

faible soit elle, sera toujours présente dans les études du comportement humain (Dunn, 1989). On ne peut pas viser l'élimination complète des différences entre codeurs, on ne peut que la minimiser et cerner son impact. Dans le cas de ce projet, cette différence semble être effectivement assez faible. Il est aussi possible que les désaccords entre les versions 2 et 3 soient dus à la variabilité de la mesure. Mais il est plus probable que l'apprentissage joue un plus grand rôle que la variabilité dans l'écart entre les versions 2 et 3. Par contre, ceci reste à confirmer.

On remarque aussi l'importance de l'accord global entre les trois versions. Cet accord n'est pas uniquement dû au fait que deux versions viennent du même codeur, car la proportion d'accord globale, de 93 %, est supérieure à la proportion d'accord entre les versions 2 et 3, de 66 %.

En somme, les résultats obtenus indiquent donc que les différentes versions n'ont pas un biais très important, et qu'on peut obtenir un codage fiable à partir de l'un ou l'autre des codeurs, du moins selon Perreault et Leigh (1989). On doit cependant s'assurer, comme pour toutes les études incluant le codage de comportements, qu'une formation suffisante est donnée aux codeurs, afin de prévenir les différences de codages.

2.3 Résumé du chapitre

L'objectivité de la méthode d'analyse proposée dans la thèse de d'Astous (1999) a été déterminée à l'aide des outils statistiques disponibles les plus appropriés. Aucune mesure n'est parfaitement adaptée à la situation étudiée, mais deux mesures semblent les plus proches, par rapport à leurs hypothèses de départ: le kappa (corrigé) de Cohen (1960) et l'index de fiabilité de Perreault et Leigh (1989). Ces deux outils montrent un "bon" accord entre la version 1 et la version 3 du codage d'une réunion.

L'interprétation d'une mesure d'accord est reconnue pour être difficile et imprécise. La majorité des auteurs proposant des bornes délimitant les "bons" accords des "moins bons" accords sont d'accord pour dire qu'un kappa supérieur à 0.7 est "bon". Le kappa corrigé de l'accord entre les versions 1 et 3 donne un résultat de 0.75, avec une borne inférieure de 0.71. Ces résultats sont au-dessus de la limite populaire de 0.7, mais en sont assez proche.

L'index de fiabilité de Perreault et Leigh donne des résultats similaires au kappa de Cohen, quoique plus positifs. La borne inférieure de l'accord des versions 1 et 3 est 0.79, bien au-dessus de la limite proposée par les auteurs de 0.7 pour ce genre d'étude. L'index de fiabilité de tous les couples de versions est supérieur à la limite.

L'accord plus grand entre les versions 1 et 3 par rapport aux versions 1 et 2 semble être une indication de l'importance de la formation du codeur des versions 2 et 3. Et les désaccords entre les versions 2 et 3 s'expliquent surtout par l'apprentissage, mais peuvent aussi être dus à la variabilité du codage.

En somme, sans être excellente, l'objectivité de la méthode semble être assez bonne.

CHAPITRE III
ÉTUDE DES FACTEURS AFFECTANT L'ACCORD

Il a été vu au dernier chapitre que l'objectivité de la méthode d'analyse est assez forte et suffisante. Mais ce projet vise une analyse plus profonde. L'accord entre codeurs doit être étudié plus en profondeur pour tenter de comprendre son fonctionnement et de cerner ses paramètres. Parmi les innombrables facteurs pouvant jouer sur l'objectivité, quatre seront étudiés dans ce chapitre: la durée de l'intervention codée, son contenu en mots-clés positifs ou négatifs, la dispersion des désaccords et l'effet de la formation d'un codeur.

3.1 Étude de la durée

L'analyse des interventions révèle un aspect important qui caractérise le codage. La durée des interventions a un impact certain sur le choix des codes. Intuitivement, il est normal qu'une intervention n'ayant qu'un seul mot ne contienne pas une grande quantité d'information. L'hypothèse posée dans cette section est que le choix du code est facilité lorsqu'une intervention contient beaucoup d'information, c'est-à-dire lorsque sa durée est plus grande. Cependant, certains codes auront des durées moyennes différentes, qui devront être prises en compte dans l'analyse.

3.1.1 Méthode de calcul

La durée d'une intervention n'est pas connue au moment du codage, il est seulement possible d'en faire l'approximation. En effet, selon la définition de d'Astous (1999), une intervention n'est pas uniquement un ensemble de phrases consécutives d'un même intervenant. Il n'est donc pas possible d'utiliser un outil de calcul automatique des durées. Par ailleurs, à partir de la transcription, il est impossible de déduire la durée, car les

pauses et la rapidité d'élocution ne sont pas indiquées. Enfin, il est préférable d'avoir une mesure de la durée qui fasse abstraction de la vitesse d'élocution de l'intervenant.

La mesure de temps qui sera utilisée n'est donc pas les secondes, mais le nombre de mots. Cette mesure a été utilisée par d'Astous (1999), et a été jugée plus fiable par celui-ci que le temps en secondes. Une certaine correspondance entre les deux mesures a été montrée dans son analyse.

Pour déterminer si la durée a un impact sur l'accord entre codeurs, on calcule la moyenne de la durée des interventions pour chaque code, dans le cas où les deux codeurs sont en accord et dans le cas où seul un des deux codeurs a choisi la catégorie concernée.

3.1.2 Analyse globale de la durée

Il y a vingt-six interventions où aucun accord, pas même partiel, n'est survenu dans les trois versions. Ces interventions représentent 7.5 % des 345 interventions de la réunion étudiée, mais ne représentent que 4.4 % de la durée de la réunion. Ceci revient à dire que les interventions "problématiques", où les trois versions sont en désaccord, ne forment pas une partie importante de la réunion. 96 % du propos de la réunion est codé identiquement dans au moins deux versions.

Par ailleurs, on note que les interventions ayant un accord complet entre les versions composent 56 % de la durée de la réunion, alors qu'ils ne sont que 51 % des interventions. La durée moyenne des interventions est de 14.7 mots pour les interventions ayant un accord total, de 13.1 mots pour les interventions ayant un accord partiel, et de 7.9 mots pour les interventions n'ayant aucun accord.

Ceci implique que les interventions plus longues sont généralement bien codées, alors que les interventions plus courtes sont plus problématiques. En particulier, en comparant uniquement les versions 1 et 3, la durée moyenne des interventions ayant un accord est de 14.6 mots, alors qu'elle est de 10.9 mots pour les interventions ayant un désaccord. L'hypothèse suivante peut donc être posée: la facilité du choix du code d'une intervention est proportionnelle à la durée de l'intervention à coder.

Certaines interventions ont des durées particulières, comme par exemple les introductions "implicites", de durée nulle. Ces interventions, ainsi que toute intervention de durée marginale (*outlier*), ont été retirées. Les durées moyennes sont alors de 14.8 mots pour les interventions en accord et de 8.8 mots pour les interventions en désaccord. La distribution des interventions, en fonction de leur durée, est décrite à la figure 3.1.

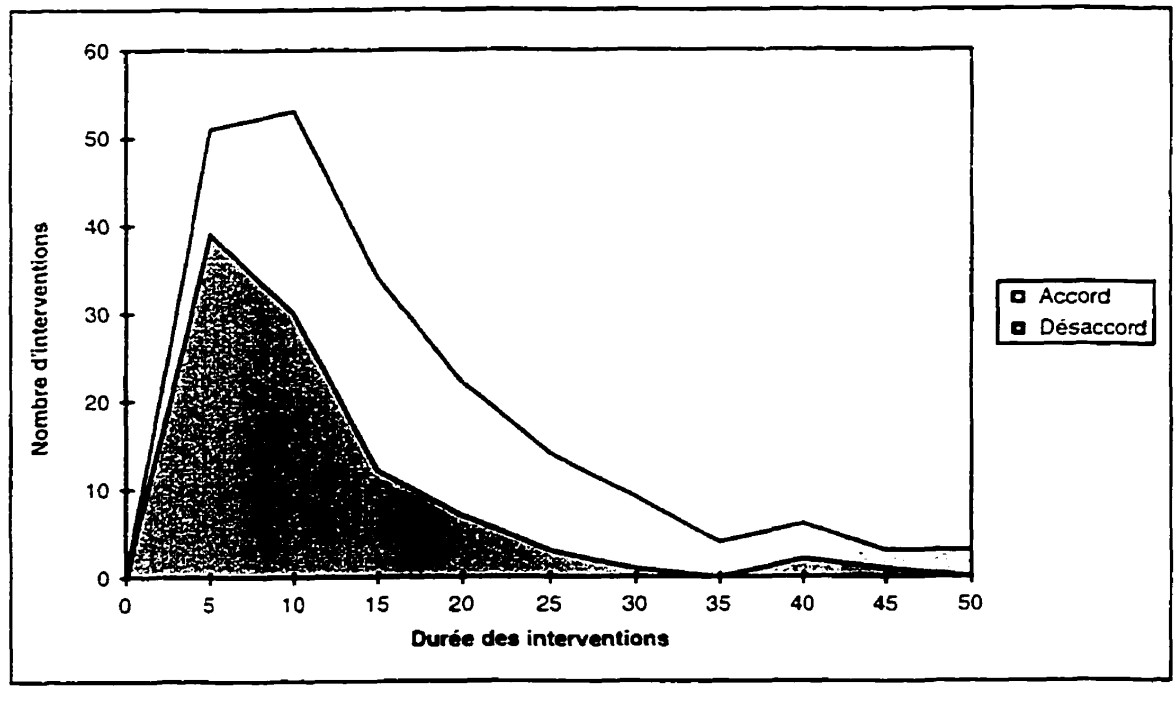

Figure 3.1 Interventions en fonction de la durée, pour les accords et désaccords

Les courbes semblent indiquer que les durées des accords sont légèrement supérieures à celles des désaccords. Mais, les deux distributions sont très semblables, elles sont seulement un peu décalées. Il est donc nécessaire d'aller plus loin dans l'analyse.

3.1.3 Résultats comparatifs

L'hypothèse doit maintenant être vérifiée pour chaque catégorie. Les résultats sont résumés dans la figure 3.2. Celle-ci montre, pour chaque code, la durée moyenne des interventions où un accord est survenu, comparée à la durée moyenne des interventions codée avec ce code dans une seule des deux versions. La figure révèle que l'hypothèse est vérifiée pour la majorité des catégories. Seuls les codes ACC et AUT ne respectent pas la règle. Les codes EVA et ACC ne font pas une bien grande distinction entre les cas d'accord et les cas de désaccord. Pour les autres codes, la différence apparaît clairement.

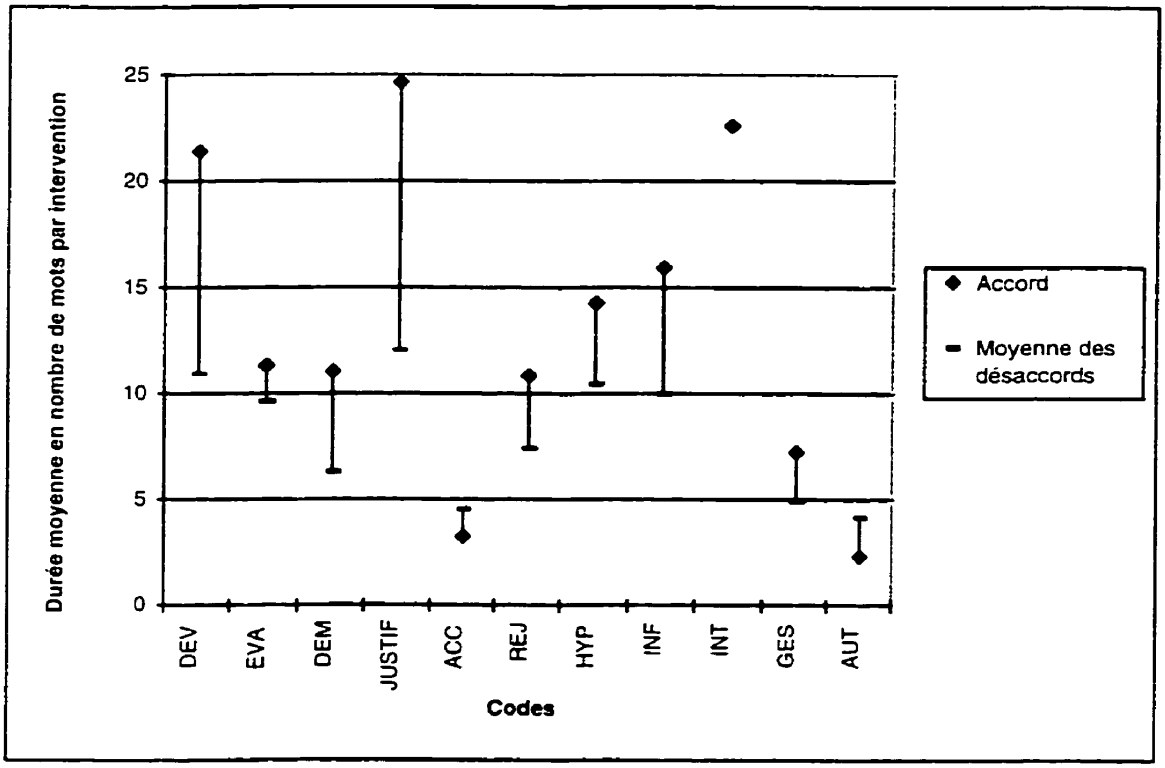

Figure 3.2 Durée comparative des accords et des désaccords, par code

Le graphe de la figure 3.2 montre clairement que, pour les codes DEV, DEM, JUSTIF et INF, la durée est un facteur majeur dans l'accord du codage. En effet, la durée des interventions codées différemment est environ la moitié de celle où les deux versions sont en accord. De plus, il est important de rappeler que les interventions codées INT sont toutes en accord, suite à leur uniformisation.

Le non respect de la règle par le code ACC peut s'expliquer par le fait qu'il peut être plus facile de coder une intervention ne contenant que "Oui" ou "D'accord" par rapport à une intervention plus longue. Dans cette optique, on peut s'attendre à ce que ACC fonctionne à l'inverse des autres codes. En revanche, ceci ne devrait pas nécessairement être de même pour REJ puisqu'un rejet s'accompagne en général de précisions, d'explications ou même simplement de mots de politesse, pour atténuer la dureté possible du commentaire.

Ceci n'est qu'une tentative d'explication de la différence observée, aucune preuve ne peut évidemment être obtenue.

La figure 3.3 présente en détail les durées des interventions en désaccord pour chaque version. On peut y voir la durée moyenne des interventions en accord pour chaque code *A*, ainsi que la durée des interventions codées *A* par la version 1 et codées autrement dans la version 3, et l'inverse, soit la durée des interventions codées *A* par la version 3 et codées autrement dans la version 1.

Ceci implique qu'une intervention où les deux versions sont en désaccord est représentée à deux endroits dans le graphique. Ceci était aussi vrai dans la figure 3.2. Si l'intervention est codée *A* dans la version 1 et *B* dans la version 3, alors la durée de l'intervention est intégrée à la moyenne des durées pour le code *A* et la version 1, et pour le code *B* et la version 3. Mais ceci n'est pas problématique puisque ce qui intéresse dans la figure 3.3 est uniquement la comparaison des durées d'interventions, codées *A* dans les deux versions, codées *A* uniquement dans la version 1 et autre chose dans la version 3, et codées *A* dans la version 3 et autre chose dans la version 1.

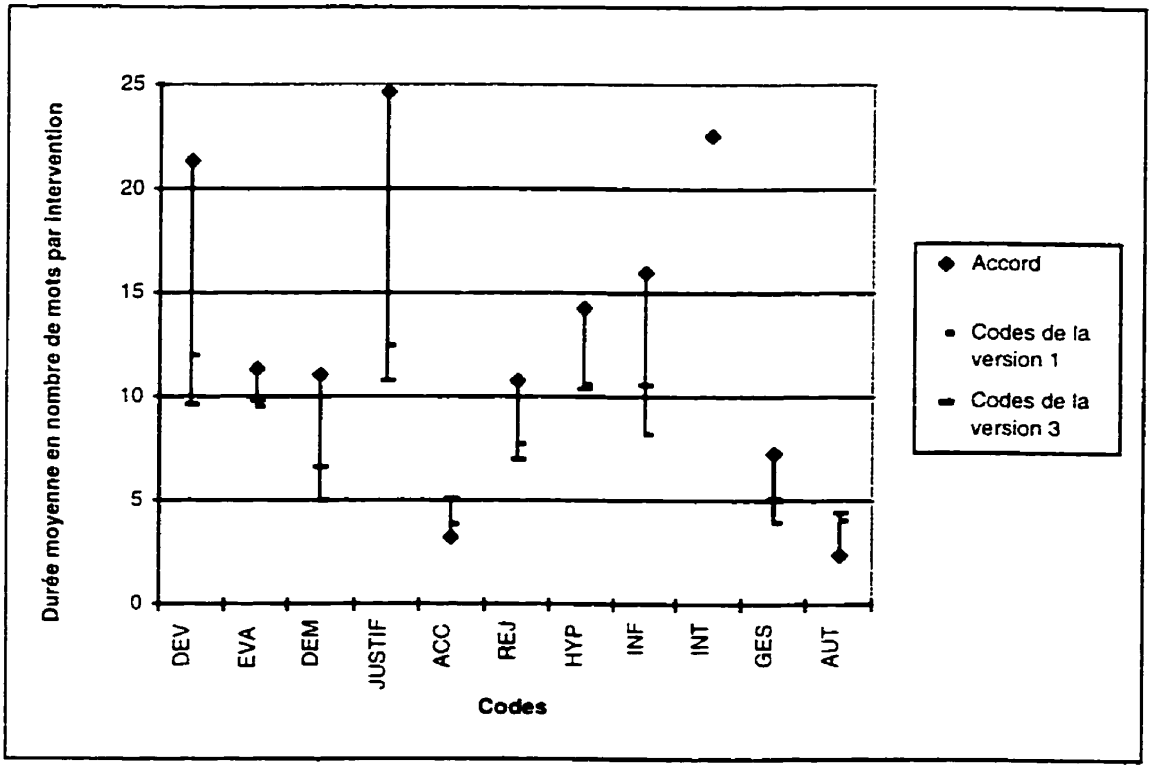

Figure 3.3 Durée comparative des accords et des désaccords par version, par code

La figure 3.3 montre clairement la proximité des durées des interventions en désaccord pour chaque version. Par exemple, pour le code DEV, la durée moyenne des accords est de 21.4 mots, celle des désaccords pour la version 1 est de 12 mots et celle des désaccords pour la version 3 est de 9.6 mots. Proportionnellement, les durées de 12 mots et de 9.6 mots sont très rapprochés, par rapport à la durée des accords, de 21.4 mots.

Il est important de noter que la quantité de données est faible dans certains cas. Le cas le plus grave est le code REJ, que chaque codeur n'a codé qu'à 9 occasions, dont 5 en accord. On doit donc tenir compte de cette restriction dans les analyses faites à partir de ces résultats. Les autres codes ont une taille d'échantillon plus raisonnable.

Évidemment, l'hypothèse étudiée ne peut être confirmée ou infirmée de manière certaine et précise à partir des données disponibles. Aucun test statistique ne peut être utilisé, seules les tendances peuvent être observées. Dans le cas présent, il est facile d'observer que la tendance est que la moyenne des durées des interventions en désaccord est inférieure à la moyenne des durées des interventions en accord.

3.2 Étude de l'impact de certains mots-clés

Le code ACC semble se comporter différemment des autres codes par rapport au lien entre la durée d'une intervention et l'accord entre les codeurs. Ceci est intéressant en soi, mais amène une nouvelle question de recherche: comment un codeur peut-il déterminer qu'une intervention est une acceptation si la quantité d'information est faible, de quelques mots seulement. Est-ce que des mots-clés pourraient avoir une influence sur le choix du code? Le raisonnement est le suivant. Si les interventions "positives" et courtes sont plus faciles à coder ACC. alors ceci suppose qu'une intervention peut être reconnue comme "positive" sans qu'un grand nombre de mots soit nécessaire. Il est donc possible qu'il existe des mots-clés permettant de qualifier une intervention de positive en peu de mots, seulement 3 en moyenne. Plus précisément, l'hypothèse posée est que les interventions comportant un ou des mots-clés positifs sont plus portées à être codées ACC. Le même raisonnement sera vérifié avec des mots-clés négatifs.

Le lien entre certains mots positifs et les interventions codées ACC est donc étudié. Trois mots-clés ont été étudiés dans l'ensemble de mots positifs: "Oui", "Ouais" et "OK". À l'inverse, des mots-clés négatifs ont aussi été étudiés, soient "Non", "Mais" et "Pas", pour vérifier si l'hypothèse pouvait être inversée aux interventions négatives. Néanmoins, l'étude n'est qu'exploratoire puisqu'elle repose sur le choix des mots-clés et que ceux-ci ont été choisis sans aucune justification théorique validée.

La méthode d'étude consiste à compter le nombre de mots positifs des interventions, en fonction du nombre de codes ACC. La figure 3.4 résume la situation pour les mots positifs. Les colonnes décrivent le nombre de mots-clés positifs par intervention, et chaque colonne est divisée en fonction du nombre de codes ACC associés à l'intervention. Deux codes ACC signifient que les codeurs étaient en accord pour cette intervention, un code signifie qu'un seul des deux codeurs a codé l'intervention ACC, et aucun code ACC signifie que les deux codeurs ont utilisé un autre code.

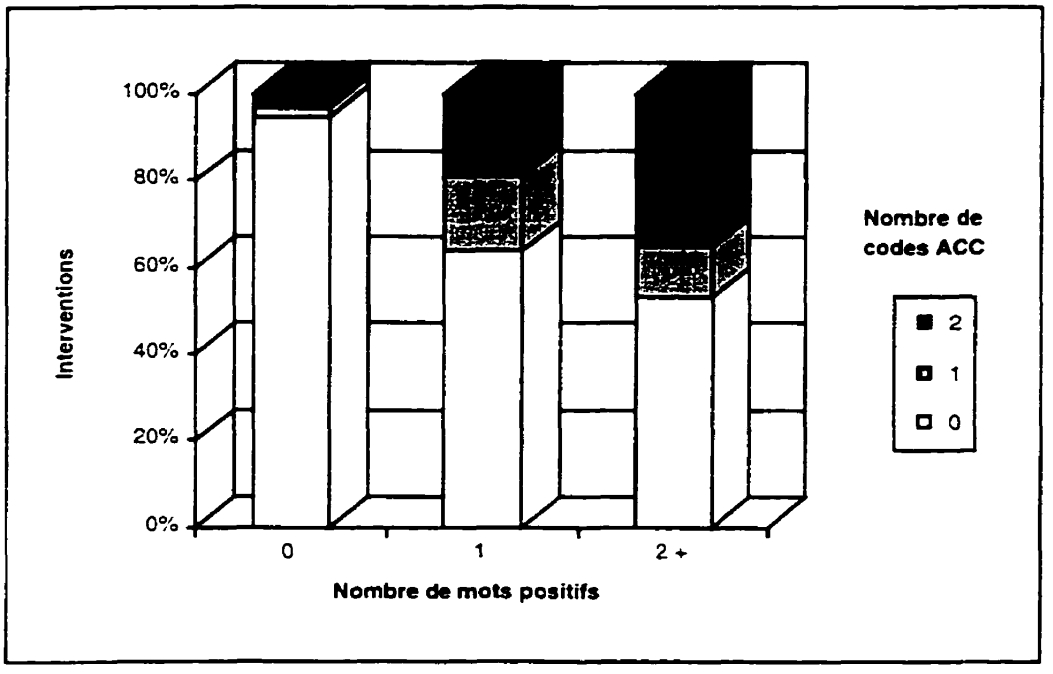

Figure 3.4 Distribution des interventions par nombre de mots-clés positifs en fonction du nombre de codes ACC

Le graphique de la figure 3.4 montre la tendance que plus le nombre de mots positifs est grand, plus une intervention a de chances d'être codée ACC. Par exemple, une intervention n'ayant aucun mot positif a seulement 5 % de chances d'être codée ACC, alors que 36 % des interventions ayant un mot positif ont été codées ACC par au moins un codeur, et que 47 % de celles ayant au moins 2 mots positifs ont été codées ACC par au moins un codeur, dont 35 % par les deux.

Cependant, les résultats ne sont pas réellement nets. Le lien n'est pas établi de façon non ambiguë, puisque plus de la moitié des interventions contenant au moins un mot-clé positif ne sont pas codées ACC par aucun codeur. L'hypothèse, telle qu'elle est formulée, n'est donc pas vérifiée, mais un lien est clairement présent entre les mots positifs et le code ACC.

La situation inverse concernant les mots-clés négatifs présente le même comportement, mais d'une manière beaucoup moins prononcée, comme le montre la figure 3.5. Seulement 6 interventions sur 39 ayant au moins 2 mots-clés négatifs sont codées REJ par au moins un codeur. Mais le faible nombre d'interventions codées REJ dans la réunion empêche de tirer quelque conclusion que ce soit.

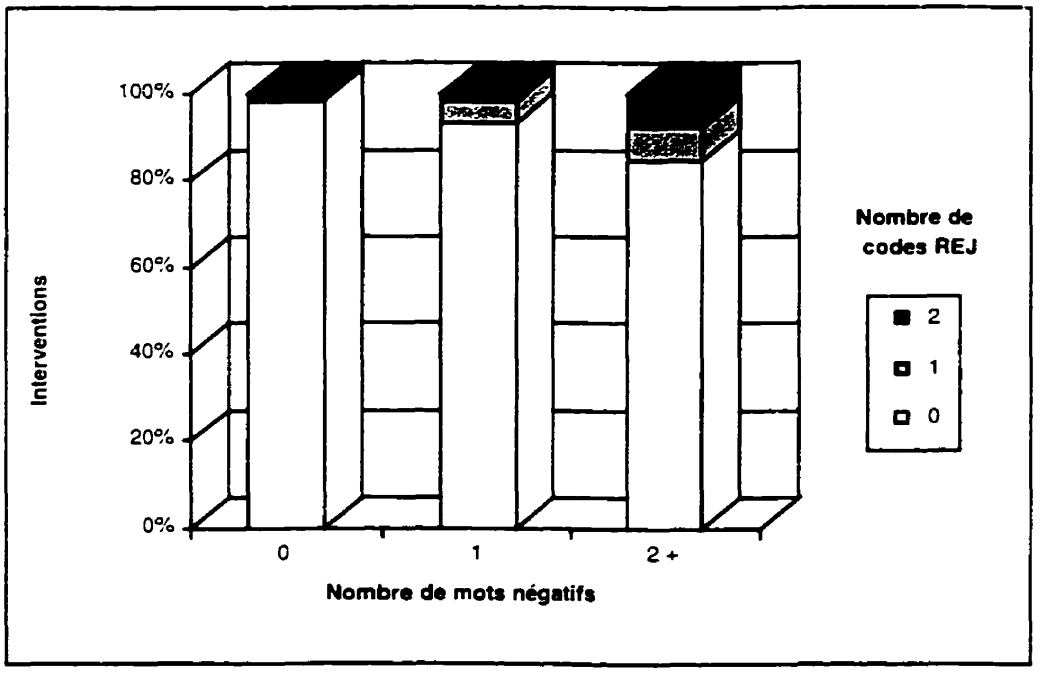

Figure 3.5 Distribution des interventions par nombre de mots-clés négatifs en fonction du nombre de codes REJ

Il est possible d'étudier la contraposée de l'hypothèse, c'est-à-dire d'étudier le nombre de mots-clés positifs pour les interventions codées ACC par au moins un des codeurs. L'hypothèse ici est que les interventions codées ACC ont la caractéristique d'avoir un nombre supérieur de mots-clés positifs que celles codées autrement. La figure 3.6 décrit la situation.

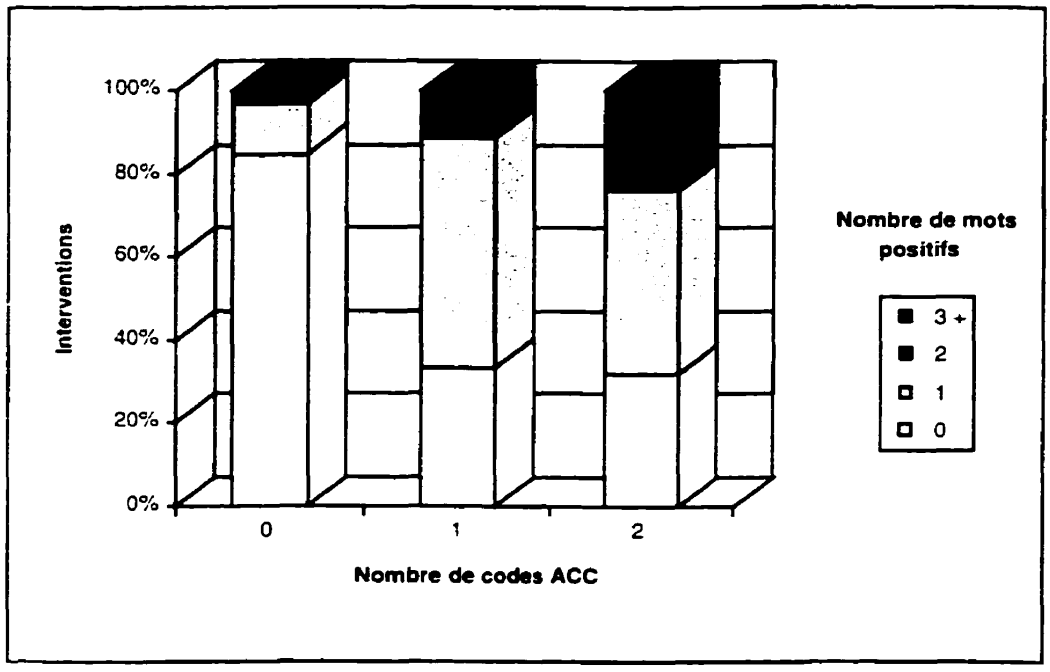

Figure 3.6 Distribution des interventions par nombre de codes ACC en fonction du nombre de mots-clés positifs

Le lien est beaucoup plus clair avec ce nouveau point de vue. La figure 3.6 montre qu'une intervention codée ACC par un ou deux codeurs a généralement au moins un mot positif. Par exemple, 85 % des interventions qui ne sont pas codées ACC n'ont aucun des mots-clés positifs, alors que 67 % des interventions codées ACC par un ou deux codeurs contiennent au moins un mot-clé positif.

La même situation est observée avec les mots négatifs et le code REJ. Ceci est présenté à la figure 3.7.

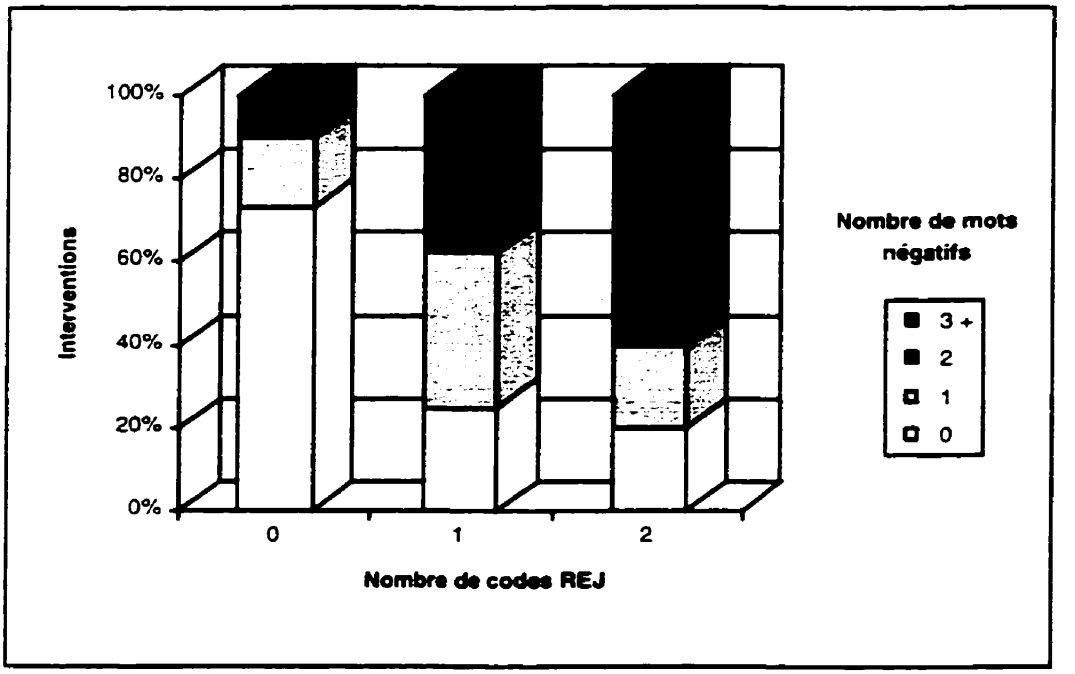

Figure 3.7 Distribution des interventions par nombre de codes REJ en fonction du nombre de mots-clés négatifs

Dans ce cas-ci, 73 % des interventions qui ne sont pas codées REJ n'ont aucun des 3 mots-clés négatifs étudiés, alors que 77 % des interventions codées REJ par au moins un codeur ont au moins un mot négatif. Toutefois, il faut être prudent en interprétant ces résultats, vu le faible nombre de rejets dans les données.

3.3 Étude de la proximité sémantique

Une observation importante ressort clairement de l'analyse du tableau de fréquences 2.1: pour certains codes, le taux d'accord est parfois faible. Il est possible que ce soit dû à un biais d'un codeur qui utilise systématiquement un autre code, et il est possible que les codes en désaccords soient répartis sur un large éventail de codes. En général, s'il y a confusion entre deux codes (*A* et *B*, par exemple), on pourra observer une fréquence importante de codes *B* dans une version qui sont codés *A* dans l'autre version. Ce projet

visait entre autres à trouver de telles confusions. Cependant, le tableau 2.1 n'indique aucune confusion importante de manière claire.

D'un autre côté, l'absence de confusion claire entre deux codes, jumelée à une proportion d'accord par code parfois faible (particulièrement pour les codes EVA, REJ et DEV), ne laisse qu'une alternative. Si les désaccords ne sont pas regroupés dans un certain code, c'est qu'ils sont dispersés sur plusieurs codes. Ce phénomène est observable à partir du tableau 2.1, où il y a des nombres non nuls dans un grand nombre de cases, pour un code. Par exemple, les interventions codées EVA, dans la version 3, sont codées à l'aide de 8 codes différents dans la version 1.

Le degré de dispersion est illustré à la figure 3.8. Les valeurs représentent la proportion des autres codes choisis, c'est-à-dire la proportion de codes pour laquelle un certain code est en désaccord dans au moins une intervention. Par exemple, dans le tableau 2.1, le code EVA, à partir de la version 1, est en désaccord avec 9 codes, et à partir de la version 3, il est en désaccord avec 7 codes. Ceci fait un total de 16 codes en désaccord, sur 20 possibles (10 codes autres que EVA, pour chaque version), c'est-à-dire 80 %.

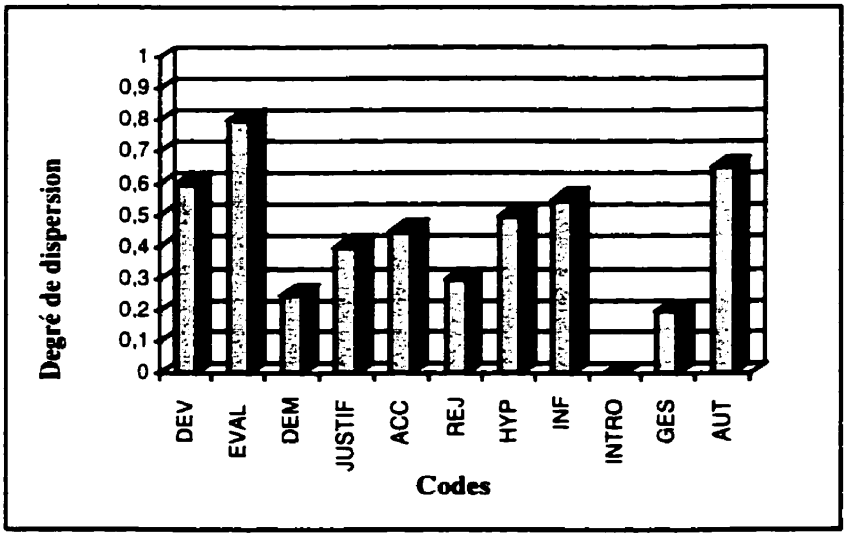

Figure 3.8 Degré de dispersion des désaccords, par code

Sur le plan théorique, le degré de dispersion des désaccords pourrait être un indicateur d'une certaine proximité sémantique entre un code et les autres codes. Deux codes sémantiquement "près" auraient des définitions plus semblables, plus proches, que des codes sémantiquement "loin". Mais ceci ne peut être vrai qu'en acceptant l'hypothèse que toutes les confusions qui pouvaient arriver sont effectivement arrivées. Mais l'absence de confusion entre un rejet et un développement indique-t-il vraiment l'impossibilité de confusion entre ces deux codes? On ne peut pas le supposer. De plus, un code comme REJ, codé seulement neuf fois dans chaque version, ne peut pas atteindre la dispersion maximale.

Cependant, il est clair que l'information sur la dispersion ne doit pas être ignorée pour autant. Sauf exception, il est possible de faire l'hypothèse que le degré de dispersion calculé pour chaque code est proportionnel à la proximité sémantique du code avec le reste des codes, et donc qu'un code ayant un haut degré de dispersion serait plus enclin à être codé différemment dans deux versions. Il n'est d'ailleurs pas surprenant que DEM et GES aient un degré faible de dispersion puisque ces codes sont très différents des autres, ils sont plus "loin" sémantiquement des autres codes que DEV l'est. DEM et GES ne sont pas dans le groupe de catégories de présentation (PRES). AUT non plus, mais par sa définition, il est tout à fait normal qu'il soit relié à un grand nombre de codes. D'ailleurs, AUT contient les interventions jugées incompréhensibles (INC) par un des codeurs, interventions que l'autre codeur peut avoir assez bien compris. Bien sûr, cette discussion est aussi imprécise que l'hypothèse étudiée et plusieurs autres raisons peuvent expliquer la situation observée.

Bref, d'après la figure 3.8, le degré de dispersion est élevé pour le code EVA, beaucoup plus que pour les autres codes. Ceci peut indiquer que EVA est sémantiquement proche du reste des codes. Un code comme EVA serait ainsi plus enclin à être codé différemment dans deux versions différentes. Sa définition n'en est pas "floue" pour autant, il y aurait

simplement plus de codes dont la définition est un peu plus proche par rapport aux autres codes.

3.4 Étude de l'apprentissage

La formation des codeurs est un aspect important. Comme le remarque Scott (1955), il est raisonnable de s'attendre à ce que des codeurs expérimentés aient un accord plus grand que des codeurs novices. L'apprentissage du système de codage et du processus de codage a un impact important sur l'accord mesuré (Reid, 1982). Même que le type de formation offert aux codeurs a un impact sur le niveau d'accord (Wildman et al., 1975).

Les versions 2 et 3 ont été codées par la même personne, une première fois avec un minimum de connaissances sur la méthodologie et la deuxième fois avec une expérience approfondie. Il est donc possible de mesurer l'impact de l'apprentissage. Une première façon est de comparer l'accord global entre les versions 2 et 1 et les versions 3 et 1, ce qui a été discuté au chapitre précédent. Il en ressortait de manière claire que l'accord des versions 3 et 1 est supérieur à celui des versions 2 et 1, ce qui permet de supposer que l'apprentissage du codeur a amélioré l'accord.

Une information plus détaillée est aussi disponible: l'apport de l'apprentissage sur chacun des codes peut être mesuré. Pour ce faire, les versions 2 et 3 sont successivement comparées à la version 1. Pour chaque instance de chaque code de la version 1, le code de l'autre version (2 ou 3) est noté. Une proportion d'accord par code est ainsi atteinte pour chaque version comparée. Pour chaque code, la différence entre les proportions d'accord indique le niveau d'impact de la formation. La figure 3.9 résume les résultats trouvés pour chaque code.

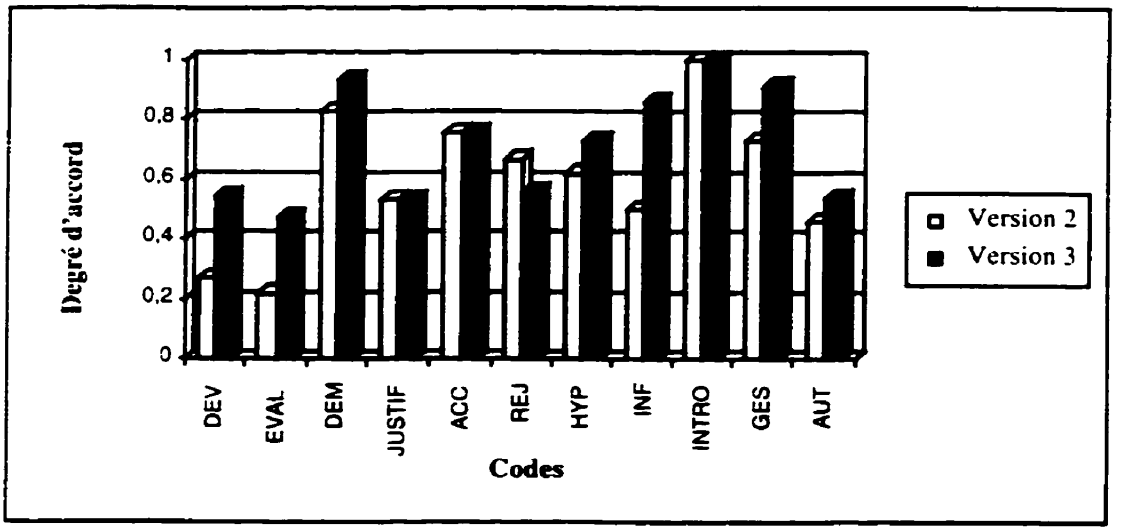

Figure 3.9 Proportion comparée d'accord avec la version 1, pour chaque code

Il en ressort des différences intéressantes. Pour tous les codes sauf REJ, la proportion d'accord entre les versions 1 et 3 est égale ou supérieure à celle entre les versions 1 et 2. Ceci montre donc l'impact positif attendu de l'apprentissage. Pour certains codes, comme DEV, EVA et INF, la différence est très grande. Il en est ainsi déduit que ces codes ont été la cause de problèmes de codage qui ont été réduits entre les versions 2 et 3. Et ces codes requièrent peut-être une formation approfondie, plus que les autres codes. Excepté que, puisqu'un seul codeur a été utilisé dans cette étude de l'apprentissage, il est possible que seules ses erreurs de compréhension personnelles soient rapportées à la figure 3.9.

Le code REJ a un accord légèrement plus faible. Mais, ici encore, le faible nombre de rejets entraîne une hypersensibilité du code aux variations. Il ne semble pas trop important de considérer la différence négative entre les versions 2 et 3 pour les rejets.

3.5 Résumé du chapitre

Divers facteurs pouvant affecter l'accord entre codeurs ont été étudiés. Certains semblent très clairs, comme l'influence de la durée sur l'accord. Pour presque tous les codes, la

durée des interventions où les deux codeurs sont en accord est nettement supérieure, parfois même le double, à la durée des interventions où il y a un désaccord.

D'autre part, certains mots-clés positifs semblent avoir une influence sur le code ACC, mais pas d'une manière très forte. Par contre, il est beaucoup plus clair que la grande majorité des interventions codées ACC ont au moins un des mots-clés positifs.

De même, un code comme EVA est noté pour la dispersion de ses désaccords, en ce sens que l'autre code choisi par un codeur, différent du EVA choisi par l'autre codeur, peut être presque n'importe quel code. Enfin, l'apprentissage semble être un facteur très important dans l'accord entre codeurs, et donc dans la qualité du codage. Presque tous les codes ont un plus grand accord dans la version 3 que dans la version 2, c'est-à-dire après un apprentissage important. Et les codes DEV, EVA et INF montrent une augmentation importante de l'accord entre les versions, signe potentiel qu'une attention particulière doit être portée à ces codes pendant la formation des codeurs.

En dépit de ces résultats intéressants, toutes ces analyses sont limitées par la quantité de données et le design expérimental. Des analyses plus approfondies, comme une étude de la corrélation entre la durée et les mots-clés, seraient intéressantes et pourraient apporter des réponses importantes. Mais plus les analyses sont pointues, et plus la quantité de données requise est grande. Bien plus, les résultats obtenus peuvent n'être que le fruit du hasard ou n'être qu'une particularité individuelle. Ils devraient être étudiés plus en détails à l'aide d'une expérience contrôlée, pour bien cerner leur importance.

CHAPITRE IV
ANALYSE DE L'IMPACT DES VARIATIONS DE CODAGE

L'objectivité a été montrée satisfaisante dans le chapitre 2 et certains facteurs pouvant influencer l'accord entre codeurs ont été étudiés dans le chapitre 3. Ces causes de bons ou moins bons accords sont importantes à bien comprendre, mais l'étude doit être complétée par l'analyse des conséquences du désaccord. En particulier, ce chapitre tentera de répéter les analyses faites par d'Astous, Robillard et autres à l'aide de la méthode d'analyse étudiée. Les différences seront étudiées, et leur étendue et leur importance seront cernées autant que possible.

La thèse de d'Astous (1999) présente trois sections de résultats. Une première section analyse globalement les résultats obtenus d'une manière statique; en résumé, les proportions de chaque catégorie sont comparées. La deuxième section fait une modélisation log-linéaire pour étudier l'influence du rôle des participants sur les interventions. Et la dernière section cherche à identifier les échanges, par l'analyse des patrons des interventions.

Ce chapitre répétera les analyses de d'Astous autant que possible. Cependant, la quantité de données différente empêche la reproduction complète des analyses. C'est pourquoi la deuxième section ne peut être répétée, puisque le niveau de détail étudié requiert une grande quantité de données. Ceci est aussi vrai pour la troisième section, mais les données manquantes sont unidimensionnelles et pourront donc être simulées. La première section sera répétée en grande partie, c'est-à-dire tant que le niveau de détail n'est pas trop restreignant.

4.1 Analyse statique de la nature des interventions

L'analyse statique est la première étape dans l'analyse, parce qu'elle permet une vue globale des données et une caractérisation générale de la situation codée. Cette section tente de reprendre les graphiques de la section 5.1 de la thèse, plus particulièrement de la figure 5.3 à la figure 5.7. Les autres figures sont soit hors sujet ou trop pointues dans leurs observations.

Au départ, les quatre groupes de catégories sont analysés, ainsi que la catégorie AUT. Deux comparaisons sont faites: on compare les versions 1 et 3 entre elles au niveau des fréquences de chaque groupe de catégories, puis au niveau de la durée relative de ces groupes. La figure 4.1 présente les résultats comparés des fréquences et la figure 4.2 présente les résultats comparés des durées. Ces figures correspondent à la figure 5.3 de la thèse de d'Astous.

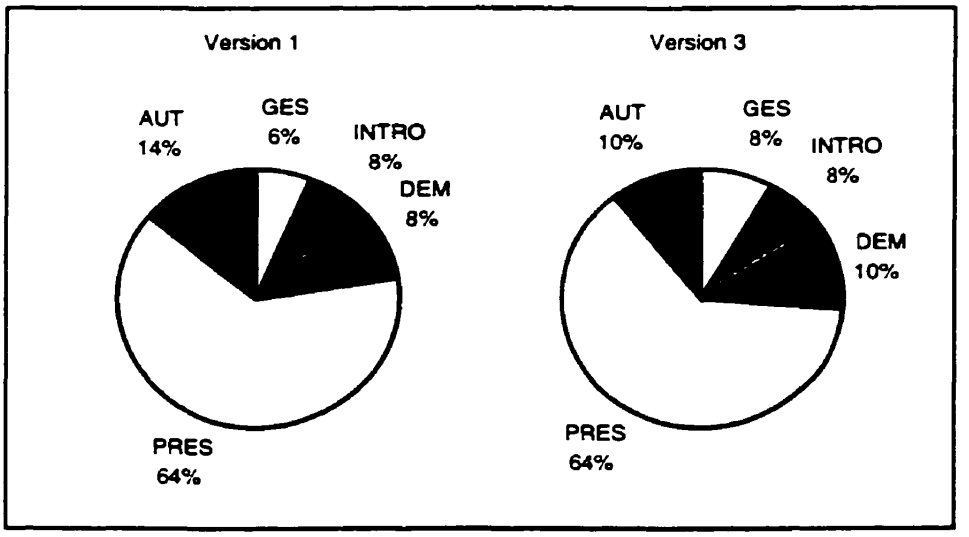

Figure 4.1 Fréquences relatives comparées des groupes de catégories

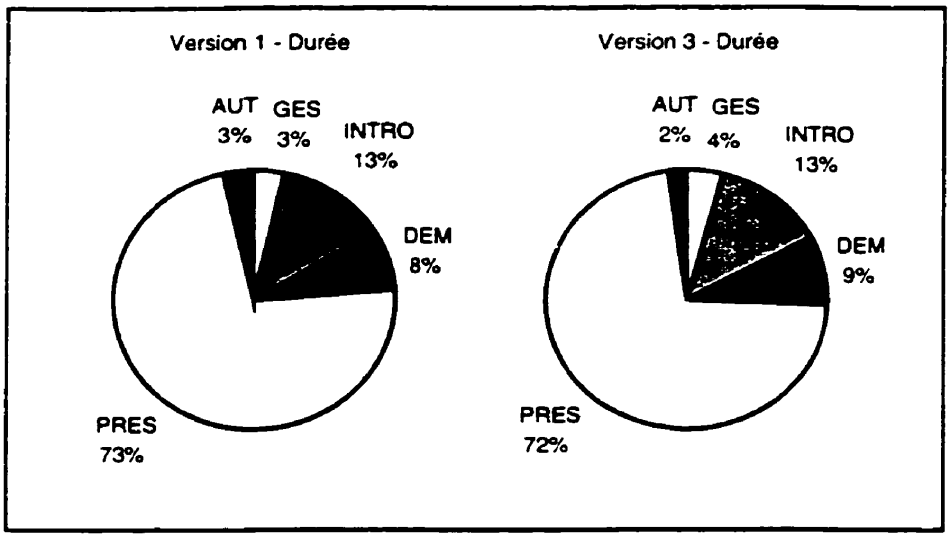

Figure 4.2 Durées relatives comparées des groupes de catégories

On observe à partir de ces deux figures que les deux versions sont très similaires à ce degré de détail. Les différences ne sont que de 1 % pour les durées relatives et de 4 % et moins pour les fréquences. La seule catégorie ayant une différence moyenne est AUT, qui varie de 10 % à 14 %. Ceci n'est pas alarmant, étant donné la nature de cette catégorie et le faible impact de cette différence.

Les prochains résultats présentés sont les mêmes que précédemment, mais en conservant uniquement les groupes de catégories formant les séquences de révision. Ceci revient à dire que les deux catégories n'apportant aucune information intéressante dans l'analyse de la réunion sont ignorées, soit GES et AUT. Ce type de manipulation est nécessaire pour avoir une meilleure vue des données et avoir des résultats plus clairs. Les figures 4.3 et 4.4 comparent ces nouveaux résultats; ils correspondent à la figure 5.4 de la thèse de d'Astous.

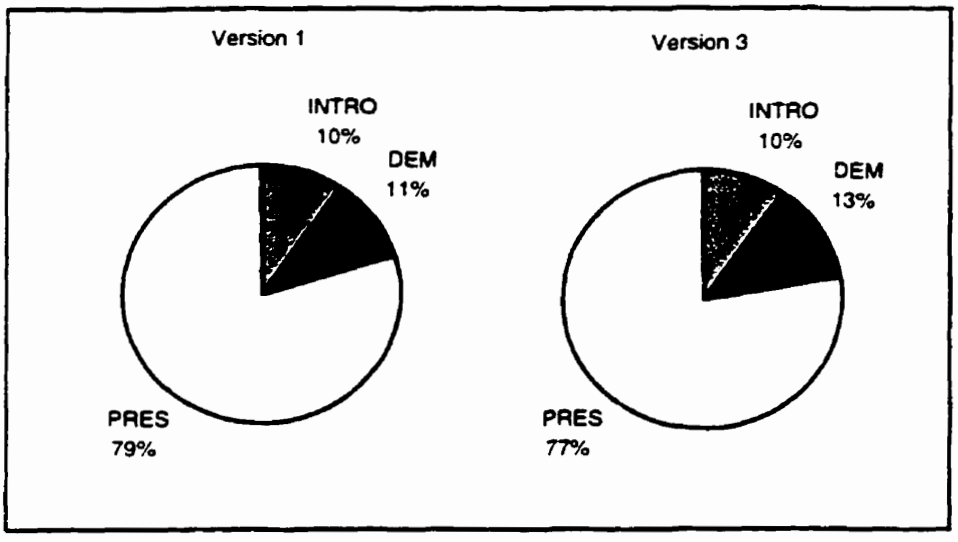

Figure 4.3 Fréquences relatives comparées des groupes de catégories de révision

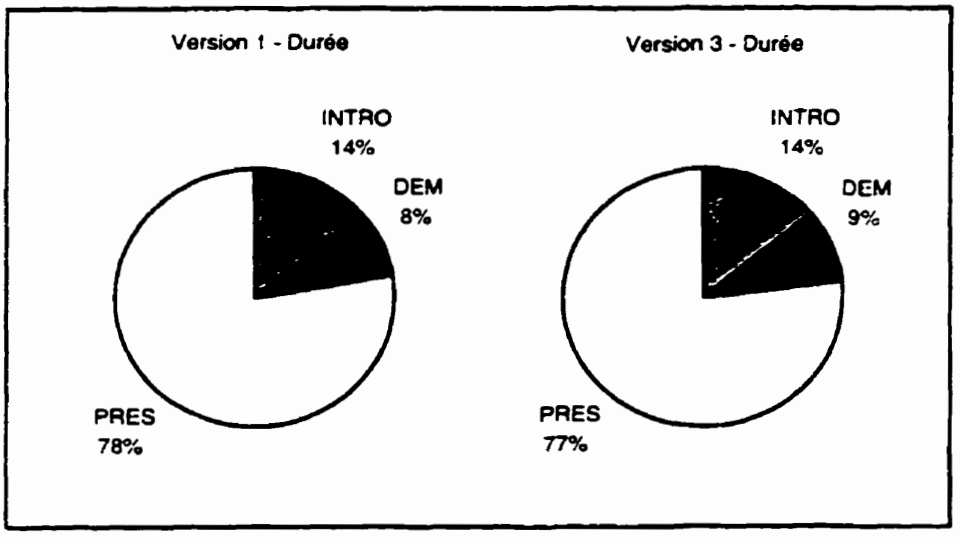

Figure 4.4 Durées relatives comparées des groupes de catégories de révision

Encore une fois, les différences entre les deux versions sont très minimes. Il y a au plus 2 % de différences au niveau des interventions et 1 % au niveau des durées. Les variations inter-codeurs sont donc beaucoup trop minimes pour invalider les analyses à ce point-ci.

D'Astous utilise ces résultats pour justifier la réécriture suivante: dans les analyses subséquentes, les interventions codées DEM seront transformées en leur sujet. Par exemple, une intervention codée DEM/EVA/... (une demande d'évaluation) sera interprétée comme une évaluation. Une deuxième modification sur les analyses est que les introductions sont ignorées. En effet, celles-ci ne sont utiles principalement que pour identifier les sujets de discussions par rapport à l'artefact étudié.

Une autre distinction est faite par d'Astous: des niveaux de conversation sont identifiés pour chaque intervention, en fonction de son sujet. Une intervention de niveau 0 est (généralement) l'introduction d'une section de l'artefact. Une intervention de niveau 1 à pour sujet une introduction (de niveau 0). Une intervention ayant pour sujet une intervention de niveau 1 est de niveau 2, et ainsi de suite. La distribution des niveaux de conversations, en fréquence et en durée, est décrite aux figures 4.5 et 4.6.

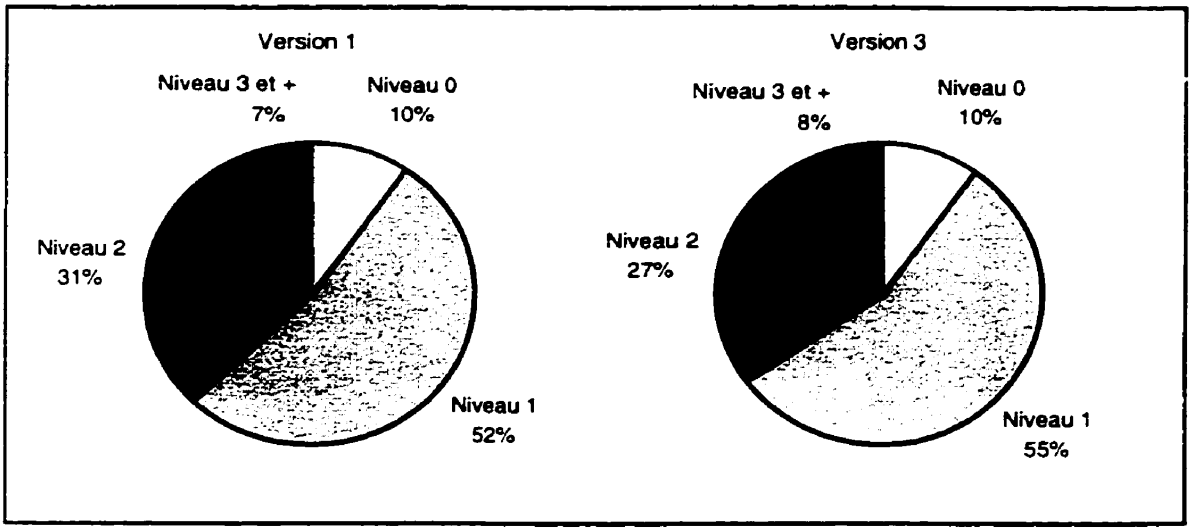

Figure 4.5 Fréquences relatives comparées des niveaux de conversation

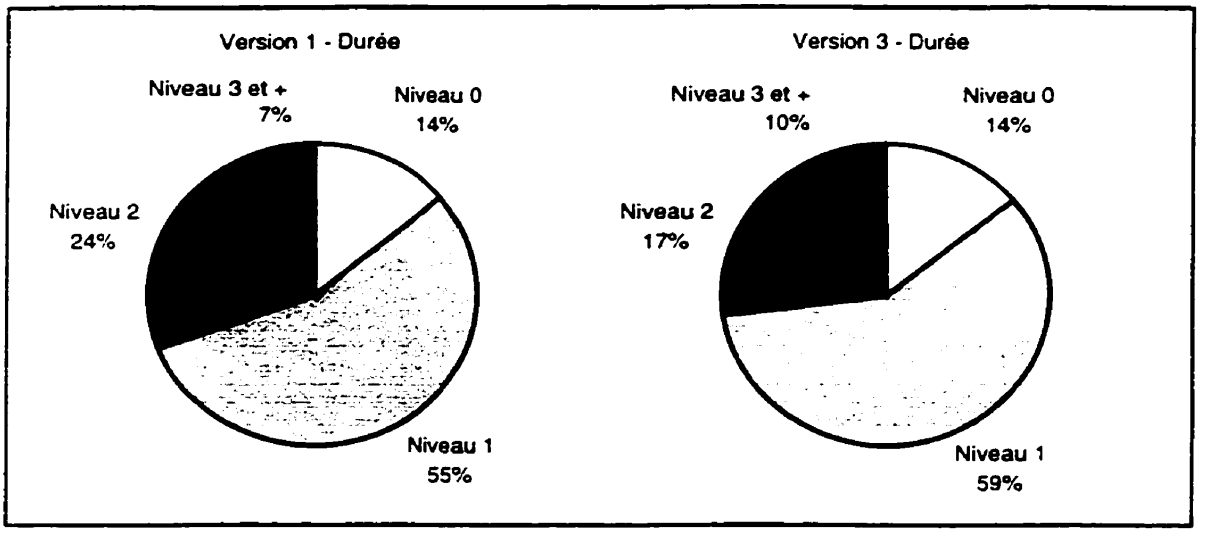

Figure 4.6 Durées relatives comparées des niveaux de conversation

Les deux figures montrent la même allure de distribution, mais les différences commencent à être plus importantes. Ce qui suit la logique que plus le degré de détail est important, plus les différences observées seront grandes. Les différences, pour les figures 4.5 et 4.6, ne sont toutefois pas vraiment importantes.

Les deux prochaines figures présentent la distribution des niveaux pour chaque code, pour chaque version.

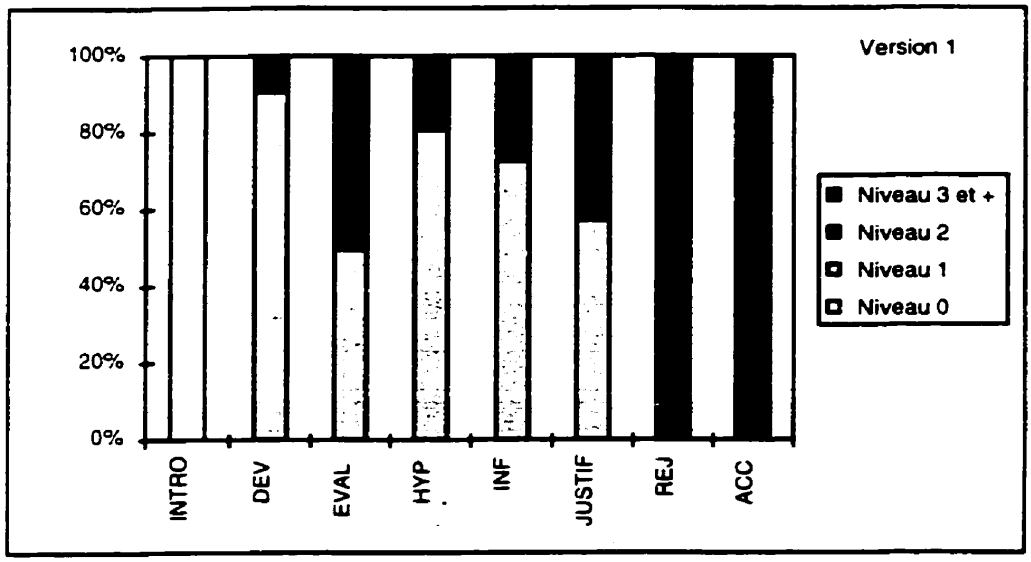

Figure 4.7 Fréquences relatives des niveaux d'interventions par catégorie - version 1

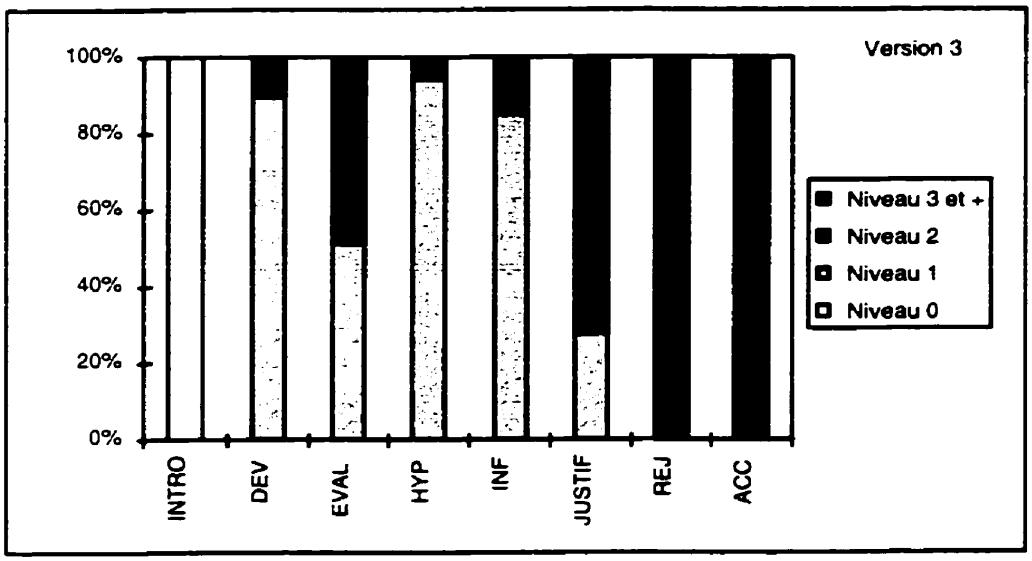

Figure 4.8 Fréquences relatives des niveaux d'interventions par catégorie - version 3

Les figures 4.7 et 4.8 montrent évidemment des différences plus importantes, ce qui était attendu. Mais ce qu'elles indiquent aussi est une remarquable similitude entre les deux versions, malgré le haut degré de détail de l'analyse. La moitié des codes n'ont aucune différence notable, et les autres ont presque tous des différences assez minimes. Le seul

cas problématique est JUSTIF, pour lequel la version 3 donne une plus grande proportion d'interventions au niveau 2 que la version 1, et donc moins d'interventions au niveau 1. Les fréquences exactes sont de 19 interventions de niveau 1 et 6 interventions de niveau 2 pour la version 1, comparativement à 8 interventions de niveau 1 et 13 interventions de niveau 2 pour la version 3. L'interprétation à donner à cette différence n'est malheureusement pas claire.

La vue opposée de la même situation est présentée aux figures 4.9 et 4.10, montrant la distribution des catégories par niveau de conversation.

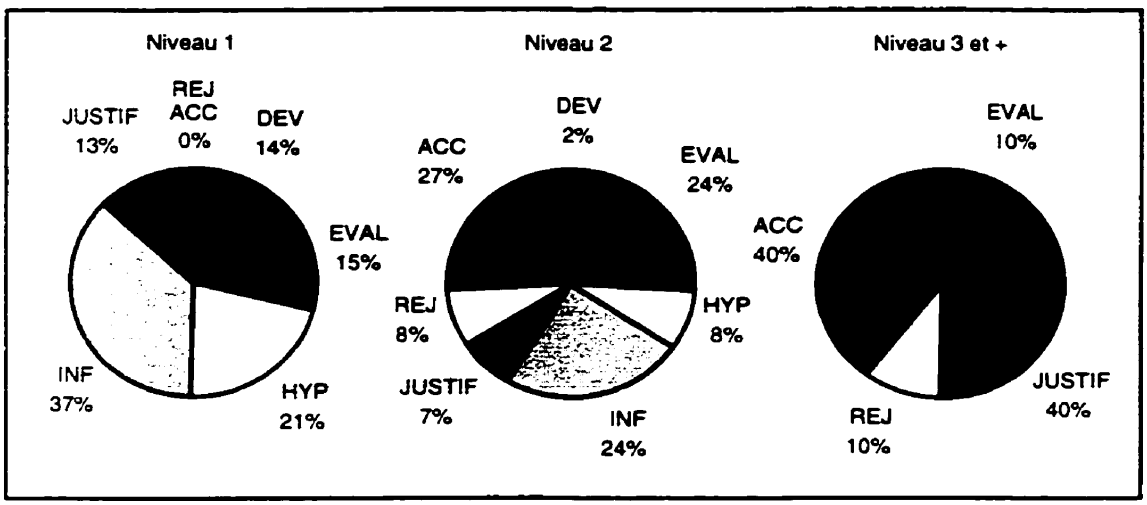

Figure 4.9 Fréquences relatives des catégories par niveau, version 1

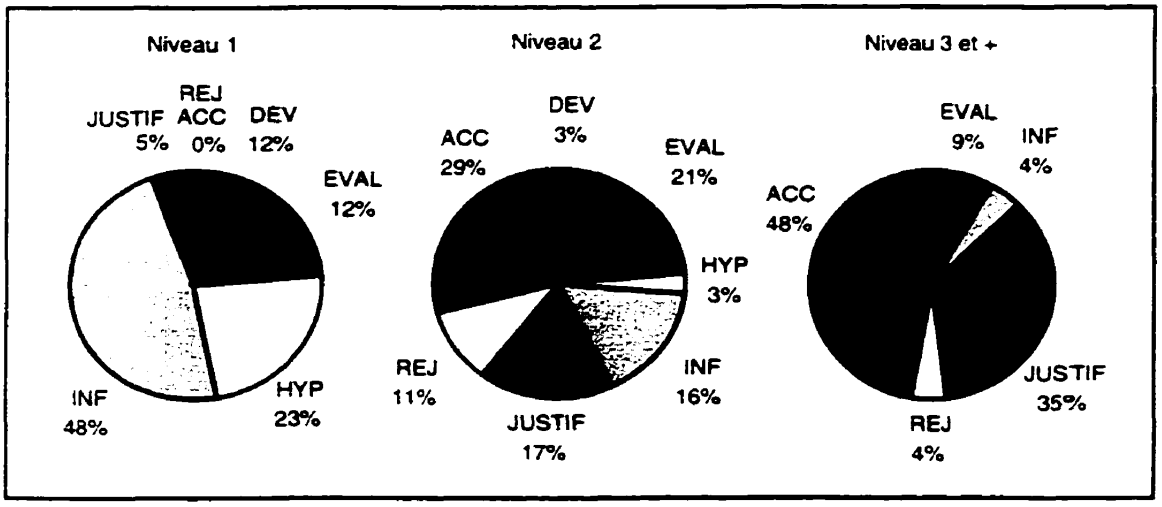

Figure 4.10 Fréquences relatives des catégories par niveau, version 3

Les mêmes différences pour JUSTIF sont observables dans ces deux dernières figures. Les répercussions semblent être sur le code INF. Ceci pourrait indiquer une différence de compréhension de la distinction entre une justification d'une intervention précédente et une information sur une intervention précédente.

L'analyse doit être arrêtée à ce point-ci, parce que la quantité de données est insuffisante pour supporter les analyses trop pointues. Seule une réunion est disponible pour comparer les deux versions, alors que sept ont été utilisées dans la thèse de d'Astous. Déjà la dernière analyse est fragile, puisque pour les niveaux 3 et plus, seuls 20 interventions étaient disponibles pour la version 1, et 23 pour la version 3. Une différence d'une intervention entraîne 5 % de variation sur le graphique, ce qui est considérable.

En somme, l'analyse statique fait ressortir bien peu de différences entre les analyses dues aux deux versions comparées. Les variations observées sont souvent le fruit de la sensibilité des analyses sur la taille de l'échantillon observé. Quelques différences sont plus sérieuses, mais semblent avoir peu d'impact sur l'interprétation qui peut être faite à ce point-ci de l'étude.

4.2 Analyse de la structure des échanges

L'analyse séquentielle de la structure des échanges est faite dans le but d'identifier les composantes du discours. Si un lien important existe entre deux codes, peut-être y a-t-il un patron de comportement significatif qui détermine au moins partiellement le déroulement d'une réunion.

L'approche utilisée dans la thèse de d'Astous est d'explorer la structure de la réunion et d'identifier les patrons clés ou les catégories qui, par leur comportement séquentiel, apportent peu à l'analyse. Les données sont ensuite réécrites pour éliminer les catégories inutiles et l'analyse séquentielle est répétée.

Les patrons de comportements sont étudiés au moyen du Lag Sequential Analysis (LSA), qui sera décrit à la section 4.2.3.2. Cet outil statistique est puissant mais requiert une bonne quantité de données. La réunion comparée n'est pas suffisante pour utiliser le LSA et répéter les analyses. Un moyen de contourner ce problème est de simuler le codage des autres réunions en reproduisant les différences observées à la réunion 02.

Cette section présentera donc la simulation des données introduite, une mesure du niveau de structure des réunions indiquant le niveau d'interprétation à donner aux résultats trouvés, et les résultats comme tels, trouvés à l'aides des nouvelles données.

Mais, avant de continuer plus loin, il est important de présenter les réécritures de données utilisées par d'Astous, pour tenter d'éliminer le bruit inhérent aux données. Les 11 catégories incluent des codes utiles pour la complétude du schéma de codage, mais peu utiles dans l'analyse. Ceux-ci sont les codes AUT et GES, auxquels s'ajoute le code DEM qui est transformé en son sujet. Ensuite, les codes de niveau 2 et supérieur sont éliminés parce que ces niveaux "ne semblent pas avoir d'impact direct sur la structure des

discussions de niveau 1." (d'Astous, 1999, p. 155). Ceci a pour effet de retrancher les codes ACC et REJ, puisqu'il n'y a aucune acceptation ou rejet directement sur l'artefact, seulement sur une intervention antérieure, comme le montrent les figures 4.7 à 4.10. Il ne reste plus que six codes: DEV, EVA, JUS, HYP, INF et INT. Enfin, avant d'examiner la structure en profondeur, les interventions successives de même catégorie sont regroupées.

4.2.1 Simulation de données

Pour mesurer l'impact des variations de codage entre deux codeurs sur l'analyse de la structure des échanges, il faut utiliser le LSA puisque c'est l'outil utilisé dans la thèse de d'Astous. Deux moyens sont disponibles pour utiliser le LSA: avoir une aussi grande quantité de données que d'Astous, ou simuler des données en respectant les différences observées à la réunion 02 pour obtenir deux ensembles de données de même taille. Les contraintes du projet, particulièrement la difficulté inhérente au codage de toutes les réunions, ont limité la possibilité de ce choix. La simulation de données semble donc être le moyen le plus efficace pour l'étude de la structure des échanges.

Un autre fait est à noter: les données des six autres réunions n'ont pas été codées par une seule et même personne. En effet, suivant la mesure de l'accord au chapitre 2, les deux codeurs avaient été montrés suffisamment en accord pour que le codage d'un codeur ou d'un autre soit utilisé sans distinction. Un codeur a donc codé les réunions 05a, 10a, 10b et 10c, et l'autre a codé les réunions 05b et 05c. La simulation cherche à compléter les deux versions, en simulant le codage de la version 1 des réunions 05b et 05c, et le codage des réunions 05a et 10 de la version 3. Le processus sera décrit dans ses grandes lignes, en simulant des données pour la version 3 à partir de réunions codées dans la version 1, mais le processus doit être répété pour chaque réunion, et donc parfois dans le sens inverse.

Le processus de simulation de données a été conçu pour être conservateur, en intégrant de l'aléatoire à tous les endroits appropriés. Cet ajout du hasard entraîne une diminution du potentiel de structure et, de cette manière, toute similitude entre les deux versions sera intéressante. Les hypothèses sur lesquelles repose l'analyse de la structure sont donc que la réunion 02 est représentative des autres réunions, et donc que les différences inter-codeurs observées seraient similaires dans toutes les autres réunions. De plus, il est évidemment supposé que la simulation de données n'introduit pas de dépendances cachées dans les données et génère des données fidèles à celles de la réunion 02. Et les codeurs sont présumés constants, leur codage sont supposés homogènes. Les analyses dépendent de la validité de ces hypothèses.

La description du processus de simulation de données est maintenant présentée. Au départ, les données existantes sont analysées. Plus précisément, la matrice d'accord entre les versions 1 et 3 est étudiée. Le tableau 4.1 présente les données étudiées suite aux réécritures diverses utilisées. En comparant ce tableau avec le tableau 2.1, on remarque que les éléments du tableau peuvent avoir un nombre inférieur d'interventions, ce qui est dû à l'élimination des niveaux supérieurs à 1. Les éléments peuvent aussi avoir un nombre supérieur d'interventions dans le cadre de la réécriture des codes DEM en leur sujet, ajoutant des interventions à ces codes du sujet. Il est intéressant de noter que ces réécritures, faites suivant la thèse de d'Astous, entraînent une augmentation du coefficient d'accord kappa de Cohen de 0.68 à 0.74, et de κ / κ_{max} de 0.75 à 0.82.

Tableau 4.1 Fréquences d'associations des codes entre la version 1 et la version 3

	Version 3						
Version 1	DEV	EVA	JUS	HYP	INF	INT	
DEV	11	3		2	4		20
EVA		15		2	4		21
JUS		2	11	1	5		19
HYP	5	1		19	3		28
INF			2	1	50		53
INT						27	27
	16	21	13	25	66	27	168

Pour générer les données de la version 3, les proportions de désaccords entre chaque couple de codes sont notées. Par exemple, pour le code DEV de la version 1, 15 % des nouveaux codes de la version 3 seront des codes EVA, soit 3 sur 20. Les proportions sont ensuite transformées en proportions cumulatives, pour une raison qui sera expliquée plus loin. Ceci résulte en la matrice des proportions d'associations entre codes, du tableau 4.2.

Tableau 4.2 Proportions cumulatives d'associations des codes entre la version 1 et la version 3

	Version 3					
Version 1	DEV	EVA	JUS	HYP	INF	INT
DEV	0.55	0.7	0.7	0.8	1	1
EVA	0	0.71	0.71	0.81	1	1
JUS	0	0.11	0.69	0.74	1	1
HYP	0.18	0.21	0.21	0.89	1	1
INF	0	0	0.04	0.06	1	1
INT	0	0	0	0	0	1

De plus, les moyennes et les écart types de la distribution de la durée des interventions sont notés pour chaque association de codes. Ils sont présentés à l'annexe A.

L'étape suivante identifie le nombre de chaque association entre codes pour les nouvelles données, à partir de la génération de nombres aléatoire. En particulier, ceci signifie qu'à chaque intervention, un nombre aléatoire de distribution de loi uniforme U(0,1) est

généré. Ce nombre est ensuite comparé aux valeurs de la rangée appropriée du tableau 4.2, pour identifier entre quelles valeurs il est situé. Le code associé à la valeur est alors temporairement choisi comme nouveau code. Par exemple, pour une intervention codée DEV dans la version 1, on génère un nombre aléatoire de 0.63, ce qui entraîne que le nouveau code temporaire est EVA, puisque 0.63 est situé entre 0.55 et 0.7. Ainsi, il est possible de générer des données respectant la distribution des désaccords entre codeurs.

Cependant, pour être plus fidèle aux divergences de la réunion 02, il est important de prendre en compte plus que la seule distribution des désaccords. Le chapitre 3 a identifié la durée des interventions comme un facteur ayant un effet marquant sur l'accord. Il est donc important de tenter de reproduire l'effet de ce facteur dans les nouvelles données.

Pour ce faire, la présence de chaque nouveau code temporaire est conservé, c'est-à-dire que s'il y a 4 DEV et 2 EVA comme nouveaux codes associés aux 6 codes DEV de la version 1 (exemple fictif), alors ces quantités seront conservées dans le nouveau codage. Ces 6 nouveaux codes seront toujours associés à un code DEV dans la version 1, mais l'intervention à laquelle chacun sera assigné sera déterminée en fonction de la durée des interventions.

La procédure est décrite ici et un exemple est disponible à l'annexe B, que le lecteur est invité à suivre en parallèle pour faciliter la compréhension. Or donc, une nouvelle série de nombres aléatoires est générée pour chaque couple de codes. La quantité de nombres générés est égale au nombre de nouveaux codes identifiés précédemment (2 codes EVA dans les 6 codes DEV de la version 1). Mais la série de nombres aléatoires suit maintenant une loi normale de moyenne égale à la moyenne des durées du couple de codes (durée de 8 mots dans l'exemple) et d'écart type égal à l'écart type du couple de codes (de 1.73 dans l'exemple). Il y a donc autant de séries de nombres aléatoires que de couples de codes, chaque série a autant de nombres aléatoires que la quantité associée au

couple trouvée plus haut, et chaque série de nombres aléatoires suit une loi normale N(*moyenne des durées du couple, écart type des durées du couple*).

Pour chaque nombre de chaque série, un identificateur est associé spécifiant le nouveau code du couple de codes de la série (EVA ici). Les séries sont ensuite combinées en une série par code original, combinant ainsi les nouveaux codes pour chaque code original. Chaque nouvelle série est ensuite triée dans l'ordre croissant des durées simulées, en conservant l'identificateur associé à chaque nombre. Et les interventions à coder sont aussi triées, tout d'abord par code, puis par durée croissante. C'est à ce point-ci que les identificateurs des nouveaux codes sont associés dans l'ordre aux interventions à coder, et forment ainsi les nouveaux codes simulés. Le tout peut alors être trié par ordre d'interventions, pour retrouver l'ordre original.

Cet ensemble de manipulations permet d'assurer que, peu importe les durées des interventions à coder, le comportement des accords et désaccords de chaque code en fonction de la durée *relative* de l'intervention sera respecté. L'intervention la plus courte codée DEV dans la version 1 sera probablement codée EVA dans la version 3, puisque la distribution de la durée pour EVA est plus petite et varie moins que celle pour DEV; la courbe normale de EVA serait plus à gauche que celle de DEV si les deux courbes étaient tracées. Le comportement en fonction de la durée est assuré en plus du respect de la distribution des associations entre codes.

Il est important de noter que la structure séquentielle n'est pas du tout prise en compte par la simulation de données. Ceci signifie que du hasard est inséré dans les séquences d'interventions, et donc que toute ressemblance entre les deux versions peut être considérée comme étant forte et donc très intéressante.

Pour uniformiser les effets du hasard, 10 simulations de données ont été faites pour chaque réunion. Des semences (*seed*) ont été données comme point de départ des

fonctions de génération de nombres aléatoires. Ces semences sont: 169, 65432, 11, 4096, 20807, 19223, 95034, 5756, 28713, 96409. Les 6 dernières sont tirées d'une table de nombres aléatoires.

Il y a donc 10 ensembles de données simulant la version 3 à partir des différences observées dans la réunion 02, et 10 autres simulant la version 1. Les données simulées des réunions sont ensuite jointes aux données réelles: la version 1 a des données réelles pour les réunions 02, 05a et 10, et des données simulées pour 05b et 05c, alors que la version 3 a des données réelles pour les réunions 02, 05b et 05c, et des données simulées pour 05a et 10. Ce sont ces données regroupées qui sont utilisées dans les analyses subséquentes. Les deux versions forment donc deux ensembles de données équivalents. Un troisième ensemble peut être étudié, celui utilisé dans les analyses de d'Astous, comprenant uniquement les données réelles de chaque réunion, et les données de la version 1 pour la réunion 02.

4.2.2 Mesure du niveau de structure

La structure d'une réunion, ou plus précisément d'une *séquence* telle que définie au premier chapitre, est importante pour comprendre le déroulement et pour modéliser une réunion. Un modèle simple montrerait les activités comme une suite standard d'activités, ou encore comme un graphe orienté d'activités. Une modélisation mathématique simple, très puissante et répandue est disponible grâce aux chaînes de Markov. Celles-ci sont très utiles en psychologie pour obtenir une idée préliminaire du modèle mathématique représentant un phénomène (Chatfield, 1973). La correspondance entre les données et les chaînes de Markov est mesurée par l'utilisation du test du χ^2 (chi-carré). Ces deux notions seront maintenant expliquées plus en détail.

4.2.2.1 Chaînes de Markov

Une chaîne de Markov, telle que définie par Cullmann (1975), est un système susceptible de prendre n états parmi $E = \{E_0, E_1, \ldots, E_i, \ldots, E_j, \ldots, E_{n-1}\}$. Une transition d'un état E_i à un autre E_j ne dépend que de ces deux états, selon la probabilité conditionnelle $P(E_j \mid E_i) = p_{ij}$. La chaîne peut donc être caractérisée par une matrice de transitions stochastiques $M = [p_{ij}]$, où la somme des p_{ij} sur i est 1. Le concept fut introduit au début du siècle par Andrei Andreyevich Markov, un des fondateurs de l'étude des processus stochastiques.

Une autre définition couramment utilisée d'une chaîne de Markov est la suivante, donnée par Isaacson et Madsen (1976). Un processus stochastique $\{X_k\}$ dont, pour tout n et pour tout état i parmi E, on a que $P[X_n=i_n \mid X_{n-1}=i_{n-1}, X_{n-2}=i_{n-2}, \ldots, X_1=i_1] = P[X_n=i_n \mid X_{n-1}=i_{n-1}]$, respecte la propriété de Markov et est donc une chaîne de Markov. Ceci revient à dire que la probabilité d'être dans l'état courant ne dépend que de l'état précédent, ou encore que la suite du processus, étant donné le présent, est indépendante du passé (Leon-Garcia, 1994).

Une chaîne de Markov, telle que définie dans le paragraphe précédent, est d'ordre 1. L'ordre d'une chaîne est simplement le nombre d'étapes précédentes déterminant l'état courant de la chaîne (Gottman et Roy, 1990). Donc, une chaîne d'ordre 3 détermine le prochain état de la chaîne à partir de l'état courant et des deux états précédents.

On peut facilement représenter une chaîne de Markov par un graphe orienté d'états, où une flèche d'un état à un autre existe si la probabilité de la transition est non nulle. On ajoute généralement les probabilités de transition aux flèches.

Les chaînes de Markov sont utilisées fréquemment dans les sciences sociales, pour modéliser des processus sociaux, plus particulièrement des séquences d'actions ou d'interactions.

4.2.2.2 Test du chi-carré

Pour mesurer la concordance entre un modèle et les données recueillies, on utilise des tests statistiques. Un de ces tests est le test du χ^2, qui mesure l'écart entre les fréquences observées et les fréquences attendues par le modèle (Scherrer, 1984). Il a été inventé en 1900 par Karl Pearson, un des pionniers des statistiques. Le test permet de déterminer si l'écart "est suffisamment faible pour être imputable aux fluctuations d'échantillonnage" (Moore et McCabe, 1993). L'écart peut montrer que le modèle ne décrit pas bien les données ou il peut montrer le contraire. La précision du test est proportionnelle à la quantité de données et est presque exact lorsque cette quantité tend vers l'infini (Chatfield et Lemon, 1970).

Le processus de test du χ^2 est défini comme suit (Moore et McCabe, 1993): une hypothèse H_0 est formulée (par exemple, il n'y a aucune association entre les données observées et le modèle théorique), ainsi que son hypothèse alternative complémentaire H_1 (il y a un lien entre les données et le modèle). Puis, on compare chaque fréquence observée à sa valeur attendue en calculant la valeur du χ^2:

$$\chi^2_{obs.} = \sum_i \frac{\left(f_{obs_i} - f_{th_i}\right)^2}{f_{th_i}} \qquad (4.1)$$

où f_{obs_i} est la fréquence observée pour l'élément i, f_{th_i} est la fréquence attendue de la théorie pour l'élément i (Scherrer, 1984). Cette définition du test fait ressortir une caractéristique importante: le test est additif. Donc, plus la quantité de données est importante, plus la valeur obtenue est grande et plus le test est conservateur (Cochran, 1952)

La statistique du χ^2, simple comparaison entre les fréquences observées et les fréquences attendues, fonctionne comme suit (Moore et McCabe, 1993). La différence directe entre chaque paire de fréquences est calculée puis mise au carré, rendant toutes les valeurs positives ou nulles. Puisqu'une grande différence n'est pas aussi significative quand la fréquence attendue est aussi grande, on divise tout simplement le carré de la différence par la fréquence attendue. Ceci est répété pour chaque élément de l'ensemble et le χ^2_{obs} est la somme des χ^2 individuels. Si les fréquences sont très différentes entre l'observation et la théorie, χ^2_{obs} sera très grand et indiquera que H_0 n'est pas très probable.

La valeur critique pour le test est aussi déterminée, habituellement à partir d'une table de statistique ou à partir de la distribution elle-même. La valeur critique dépend de deux facteurs: le degré de liberté $df = (r\text{-}1)^2$, où r est le nombre de catégories étudiées, et α, associé au niveau de confiance désiré. Le degré de liberté détermine la forme de la distribution: le χ^2 s'obtient à partir d'une loi normale N(df, $2df$), quand df est grand.

La valeur calculée de $\chi^2_{obs.}$ est ensuite comparée à la valeur critique. Si $\chi^2_{obs.}$ est inférieure à la valeur critique, l'hypothèse H_0 est acceptée et H_1 est refusée, et vice versa (Moore et McCabe, 1993). On dira que le modèle décrit les données à un niveau de confiance de $1\text{-}\alpha$.

Les lecteurs remarqueront qu'il est possible d'utiliser le test du χ^2 pour mesurer l'accord entre deux codeurs. Un des deux codeurs est considéré comme déterminant la valeur théorique attendue, et l'autre est la valeur observée. Cependant, utiliser le test ainsi suppose que la mesure est dans une échelle intervalle, ce qui n'est pas le cas dans cette étude.

4.2.2.3 Mesure de l'ordre d'une chaîne de Markov

On peut utiliser le test du χ^2 pour déterminer si une séquence est une chaîne de Markov, et pour déterminer son ordre si c'est le cas. Divers mathématiciens ont participé à cette adaptation, les principaux étant Haldane (1939), Bartlett (1951), Hoel (1954), et Anderson et Goodman (1957).

On veut déterminer l'ordre d'une chaîne de Markov. Le test du χ^2 détermine la concordance entre la théorie et les observations. On doit donc choisir un modèle théorique pour comparer avec les données. On commencera par tester si les données résultent d'un phénomène complètement aléatoire (H_0), versus un phénomène markovien d'un ordre quelconque (H_1). Si H_0 est acceptée, on arrête de tester car les données ne sont pas une chaîne de Markov, il n'y a pas de structure inhérente aux données. Si H_0 est rejetée, alors on teste une seconde fois pour vérifier si les données obéissent à une chaîne de Markov de premier ordre (H_0), versus une chaîne de Markov d'un ordre supérieur (H_1). Si H_0 est acceptée, les données suivent une chaîne de Markov d'ordre 1, mais si H_0 est rejetée, alors on fait un troisième test pour voir si les données obéissent à une chaîne markovienne d'ordre 2, versus un ordre supérieur à 2. On continue le processus jusqu'à ce que l'ordre de la chaîne soit déterminé, que les données soient montrées séquentiellement aléatoires ou que les données soient insuffisantes pour continuer de tester.

En effet, la quantité de données requises croît avec l'ordre testé. Pour le test 1, on teste le lien entre deux codes, on vérifie si les couples de codes sont structurés. Mais, pour le test 2 sur l'ordre 1, on teste si les triplets de codes sont structurés, et ainsi de suite. Cependant, Chatfield et Lemon (1970) remarquent que le test du χ^2 est plus précis quand la quantité de données est grande (il est exact lorsque le nombre tend vers l'infini), et donc l'approximation du test s'appauvrit quand l'ordre testé s'accroît, pour une quantité

constante de données. Par exemple, pour tester l'ordre 1 avec les 6 catégories, $6^3 = 216$ triplets doivent être examinées.

Une quantité minimale généralement acceptée est de 5 à 10 éléments par couple ou triplet (Cochran, 1952, 1954). Mais Halperin a montré que, dans l'étude de chaînes d'ordres 1 ou 2, le test est pratiquement valide pour de faibles quantités de données (Guthrie et Youssef, 1970). À plus forte raison, une importante étude de Roscoe et Byars (1971) a montré que 2 éléments par couple ou triplet sont amplement suffisants, peu importe l'allure de la répartition des données. Donc, si on examine l'ordre 1 avec 6 catégories, au moins $216 \times 2 = 432$ triplets doivent être disponibles (soit à peu près autant d'interventions), répartis d'une manière assez uniforme parmi tous les triplets possibles. Notons qu'environ $6^4 \times 2 = 2592$ interventions sont requises pour le test de l'ordre 2. Ceci étant dit, la description du test est nécessaire avant de continuer plus loin.

Le test du χ^2 est adapté pour ne contenir que des variables dérivables des observations. Tout ce qui peut être déduit d'une séquence de codes est la fréquence $n_{ij}(t)$ de liens entre un code i au temps t-1 suivi d'un code j au temps t. À partir de cette fréquence, on définit les fréquences suivantes:

$$n_{ij} = \sum_t n_{ij}(t) \quad , \quad n_i = \sum_j n_{ij} \quad , \quad n_j = \sum_i n_{ij} \quad , \quad N = \sum_{i,j} n_{ij} \tag{4.2}$$

Alors, pour l'hypothèse nulle H_0: les données sont statistiquement indépendantes (d'ordre zéro), par rapport à l'hypothèse alternative H_1: le processus est une chaîne de Markov d'ordre un, le test est:

$$\chi^2 = \sum_{i,j} n_i \frac{\left(n_{ij}/n_i - n_j/N \right)^2}{n_j/N} \tag{4.3}$$

avec $(r\text{-}1)^2$ degrés de liberté. Pour H_0: la chaîne de Markov est d'ordre un, vs. H_1: la chaîne est d'ordre deux, le test est:

$$\chi^2 = \sum_{i,j,k} n_{ij} \frac{\left(n_{ijk}/n_{ji} - n_{jk}/n_{j} \right)^2}{n_{jk}/n_{j}} \qquad (4.4)$$

avec $r(r\text{-}1)^2$ degrés de liberté (Suppes et Atkinson, 1960, pp. 56-57).

Il est important de noter comment sont calculés les degrés de liberté dans le test χ^2 de l'ordre d'une chaîne de Markov. Soit le test de l'ordre y vs. l'ordre y-1, avec s catégories. Alors le nombre de degrés de liberté est $df = s^{y-1}(s\text{-}1)^2$ (Gottman et Roy, 1990, p. 58). Cependant, il est possible que certains éléments (couples, triplets) ne soient pas possibles en fonction du design expérimental. Par exemple, si les interventions successives ayant le même code sont regroupées, alors la diagonale de la matrice i, j comprendra uniquement des zéros. Ces zéros sont qualifiés de structurels, et modifient l'interprétation du test. Goodman (1968, 1983) spécifie que le degré de liberté doit être réduit du nombre de zéros structurels dans la chaîne.

Dans les chaînes étudiées, il y a 11 zéros structurels dans le premier test: 6 pour les éléments de la diagonale, qui sont impossibles à atteindre puisque les éléments successifs identiques sont regroupés, et les 6 éléments qui ont une INTRO comme successeur sont aussi éliminés, pour conserver l'intérêt de l'étude sur les séquences, ce qui donne 11 puisqu'un élément est commun aux deux groupes de zéros structurels. Ainsi, pour les 6 catégories analysées et pour le premier test, il y a 25 degrés de liberté au départ, auxquels on retire les 11 zéros structurels. Il y a donc 14 degrés de liberté. Ces mêmes raisons donnent 106 degrés de liberté pour le deuxième test.

4.2.2.4 Résultats du niveau de structure

Concrètement, l'utilisation d'un test d'ordre requiert la série de tests décrite plus haut. Pour chaque test, H_0 associe un ordre y à la chaîne et H_1 associe un ordre supérieur à y. La valeur obtenue par l'application de l'équation appropriée est comparée à la valeur critique, déterminée par le degré de liberté du test. Par exemple, pour un niveau de confiance de 95 % et 14 degrés de liberté, la valeur critique est d'environ 24 (ou 29 à 99 % de confiance), et pour 106 degrés de liberté, la valeur critique est d'environ 131 (ou 143 à 99 % de confiance). Si la valeur calculée est supérieure à la valeur critique, l'hypothèse alternative est acceptée et par conséquent l'hypothèse nulle est rejetée. Sinon, on ne peut pas rejeter l'hypothèse nulle à ce niveau de confiance.

Le premier résultat intéressant est celui du premier test, où les données sont testées pour l'indépendance statistique versus une chaîne de Markov d'un ordre quelconque. Pour tous les ensembles de données étudiés, ce test donne un résultat très fort confirmant que les données forment une chaîne de Markov d'un ordre 1 ou supérieur. Par exemple, les données réelles utilisées dans les analyses de d'Astous et Robillard ont une valeur du test 1 de 263, très nettement au-dessus de la valeur critique de 24. Le niveau de confiance est de $1 - 5.33 \times 10^{-48}$, ce qui nous suffit amplement pour affirmer que les données étudiées forment une chaîne de Markov.

Ensuite, le deuxième test est également important, puisqu'il compare comme modèles une chaîne de Markov d'ordre 1 et une chaîne d'un ordre 2 ou supérieur. Quoique moins radicalement significatif, il donne un résultat de 153 avec les données réelles, nettement supérieur à la valeur critique de 131. Le niveau de confiance est ici de 99.8 %. Pour ces deux tests, la quantité de données est suffisante, puisqu'elle est supérieure à la limite de Roscoe et Byars (1971) de 2 éléments par couple ou triplet.

Les tests pour les versions 1 et 3 séparément sont plus complexes. En effet, il y a 10 ensembles de données à étudier pour chaque version. Tel que mentionné, le test 1 confirme fortement pour toutes les simulations que les données forment une chaîne de Markov, à un niveau de confiance supérieur à $1 - 10^{-31}$ pour tous les ensembles de données.

Cependant, l'image est moins claire pour le test 2. Pour la version 1, tous les ensembles rejettent l'hypothèse H_0 que la chaîne est d'ordre 1 à plus de 97.5 % de certitude. Pour la version 3, en moyenne, le test permet de rejeter l'hypothèse nulle à 96 % de certitude. Mais il y a quelques simulations pour lesquelles la structure semble être plus forte et que le niveau de confiance n'est que de 90 % environ. À l'inverse, 2 cas ont un niveau de confiance de plus de 99.5 %. Néanmoins, même si on ne peut pas rejeter H_1 avec autant de confiance, l'acceptation de H_0 n'est pas nécessairement certaine. Globalement, les deux versions sont probablement des chaînes de Markov d'ordre 2 ou plus.

Tester l'ordre 2 n'est malheureusement pas possible, étant donné que la quantité de données requise pour l'utilisation du test est trop importante pour cette étude. Il n'y a qu'environ 600 interventions disponibles par ensemble de données, après les réécritures. Et la concaténation des ensembles de données de chaque version n'est pas possible puisque ceci insérerait un biais dans le test, en répétant des longues séquences d'interventions et ainsi augmentant le degré de structure. Il n'est donc pas possible de tester si les chaînes sont d'ordre 2 par rapport à un ordre supérieur, et donc, puisque les chaînes ne sont pas d'ordre 1, il n'est pas possible de savoir l'ordre exact des chaînes de Markov. Tout ordre supérieur à 1 peut caractériser les chaînes étudiées.

Ces informations sont importantes pour l'interprétation des modèles qui suivent. Une chaîne d'ordre 1 est intuitive et très simple à interpréter, mais une chaîne d'ordre 2 est déjà plus complexe à modéliser. L'ordre des chaînes n'est pas connu, et il n'est pas

possible d'utiliser des outils de modélisation particuliers à un ordre précis, ce qui aurait facilité la visualisation.

À partir des tests effectués dans cette section, il est possible de restreindre l'interprétation des modèles de structure pour empêcher de lire les modèles séquentiellement. Les différences qui seront observées entre les deux versions ne sont pas nécessairement fortes si elles n'impliquent qu'un lien ne modifiant pas l'analyse qui en dépend. La section 4.2.4 illustrera ce point à l'aide des modèles obtenus des données.

4.2.3 Modélisation de la structure

La modélisation d'une structure cherche à identifier quels liens entre deux codes sont importants. Cette identification sert à cerner les patrons d'échanges, tels qu'ils sont décrits au tableau 6.1 de la thèse de d'Astous.

De nombreux chercheurs ont contribué au développement de ce domaine. Parmi eux, on note Cooke et al. (1996), qui ont proposé la méthode PRONET permettant de représenter facilement des données séquentielles à l'aide de graphes. Applbaum (1988) a synthétisé divers modèles de structure dans les réunions de prises de décisions, tout comme Katzenstein (1996), qui a étudié un niveau plus élevé. Et divers facteurs affectant la structure des réunions ont été étudiés: la familiarité par Gruenfeld et al. (1996), le raisonnement collectif par Resnick et al. (1993), et le comportement cognitif par Street et Cappella (1985), entre autres. De plus, un important travail de linguistique de Winograd et Flores (1986; Winograd, 1987) examine la structure générique d'une conversation à partir d'actes de langages. Lubich (1995) résume d'autres modèles du genre.

Les structures brutes des deux versions sont tout d'abord présentées. Mais une analyse plus poussée est nécessaire. Le Lag Sequential Analysis est utilisé pour déterminer si un

lien est statistiquement significatif. Cet outil sera donc décrit et les résultats ainsi obtenus seront présentés par après.

4.2.3.1 Structures préliminaires

Avant d'utiliser le LSA pour identifier les liens séquentiels significatifs, il est important d'avoir une idée de la structure brute entre les codes. Ceci est fait en calculant les proportions de liens directs, c'est-à-dire que pour chaque couple de codes, le nombre de liens entre les deux codes est divisé par la fréquence du code d'origine. Pour l'ensemble de données originales, composé uniquement de données réelles, le tableau 4.3 présente ces proportions.

Tableau 4.3 Proportions de liens séquentiels entre chaque code, données originales

	Suivant					
Précédent	DEV	EVA	JUS	HYP	INF	INT
DEV	0	0.59	0.13	0.12	0.16	0
EVA	0.45	0	0.22	0.11	0.22	0
JUS	0.16	0.27	0	0.20	0.38	0
HYP	0.21	0.04	0.17	0	0.57	0
INF	0.15	0.32	0.10	0.44	0	0
INT	0.21	0.41	0.06	0.09	0.22	0

Pour les versions 1 et 3, les proportions de liens ne peuvent être obtenues directement, puisqu'il y a 10 simulations des données pour chaque version. Chaque simulation étant de même taille, il est valide de les regrouper en deux grands ensembles sans perte de généralité. Une simple addition des fréquences de chaque lien séquentielle est effectuée, à partir de laquelle les tableaux 4.4 et 4.5 de proportions peuvent être obtenus. La comparaison des trois tableaux montre que presque tous les couples de codes ont des proportions différentes et que les différences sont souvent assez faibles (en moyenne de 0.048) mais peuvent parfois être plus fortes (jusqu'à 0.17 pour DEV-EVA). Les différences observées sont nécessairement réparties de telle manière à ce que le total de

chaque ligne soit toujours 1. Bien sûr, on doit tenir compte qu'il est possible que, suite aux approximations, certaines lignes totalisent 0.99 ou 1.01.

Tableau 4.4 Proportions de liens séquentiels entre chaque code, version 1

	Suivant					
Précédent	DEV	EVA	JUS	HYP	INF	INT
DEV	0	0.42	0.18	0.22	0.17	0
EVA	0.39	0	0.22	0.19	0.20	0
JUS	0.18	0.25	0	0.22	0.34	0
HYP	0.23	0.13	0.18	0	0.46	0
INF	0.18	0.28	0.11	0.44	0	0
INT	0.20	0.36	0.07	0.15	0.21	0

Tableau 4.5 Proportions de liens séquentiels entre chaque code, version 3

	Suivant					
Précédent	DEV	EVA	JUS	HYP	INF	INT
DEV	0	0.50	0.09	0.17	0.24	0
EVA	0.36	0	0.18	0.12	0.35	0
JUS	0.13	0.25	0	0.14	0.48	0
HYP	0.22	0.12	0.11	0	0.56	0
INF	0.18	0.33	0.12	0.36	0	0
INT	0.22	0.35	0.05	0.09	0.28	0

Pour mieux interpréter les différences entre les tableau 4.3 à 4.5, d'autres outils d'analyse sont nécessaires. Une représentation graphique de la situation est possible. Une manière simple est d'utiliser un graphe orienté avec un poids sur chaque arête, où le poids est la proportion présentée dans un des tableaux; ceci est utilisé par Gottman et Bakeman (1979). Cependant, cette manière n'est pas d'une grande utilité puisqu'elle n'augmente pas la lisibilité des résultats.

Une méthode plus appropriée est la transformation de la largeur des arêtes en fonction du poids de l'arête. Cette méthode est utilisée par Vortac et al. (1994), et est représentée aux figures 4.11 et 4.12. Les proportions sont classées par dizaines, où une ligne pointillée représente un lien occurrent moins de 10 % des fois, une ligne mince sans pointillés est

associée à un lien entre 0.1 et 0.2, et chaque largeur supplémentaire augmente de 10 % la force du lien.

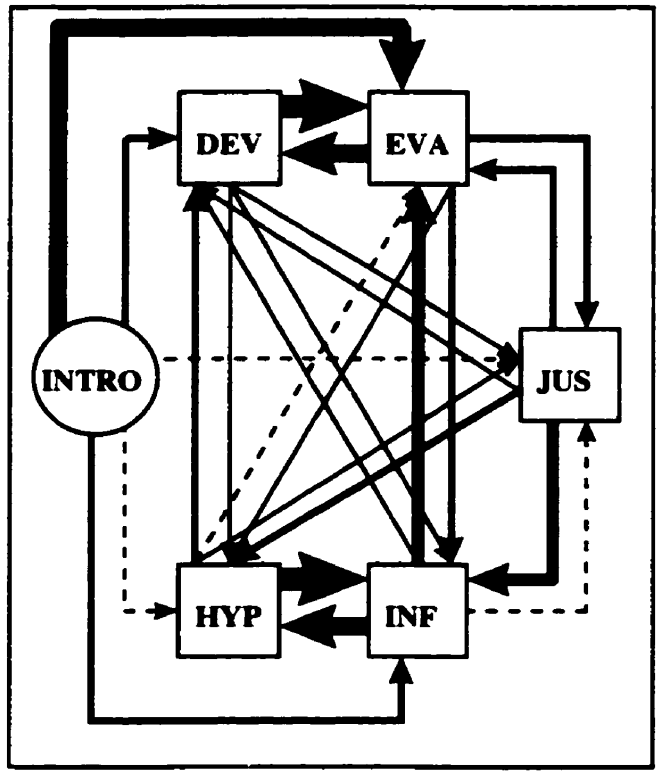

Figure 4.11 Structure des échanges, proportions brutes, données originales

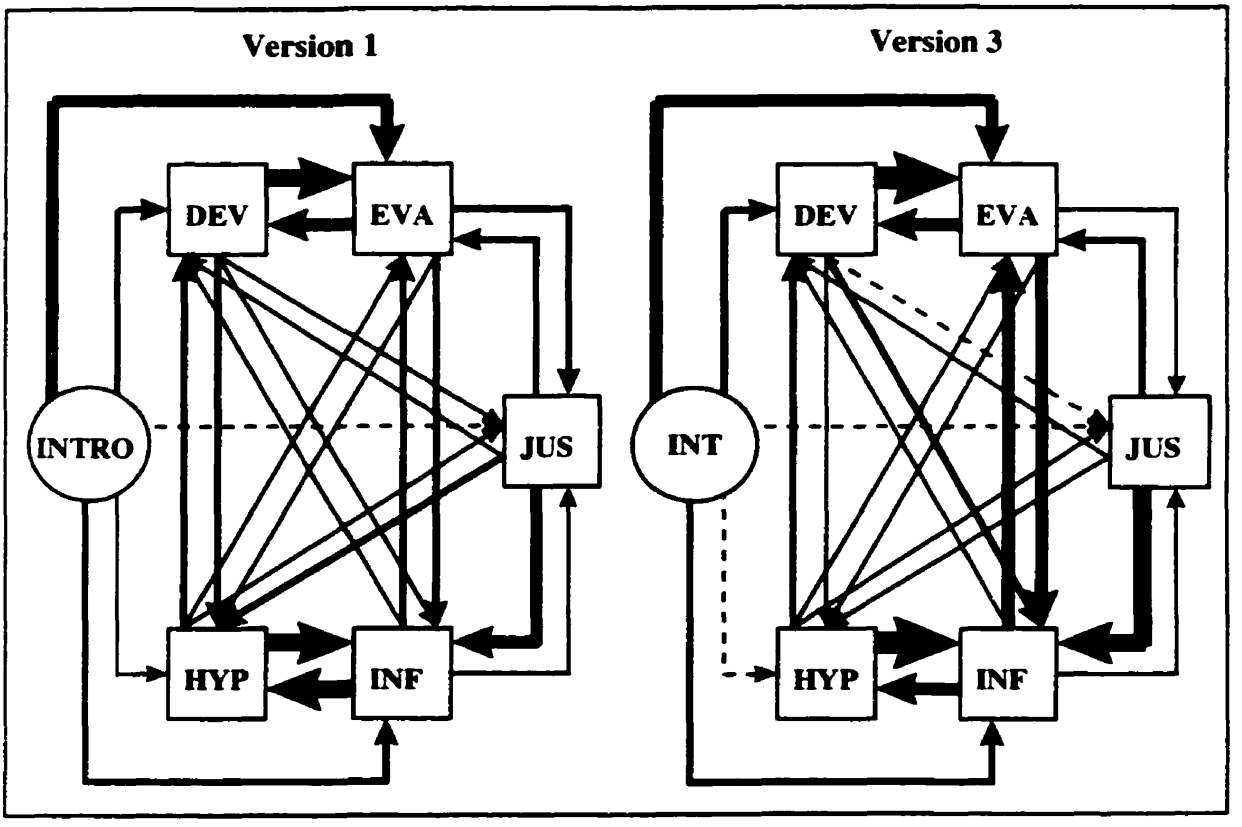

Figure 4.12 Structure des échanges, proportions brutes, versions 1 et 3

Les deux figures font ressortir la ressemblance de la structure des trois versions. Les liens principaux sont respectés dans les trois graphes: INT-EVA, DEV-EVA, EVA-DEV, HYP-INF, INF-HYP, JUS-INF ainsi que INF-EVA. Bien sûr, certaines lignes ont une largeur différente, mais l'impact ne semble pas nécessairement important sur la structure globale des échanges.

Une petite curiosité peut être notée: dans la version 3, le code INF est plus souvent successeur d'un autre code que dans la version 1. Cette observation est aussi possible en comparant les tableaux 4.4 et 4.5, où le total de la colonne des codes INF est de 1.38 dans la version 1 et de 1.91 dans la version 3. L'explication de ce phénomène pour INF n'est pas connue et ne sera pas hasardée.

L'observation des proportions brutes n'adapte pas les liens pour tenir compte de telles curiosités, mais ceci devrait être fait pour avoir une solide idée statistique de la structure. L'utilisation d'un outil statistique plus approprié est donc nécessaire à ce point-ci. Le Lag Sequential Analysis est un outil qui tient justement compte des situations observées ici.

4.2.3.2 Lag Sequential Analysis

Le LSA fut développé principalement par Gene Sackett (1979), et enrichi par Bakeman (1978), Gottman (1980), et Allison et Liker (1982). Le but principal est justement de résumer les interactions entre les codes d'actions, en déterminant la dépendance séquentielle d'un code sur un autre. Faraone et Dorfman (1987) remarquent que l'outil est utilisé dans une optique d'analyse exploratoire des données, en ce sens qu'il est relativement simple, qu'il opère sur des données plutôt complexes et qu'aucune hypothèse n'est formulée avant l'analyse des données. Utiliser le LSA permet de contourner le problème du manque de données pour trouver la structure markovienne (Gottman et Roy, 1990).

Le fonctionnement du LSA est simple. Il est adapté de la cote z, qui est une standardisation d'un résultat pour trouver sa signification (Moore et McCabe, 1993, p. 455). La cote z calcule la distance d'un ensemble de résultats par rapport à la moyenne attendue, et corrige cette distance en la divisant par l'écart type de l'échantillon. L'équation 4.5 présente la formule standard d'une cote z d'un test de signification.

$$z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}} \tag{4.5}$$

Le LSA vise à établir la signification d'un lien entre deux codes. Allison et Liker (1982) ont proposé l'équation 4.6, pour adapter l'équation 4.5 à la situation de séquences de codes nominaux. Cette équation compare le lien entre les codes *A* et *B*.

$$z = \frac{p_{A|B} - p_A}{\sqrt{\frac{p_A(1-p_A)(1-p_B)}{Np_B}}} \qquad (4.6)$$

Les équations 4.5 et 4.6 sont similaires. La distance entre la probabilité du lien *A-B* et la probabilité de *A* en général est calculée, puis corrigée par l'écart type lié à la distribution des codes *A* et *B*. Cet écart type est basé sur l'approximation normale d'une distribution binomiale.

L'interprétation du résultat du z de l'équation 4.6 est la même que l'interprétation de toute cote z, et fonctionne de la même façon que l'interprétation du résultat d'un test du χ^2. La valeur du z est calculée et comparée à une valeur critique, qui dépend du niveau de confiance désiré. À 95 % de confiance, la valeur critique est 1.96 et, à 99 % de confiance, la valeur critique est 2.576. Si la cote z pour le lien *A-B* est supérieure à la valeur critique, le lien est significatif à un certain niveau de confiance, sinon le lien n'est pas significativement plus présent que ce qui serait attendu par le hasard.

Plusieurs variations sur l'équation existent, bien sûr celle de Sackett (1979), mais aussi une adaptation de Yates, de Cohen (1977) et de Overall (1980). Une analyse comparative de Cousins et al. (1986) a cependant montré que l'équation de Allison et Liker (1982) est une excellente estimation de la distribution réelle. Son utilisation est de plus très répandue pour modéliser toute séquence de situations quelconques.

Par ailleurs, comme le titre d'un article de Gottman (1980) l'indique, le LSA peut être utilisé pour mesurer la fiabilité entre codeurs. Au lieu d'étudier les couples successifs d'interventions, les couples inter-codeurs sont étudiés. Un problème avec cette utilisation du LSA est qu'il requiert beaucoup de données codées deux fois, puisque chaque couple

de codes doit avoir une quantité supérieure au niveau minimum. Aussi, il donne des résultats au niveau des couples de codes, mais aucun index global ne peut être obtenu.

4.2.3.3 Structures résultantes du LSA

L'application du LSA donne un ensemble de cotes z signifiant le degré de certitude de la signification du lien séquentiel. Le tableau 4.6 présente ces résultats pour les données originales non simulées. Les cases ombragées ont une valeur qui dépassent la valeur critique de 1.96 associée à 95 % de certitude, et sont donc considérées comme étant significatives.

Tableau 4.6 Application du LSA sur les données originales

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.59	**4.90**	-0.42	-2.29	-2.63	-6.19
EVA	**5.99**	-6.50	**2.49**	-2.55	-1.56	-7.30
JUS	-0.08	0.70	-2.34	0.69	**2.56**	-3.95
HYP	1.66	-3.40	**2.26**	-3.61	**7.93**	-4.83
INF	-1.72	0.34	-1.41	**5.31**	-6.45	-7.33
INT	**4.89**	**9.79**	-0.001	0.04	**3.40**	-4.96

Une représentation graphique est aussi possible pour le LSA. L'ajout de poids aux arêtes n'est pas très utile dans ce cas-ci puisque les cotes z ne sont pas une mesure de la force du lien, mais uniquement de la certitude que le lien est significatif. La figure 4.13 présente ces données.

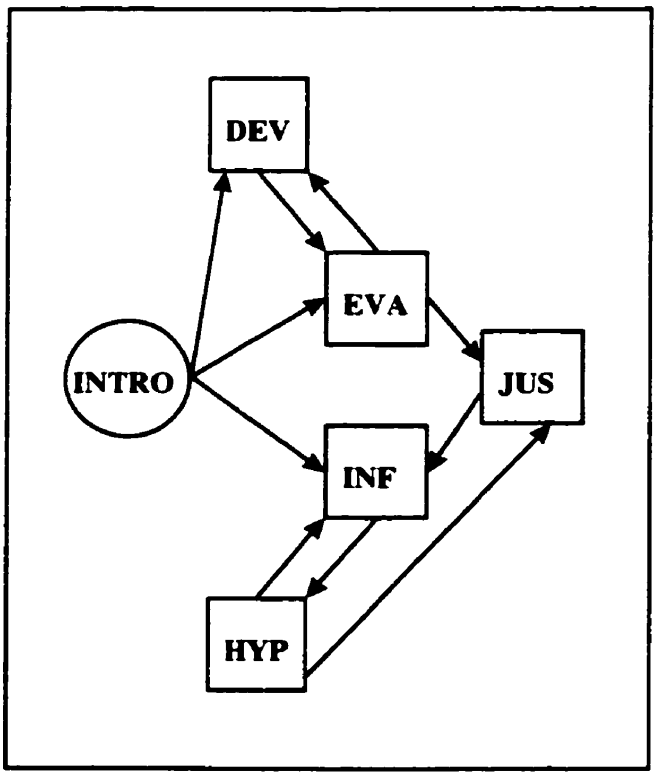

Figure 4.13 Structure d'après le LSA, données originales

Certaines observations sont possibles à partir de l'observation du tableau 4.6 et de la figure 4.13. Tout d'abord, certains liens sont beaucoup plus certains d'être significatifs que d'autres: INT-EVA a une cote z de 9.79, alors que la cote z de HYP-JUS n'est que 2.26. Ensuite, la force des liens telle que vue à la section 4.2.3.1 est dans le même ordre de grandeur que la taille de la cote z. Par exemple, les couples très forts listés plus haut sont presque tous significatifs avec un bon degré de certitude.

Les versions 1 et 3, encore une fois, sont plus complexes à comparer. L'addition pure des ensembles de données simulés n'est pas une option appropriée, puisqu'elle résulte en un ensemble de taille 10 fois supérieure à celle des données originales. Une telle modification de la taille a un impact important sur les analyses de signification, puisque tout lien accidentel est répété et peut être vu comme significatif. Le LSA n'est pas sensible à la quantité de données analysées, mais est sensible à l'indépendance relative

des sous-ensembles de données. En répétant 10 fois des séquences identiques représentant plus de la moitié des données, l'indépendance n'est plus du tout valide et le LSA ne peut pas être utilisé.

Pour étudier les structures des divers ensembles de données simulés, le LSA est appliqué à chaque ensemble de manière indépendante. Il en résulte 10 structures de liens significatifs par version, qui sont présentées à l'annexe C. Chaque structure d'une même version est comparée et combinée dans une même structure, et les liens communs à toutes les structures de simulation sont identifiés. La figure 4.14 présente les structures combinées de chaque version. Les liens gras sont communs à toutes les simulations.

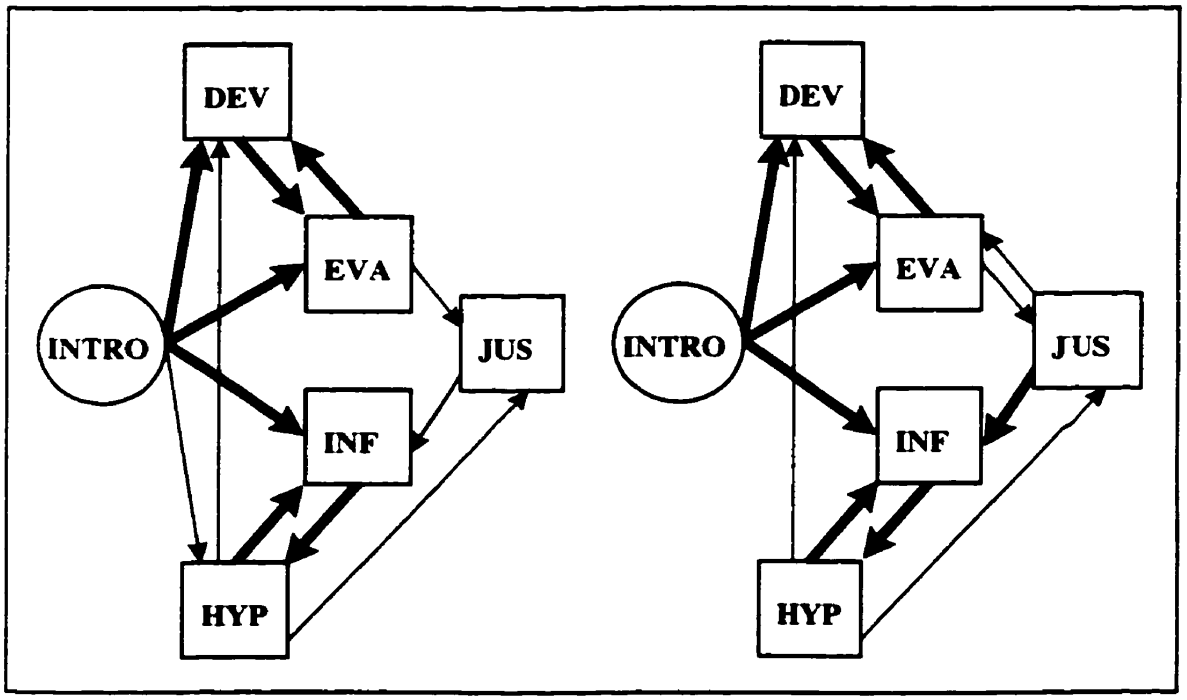

Figure 4.14 Comparaison des structures du LSA, versions 1 et 3

L'observation de cette structure fait ressortir deux similitudes importantes. D'abord, les liens communs aux simulations d'une même version sont presque identiques dans les deux versions. Ensuite, les liens qu'on pourrait qualifier de secondaires sont aussi très

semblables. Un seul lien commun est différent entre les deux versions (JUS-INF), et deux liens secondaires sont différents: INT-HYP dans la version 1 et JUS-EVA dans la version 3.

Une telle similitude entre les deux versions, et entre les simulations en général, est remarquable puisqu'une portion importante des données est générée sans tenter de respecter les liens séquentiels de la réunion 02. L'introduction du hasard à chaque étape importante de la simulation, et l'attention mise uniquement sur la distribution des désaccords et sur la durée des interventions, fait qu'une structure résultante aussi forte est soit liée simplement à ces facteurs, soit simplement assez forte pour être considérée fiable.

4.2.4 Analyse des résultats

Cette section 4.2 analysait l'impact des différences de codage entre codeurs. Les données ont été comparées à l'aide de trois outils statistiques: les proportions brutes de liens entre deux codes, le niveau de structure déterminé à l'aide de tests d'ordre markovien, et la structure résultante du Lag Sequential Analysis.

Chaque outil fourni un regard différent et complémentaire sur les données. Les proportions brutes donnent fidèlement la structure obtenue à l'aide des divers ensembles de données étudiés, alors que le LSA interprète les séquences et en extrait les liens qui sont statistiquement plus importants que ce que le hasard aurait donné. Ces deux points de vue sont importants dans l'analyse. L'impact sur les analyses de d'Astous est uniquement au niveau du LSA, mais il est important d'avoir un aperçu juste de la structure brute pour qu'elle soit utilisée comme contexte à l'interprétation du LSA.

Les structures brutes des trois ensembles de données apparaissent assez semblables entre elles. Les proportions ont des différences assez minces, d'en moyenne 4.8 %. Sur les figures, les différences apparaissent mais la structure globale ne semble pas tellement différente entre les différents ensembles de données.

Le LSA donne un portrait encore plus intéressant. Tout d'abord, le bruit est éliminé, c'est-à-dire que les liens faibles ne sont pas pris en compte, ce qui rend la structure plus claire. Ensuite, la puissance statistique du LSA donne une base solide de comparaison des liens entre eux. La présence d'un lien dans tous les ensembles de données est une indication que ce lien est très fort dans les échanges codés. Par exemple, les proportions et le degré de certitude du LSA pour les liens DEV-EVA et EVA-DEV, et HYP-INF et INF-HYP démontrent à quel point ces deux couples de codes sont importants. Par contre, les liens de INT à DEV ou INF ne semblent pas si importants du point de vue des proportions brutes, mais le LSA montre qu'ils sont significatifs.

Les différences entre les structures de l'analyse séquentielle sont peu nombreuses. Les liens communs à chaque simulation d'une version sont les mêmes à une exception près. Et ce lien ajouté dans la version 3 est présent dans plus de la moitié des simulations de la version 1. Les liens secondaires, présents dans plusieurs mais pas toutes les simulations d'une version, sont aussi très semblables. Seuls deux liens secondaires sont différents, ceux-ci sont les deux seuls liens différents si tous les liens sont pris en compte.

Évidemment, une seule différence dans une structure peut modifier grandement l'interprétation qui en est faite. Par exemple, si un graphe était divisé en deux groupes de codes disjoints dans une version et liés dans l'autre, la situation semblerait très différente. Cependant, la mesure du degré de structure indique qu'une telle lecture des structures est inappropriée. En effet, celle-ci supposerait que les échanges étudiés forment une chaine de Markov d'ordre 1, ce qui n'est pas le cas. L'ordre n'a pu être mesuré précisément, sauf

pour indiquer qu'il est supérieur ou égal à 2. Même l'ordre 2 ne peut pas être confirmé, puisque la quantité de données est insuffisante pour le tester.

L'utilisation des analyses séquentielles de d'Astous vise l'identification de patrons d'échanges, principalement basés sur les liens entre DEV et EVA, et entre HYP et INF. Ces patrons peuvent aussi être identifiés à l'aide des structures du LSA calculées dans ce projet. Ces deux couples de codes ont à la fois des liens très forts dans les graphes de structure brute et des liens significatifs très certains dans le LSA. L'impact des différences de codage est donc pratiquement nul sur l'identification de patrons d'échange.

4.3 Résumé du chapitre

L'impact de l'utilisation d'un codeur différent sur l'analyse des réunions a été étudié dans ce chapitre. Deux analyses ont été effectuées, une première statique concernant la distribution des codes selon divers points de vue et une seconde séquentielle cherchant à identifier des patrons d'échange dans les enchaînements d'interventions.

Les deux analyses ont révélé des différences faibles mais nombreuses entre les deux versions. Ceci n'est pas surprenant, puisque le codage est différent dans plus du quart des interventions. Toute analyse numérique précise est certaine d'avoir des décimales différentes quand le niveau de détail étudié est grand. Ceci est démontré dans les analyses statiques portant sur des notions pointues, où la quantité de données est relativement faible.

À un niveau plus global, l'impact mesuré des différences entre codeurs est faible sur les analyses. Par conséquent, il est possible d'affirmer que l'utilisation du codage d'un codeur ou d'un autre dans les analyses a peu d'impact sur les résultats, en fonction du niveau d'accord mesuré dans le chapitre 2. De plus, les analyses d'impact révèlent que les

analyses de d'Astous ne sont pas inutiles, en ce sens qu'elles dépendent directement des données, qu'elles sont sensibles aux variations de données. Si deux analyses avaient donné exactement les mêmes résultats pour les deux versions du codage, leur valeur aurait paru faible. Ceci n'est pas le cas, l'impact étant montré à la fois présent et faible en conséquences.

CONCLUSION

Ce projet visait une compréhension approfondie des concepts et des méthodes de validation d'expériences à caractère humain en génie logiciel. Il a été vu qu'étant donné l'imprécision et la variabilité du domaine humain, la mesure de la fiabilité demeure problématique. Plusieurs outils et méthodes existent pour estimer certains aspects de la fiabilité, mais aucune méthode ne permet de la mesurer avec précision.

Le génie logiciel est à un point où l'expérimentation sur des sujets plus "mous" est rendue nécessaire. On doit s'assurer d'être capable de faire face à ce défi avec les bons outils pour estimer la fiabilité d'une expérience puisqu'il est facile qu'une erreur s'insère dans une étape de l'expérience sans être détectée.

La contribution de ce projet est à la fois éducative et pratique. En effet, ce projet a tout d'abord introduit et discuté des concepts importants à la base de la validation expérimentale. D'après ce qui a été vu de la littérature, aucune autre publication n'a présenté cette problématique pour l'étude des aspects humains du génie logiciel.

Ensuite, le projet a fourni un exemple de la démarche nécessaire lors de la validation d'une expérience. Il a aussi donné un aperçu des outils disponibles dans la validation d'expériences impliquant un système de mesure nominal, système courant dans les études sur les aspects humains.

Enfin, ce projet a permis de déterminer que la méthode utilisée par d'Astous et Robillard est fiable. Le deuxième chapitre a montré que la méthode d'analyse est objective et le

dernier chapitre a montré que les résultats sont stables, en ce sens qu'ils ne sont pas très influencés par les différences d'interprétation des codeurs.

Les résultats présentés sont contraints par les diverses hypothèses posées tout au long de ce projet. Parmi celles-ci, la plus importante est l'hypothèse que la réunion 02 est représentative de toutes les réunions du projet. De plus, le projet suppose que les deux versions de codages sont représentatives de l'ensemble des codages qui auraient pu être obtenus par des codeurs formés également.

Quatre voies paraissent particulièrement intéressantes pour continuer ce projet. Il serait d'abord intéressant que les codeurs complètent le codage des six réunions codées une seule fois, pour pouvoir comparer deux ensembles complets de codes à l'aides des outils utilisés dans le dernier chapitre, sans avoir recours à la simulation de données. Ensuite, il serait intéressant qu'une troisième personne code la réunion 02 (ou toutes les réunions) pour la comparer avec les deux versions étudiées dans ce projet. Aussi, il serait très utile d'utiliser la démarche et les outils du projet pour tenter la validation d'une expérience similaire déjà validée, pour valider la démarche utilisée et le choix d'outils statistiques de ce projet. Une dernière possibilité d'étude serait la transcription et le codage des réunions non codées, pour augmenter la quantité de données disponibles et pouvoir mieux calculer l'ordre de la chaîne de Markov obtenue.

Ce projet montre qu'il est possible de valider une expérience étudiant des aspects plus humains du génie logiciel.

RÉFÉRENCES

ALLISON, P. D., LIKER, J. K. (1982) "Analyzing Sequential Categorical Data on Dyadic Interaction: A Comment on Gottman", *Psychological Bulletin*, Vol. 91, No 2, 393-403.

ANDERSON, T. W., GOODMAN, L. A. (1957) "Statistical Inference about Markov Chains", *Annals of Mathematical Statistics*, Vol. 28, 89-110.

APPLBAUM, R. L. (1988) "Structure in Group Decision Making", CATHCART, R. S., SAMOVAR, L. A. (eds.) "Small Group Communication, A Reader", Wm. C. Brown Publishers, Dubuque, Iowa, 174-184.

BAER, D. M. (1977) "Reviewer's Comment: Just Because It's Reliable Doesn't Mean That You Can Use It", *Journal of Applied Behavior Analysis*, Vol. 10, No 1, 117-119.

BAKEMAN, R. (1978) "Untangling Streams of Behavior: Sequential Analysis of Observation Data", SACKETT, G. P. (ed.) "Observing Behavior. Vol.2: Data Collection and Analysis Methods", University Park Press, Baltimore, 63-78.

BAKEMAN, R., GOTTMAN, J. M. (1986) "Observing Interaction: An Introduction to Sequential Analysis", Cambridge University Press, Cambridge, 221 p.

BALES, R. F. (1951) "Interaction Process Analysis: A Method for the Study of Small Groups", Addison-Wesley, Cambridge, 203 p.

BALES, R. F., GERBRANDS, H. (1948) "The Interaction Recorder: An Apparatus and Check List for Sequential Content Analysis of Social Interaction", *Human Relations*, Vol. 1, No 4, 456-463.

BANGE, P. (1992) "Analyse conversationnelle et théorie de l'action", Hatier-Didier, Paris, 223 p.

BARTLETT, M. S. (1951) "The Frequency Goodness of Fit Test for Probability Chains", *Proceedings - Cambridge Philosophical Society*, Vol. 47, 86-95.

BASAWA, I. V., PRAKASA RAO, B. L. S. (1980) "Statistical Inference for Stochastic Processes", Academic Press, London, 435 p.

BASILI, V. R., ROMBACH, H. D. (1988) "The TAME Project: Towards Improvement-Oriented Software Environments", *IEEE Transactions on Software Engineering*, Vol. 14, No 6, 758-773.

BASILI, V. R., SELBY, R. W., HUTCHENS, D. H. (1986) "Experimentation in Software Engineering", *IEEE Transactions on Software Engineering*, Vol. 12, No 7, 733-743.

BENNETT, E. M., ALPERT, R., GOLDSTEIN, A. C. (1954) "Communications Through Limited Response Questioning", *The Public Opinion Quarterly*, Vol. 18, No 3, 303-308.

BERK, R. A. (1979) "Generalizability of Behavioral Observations: A Clarification of Interobserver Agreement and Interobserver Reliability", *American Journal of Mental Deficiency*, Vol. 83, No 5, 460-472.

BIRKIMER, J. C., BROWN, J. H. (1979) "Back to Basics: Percentage Agreement Measures Are Adequate, But There Are Easier Ways", *Journal of Applied Behavior Analysis*, Vol. 12, No 4, 535-543.

BODEN, M. A. (1988) "Computer Models of Mind: Computational approaches in theoretical psychology", Cambridge University Press, Cambridge, 289 p.

BOEHM, B. W. (1976) "Software Engineering", *IEEE Transactions on Computers*, Vol. 25, No 12, 1226-1241.

BOURGEOIS, K. V. (1996) "Process Insights from a Large-Scale Software Inspections Data Analysis", *CrossTalk*, 14 p.
http://www.stsc.hill.af.mil/Crosstalk/1996/oct/xt96d10e.html

BRENNAN, R. L., PREDIGER, D. J. (1981) "Coefficient Kappa: Some Uses, Misuses, and Alternatives", *Educational and Psychological Measurement*, Vol. 41, 687-699.

BRILLIANT, S. S., KNIGHT, J. C. (1999) "Empirical Research in Software Engineering: A Workshop", *ACM SIGSOFT Software Engineering Notes*, Vol. 24, No 3, 45-52.

BROOKS, A., DALY, J., MILLER, J., ROPER, M., WOOD, M. (1994) "Replication's Role in Experimental Computer Science", Rapport technique RR/172/94 (EFoCS-5-94), University of Strathclyde, Glasgow, 14 p.

BROOKS, F. P. Jr. (1975) "The Mythical Man-Month: Essays on Software Engineering", Addison-Wesley, Reading, Massachusetts, 195 p.

BROWN, B. B., MENDENHALL, W., BEAVER, R. (1968) "The Reliability of Observations of Teachers' Classroom Behavior", *The Journal of Experimental Education*, Vol. 36, No 3, 1-10.

BUCK, F. O. (1981) "Indicators of Quality Inspections", Rapport technique 21.802, IBM Systems Products division, Kingston, NY.

CAMPBELL, D. T., FISKE, D. (1959) "Convergent and Discriminant Validation by the Multi-Trait, Multi-Method Matrix", *Psychological Bulletin*, Vol. 56, 81-105.

CHATFIELD, C. (1973) "Statistical Inference Regarding Markov Chain Models", *Applied Statistics*, Vol. 22, 7-20.

CHATFIELD, C., LEMON, R. E. (1970) "Analysing Sequences of Behavioural Events", *Journal of Theoretical Biology*, Vol. 29, 427-445.

CHI, M. T. H. (1997) "Quantifying Qualitative Analyses of Verbal Data: A Practical Guide", *The Journal of the Learning Sciences*, Vol. 6, No 3, 271-315.

COCHRAN, W. G. (1952) "The χ^2 Test of Goodness of Fit", *Annals of Mathematical Statistics*, Vol. 23, 315-345.

COCHRAN, W. G. (1954) "Some Methods for Strengthening the Common χ^2 Tests", *Biometrics*, Vol. 10, 417-451.

COHEN, J. (1960) "A Coefficient of Agreement for Nominal Scales", *Educational and Psychological Measurement*, Vol. 20, No 1, 37-46.

COHEN, J. (1968) "Weighted Kappa: Nominal Scale Agreement with Provision for Scaled Disagreement or Partial Credit", *Psychological Bulletin*, Vol. 70, No 4, 213-220.

COHEN, J. (1977) "Statisticai Power Analysis for the Behavioral Sciences", Academic Press, New York, 474 p.

CONE, J. D. (1977) "The Relevance of Reliability and Validity for Behavioral Assessment", *Behavior Therapy*, Vol. 8, 411-426.

CONGER, A. J. (1980) "Integration and Generalization of Kappas for Multiple Raters", *Psychological Bulletin*, Vol. 88, No 2, 322-328.

COOIL, B., RUST, R. T. (1994) "Reliability and Expected Loss: A Unifying Principle", *Psychometrika*, Vol. 59, No 2, 203-216.

COOIL, B., RUST, R. T. (1995) "General Estimators for the Reliability of Qualitative Data", *Psychometrika*, Vol. 60, No 2, 199-220.

COOKE, N. J., NEVILLE, K. J., ROWE, A. L. (1996) "Procedural Network Representations of Sequential Data", *Human-Computer Interaction*, Vol. 11, 29-68.

COUSINS, P. C., POWER, T. G., CARBONARI, J. P. (1986) "Power and Bias in the z Score: A Comparison of Sequential Analytic Indices of Contingency", *Behavioral Assessment*, Vol. 8, 305-317.

CRONBACH, L. J., GLESER, G. C., NANDA, H., RAJARATNAM, N. (1972) "The Dependability of Behavioral Measurements: Theory of Generalizability for Scores and Profiles", John Wiley & Sons, New York, 410 p.

CROOKALL, D., SAUNDERS, D. (1989) "Towards an Integration of Communication and Simulation", CROOKALL, D., SAUNDERS, D. (eds.) "Communication and Simulation: From Two Fields to One Theme", Multilingual Matters, Clevedon, England, 3-29.

CULLMANN, G. (1975) "Initiation aux chaînes de Markov: méthodes et applications", Masson et cie, Paris, 140 p.

CURTIS, B. (1980) "Measurement and Experimentation in Software Engineering", *Proceedings of the IEEE*, Vol. 68, No 9, 1144-1157.

CURTIS, B., KRASNER, H., ISCOE, N. (1988) "A Field Study of the Software Design Process for Large Systems", *Communications of the ACM*, Vol. 31, No 11, 1268-1287.

CURTIS, B., WALZ, D. (1990) "The Psychology of Programming in the Large: Team and Organizational Behaviour", HOC, J.-M., GREEN, T. R. G., SAMURÇAY, R., GILMORE, D. J. (eds.) "Psychology of Programming", Academic Press, London, 253-270.

DAFT, R. L., LENGEL, R. H. (1986) "Organizational Information Requirements, Media Richness, and Structural Design", *Management Science*, Vol. 32, No 5, 554-571.

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.

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", présenté à *International Conference on the Design of Cooperative Systems* (COOP'98), Cannes, France, 26-29 mai.

DÉTIENNE, F., VISSER, W. (1997) Réunions de travail, Projet ACGL, École Polytechnique de Montréal - INRIA.

DEWEY, M. E. (1983) "Coefficients of Agreement", *British Journal of Psychiatry*, Vol. 143, 487-489.

DUNN, G. (1989) "Design and Analysis of Reliability Studies: The Statistical Evaluation of Measurement Errors", Oxford University Press, New York, Edward Arnold, London, 198 p.

DUTOIT, A. A., BRUEGGE, B. (1998) "Communication Metrics for Software Development", *IEEE Transactions on Software Engineering*, Vol. 24, No 8, 615-628.

ERICSSON, K. A., SIMON, H. A. (1993) "Protocol Analysis: Verbal Reports as Data, Revised Edition", MIT Press, Cambridge, 443 p.

FAGAN, M. E. (1976) "Design and Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, Vol. 15, No 3, 182-210.

FARAONE, S. V., DORFMAN, D. D. (1987) "Lag Sequential Analysis: Robust Statistical Methods", *Psychological Bulletin*, Vol. 101, No 2, 312-323.

FENTON, N. (1994) "Software Measurement: A Necessary Scientific Basis", *IEEE Transactions on Software Engineering*, Vol. 20, No 3, 199-206.

FENTON, N., PFLEEGER, S. L., GLASS, R. L. (1994) "Science and Substance: A Challenge to Software Engineers", *IEEE Software*, Vol. 11, No 4, 86-95.

FLEISS, J. L. (1981) "Statistical Methods for Rates and Proportions, 2nd edition", John Wiley and Sons, New York, 321 p.

FLEISS, J. L., COHEN, J., EVERITT, B. S. (1969) "Large Sample Standard Errors of Kappa and Weighted Kappa", *Psychological Bulletin*, Vol. 72, No 5, 323-327.

FRICK, T., SEMMEL, M. L. (1978) "Observer Agreement and Reliabilities of Classroom Observational Measures", *Review of Educational Research*, Vol. 48, No 1, 157-184.

GERO, J. S., McNEILL, T. (1997) "An Approach to the Analysis of Design Protocols", *Design Studies*, Vol. 19, No 1, 21-61.

GILB, T. (1998) "Optimizing Software Inspections", *CrossTalk*, 7 p.
http://www.stsc.hill.af.mil/Crosstalk/1998/mar/optimizing.html

GILB, T., GRAHAM, D. (1993) "Software Inspection", Addison-Wesley, Reading, Massachusetts, 471 p.

GILMORE, D. J. (1990) "Methodological Issues in the Study of Programming", HOC, J.-M., GREEN, T. R. G., SAMURÇAY, R., GILMORE, D. J. (eds.) "Psychology of Programming", Academic Press, London, 83-98.

GOODMAN, L. A. (1968) "The Analysis of Cross-Classified Data: Independence, Quasi-Independence, and Interactions in Contingency Tables With or Without Missing Entries", *Journal of the American Statistical Association*, Vol. 63, 1091-1131.

GOODMAN, L. A. (1983) "A Note on a Supposed Criticism of an Anderson-Goodman Test in Markov Chain Analysis", KARLIN, S., AMEMIYA, T., GOODMAN, L. A. (eds.) "Studies in Econometrics, Time Series, and Multivariate Statistics", Academic Press, New York, 85-92.

GOODMAN, L. A., KRUSKAL, W. H. (1954) "Measures of Association for Cross Classifications", *Journal of the American Statistical Association*, Vol. 49, 732-764.

GOTTMAN, J. M. (1980) "Analyzing for Sequential Connection and Assessing Interobserver Reliability for the Sequential Analysis of Observational Data", *Behavioral Assessment*, Vol. 2, 361-368.

GOTTMAN, J. M., BAKEMAN, R. (1979) "The Sequential Analysis of Observational Data", LAMB, M. E., SOUMI, S. J., STEPHENSON, G. R. (eds.) "Social interaction analysis: Methodological issues", University of Wisconsin Press, Madison, 185-206.

GOTTMAN, J. M., ROY, A. K. (1990) "Sequential Analysis: A Guide for Behavioral Researchers", Cambridge University Press, Cambridge, 275 p.

GRADY, R. B. (1992) "Practical Software Metrics for Project Management and Process Improvement", Prentice Hall, Englewood Cliffs, 270 p.

GRUENFELD, D. H., MANNIX, E. A., WILLIAMS, K. Y., NEALE, M. A. (1996) "Group Composition and Decision Making: How Member Familiarity and Information Distribution Affect Process and Performance", *Organizational Behaviour and Human Decision Processes*, Vol. 67, No 1, 1-15.

GUTHRIE, D., YOUSSEF, M. N. (1970) "Empirical Evaluation of Some Chi-Square Tests for the Order of a Markov Chain", *Journal of the American Statistical Association*, Vol. 65, No 350, 631-634.

HA, L. (1998) "Advertising in Hong Kong under Political Transition: A Longitudinal Analysis", *Web Journal of Mass Communication Research*, Vol. 1, No 3.

HAGGARD, E. A. (1958) "Intraclass Correlation and the Analysis of Variance", Dryden Press, New York, 171 p.

HALDANE, J. B. S. (1939) "The Mean and Variance of χ^2, When Used as a Test of Homogeneity, when Expectations Are Small", *Biometrika*, Vol. 31, 346-355

HARTMANN, D. P. (1977) "Considerations in the Choice of Interobserver Reliability Estimates", *Journal of Applied Behavior Analysis*, Vol. 10, No 1, 103-116.

HARTMANN, D. P. (1982) "Assessing the Dependability of Observational Data", HARTMANN, D. P. (ed.) "Using Observers to Study Behavior", Jossey-Bass, San Francisco, 51-65.

HAWKINS, R. P., FABRY, B. D. (1979) "Applied Behavior Analysis and Interobserver Reliability: A Commentary on Two Articles by Birkimer and Brown", *Journal of Applied Behavior Analysis*, Vol. 12, No 4, 545-552.

HERBERT, J., ATTRIDGE, C. (1975) "A Guide for Developers and Users of Observation Systems and Manuals", *American Educational Research Journal*, Vol. 12, No 1, 1-20.

HERBSLEB, J. D., KLEIN, H., OLSON, G. M., BRUNNER, H., OLSON, J. S., HARDING, J. (1995). "Object-Oriented Analysis and Design in Software Project Teams", *Human-Computer Interaction*, Vol. 10, 249-292.

HOEL, P. G. (1954) "A Test for Markoff Chains", *Biometrika*, Vol. 41, 430-433

HOC, J.-M., GREEN, T. R. G., SAMURÇAY, R., GILMORE, D. J. (1990) "Psychology of Programming", Academic Press, London, 83-98.

HOLLENBECK, A. R. (1978) "Problems of Reliability in Observational Research", SACKETT, G. P. (ed.) "Observing Behavior. Vol.2: Data Collection and Analysis Methods", University Park Press, Baltimore, 79-98.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. (1990) "IEEE Standard Glossary of Software Engineering Terminology - Std 610.12-1990", IEEE.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. (1988) "IEEE Standard for Software Review and Audits - Std 1028-1988", IEEE.

ISAACSON, D. L., MADSEN, R. W. (1976) "Markov Chains Theory and Applications", John Wiley & Sons, New York, 256 p.

JAMES, L. R., DEMAREE, R. G., WOLF, G. (1984) "Estimating Within-Group Interrater Reliability With and Without Response Bias", *Journal of Applied Psychology*, Vol. 69, No 1, 85-98.

JAMES, L. R., DEMAREE, R. G., WOLF, G. (1993) "r_{wg}: An Assessment of Within-Group Interrater Agreement", *Journal of Applied Psychology*, Vol. 78, No 2, 306-309.

JOHNSON, S. M., BOLSTAD, O. D. (1973) "Methodological Issues in Naturalistic Observation: Some Problems and Solutions for Field Research", HAMERLYNCK, L. A., HANDY, L. C., MASH, E. J. (eds.) "Behavior Change: Methodology, Concepts and Practice", Research Press, Champaign, Illinois, 7-67.

JORDAN, B., HENDERSON, A. (1995) "Interaction Analysis: Foundations and Practice", *The Journal of the Learning Sciences*, Vol. 4, No 1, 39-103.

KAPLAN, A. (1964) "The Conduct of Inquiry: Methodology for Behavioral Science", Chandler, San Francisco, 428 p.

KATZENSTEIN, G. (1996) "The Debate on Structured Debate: Toward a Unified Theory", *Organizational Behaviour and Human Decision Processes*, Vol. 66, No 3, 316-332.

KAZDIN, A. E. (1977) "Artifact, Bias, and Complexity of Assessment: The ABCs of Reliability", *Journal of Applied Behavior Analysis*, Vol. 10, No 1, 141-150.

KERLINGER, F. N. (1973) "Foundations of Behavioral Research, 2nd edition", Holt, Rinehart and Winston, New York, 741 p.

KRATOCHWILL, T. R., WETZEL, R. J. (1977) "Observer Agreement, Credibility, and Judgment: Some Considerations in Presenting Observer Agreement Data", *Journal of Applied Behavior Analysis*, Vol. 10, No 1, 133-139.

KUDER, G. F., RICHARDSON, M. W. (1937) "The Theory of the Estimation of Test Reliability", *Psychometrika*, Vol. 2, 151-160.

LANDIS, J. R., KOCH, G. G. (1977) "The Measurement of Observer Agreement for Categorical Data", *Biometrics*, Vol. 33, 159-174.

LEON-GARCIA, A. (1994) "Probability and Random Processes for Electrical Engineering, 2nd edition", Addison-Wesley, Reading, Massachusetts, 596 p.

LETHBRIDGE, T. C. (1999) Discussions informelles, Séminaire *Measuring Success: Empirical Studies of Software Engineering*, 4-5 mars, NRC, Ottawa.

LETOVSKY, S., PINTO, J., LAMPERT, R., SOLOWAY, E. (1987) "A Cognitive Analysis of a Code Inspection", OLSON, G., SHEPPARD, S., SOLOWAY, E. (eds.) "Empirical Studies of Programming", Ablex Publishing, 231-247.

LUBICH, H. P. (1995) "Towards a CSCW Framework for Scientific Cooperation in Europe", Lecture Notes in Computer Science 889, Springer-Verlag, Berlin, 268 p.

MEDLEY, D. M., MITZEL, H. E. (1963) "Measuring Classroom Behavior by Systematic Observation", GAGE, N. L. (ed.) "Handbook of Research on Teaching", Rand McNally, Chicago, 247-328.

MEISTER, D. (1990) "Simulation and modeling", WILSON, J. R., CORLETT, E. N. (eds.) "Evaluation of human work: A practical ergonomics methodology", Taylor & Francis, London, 180-199.

MITCHELL, S. K. (1979) "Interobserver Agreement, Reliability, and Generalizability of Data Collected in Observational Studies", *Psychological Bulletin*, Vol. 86, No 2, 376-390.

MITROFF, I. I., TUROFF, M. (1975) "Philosophical and Methodological Foundations of Delphi", LINSTONE, H. A., TUROFF, M. (eds.) "The Delphi Method: Techniques and Applications", Addison-Wesley, Reading, Massachusetts, 17-36.

MOORE, D. S., McCABE, G. P. (1993) "Introduction to the Practice of Statistics, 2nd edition", W.H. Freeman, New York.

MYERS, G. J. (1978) "A Controlled Experiment in Program Testing and Code Walkthroughs / Inspections", *Communications of the ACM*, Vol. 21, No 9, 760-768.

NELSON, R. O., HAY, L. R., HAY, W. M. (1977) "Comments on Cone's "The Relevance of Reliability and Validity for Behavioral Assessment"", *Behavior Therapy*, Vol. 8, 427-430.

NEUMANN, P. G. (1994) "Computer-Related Risks", Addison-Wesley, Reading, Massachusetts, 367 p.

OLSON, G. M., HERBSLEB, J. D., RUETER, H. H. (1994) "Characterizing the Sequential Structure of Interactive Behaviors Through Statistical and Grammatical Techniques", *Human-Computer Interaction*, Vol. 9, 427-472.

OLSON, G. M., OLSON, J. S., CARTER, M. R., STORRØSTEN, M. (1992) "Small Group Design Meetings: An Analysis of Collaboration", *Human-Computer Interaction*, Vol. 7, 347-374.

OVERALL, J. E. (1980) "Continuity Correction for Fisher's Exact Probability Test", *Journal of Educational Statistics*, Vol. 5, 177-190.

PARNAS, D. L., WEISS, D. M. (1987) "Active Design Review: Principles and Practices", *Journal of Systems and Software*, Vol. 7, 259-265.

PATTERSON, G. R. (1982) "Coercive Family Process", Castalia Press, Eugene, Oregon.

PEARSON, K. (1900) "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling", *The Philosophical Magazine*, Series 5, Vol. 50, 157-172. .

PERREAULT, W. D. J., LEIGH, L. E. (1989) "Reliability of Nominal Data Based on Qualitative Judgments", *Journal of Marketing Research*, Vol. 26, 135-148.

PERRY, D. W., STAUDENMAYER, N. A, VOTTA, L. G. (1994) "People, Organizations and Process Improvement", *IEEE Software*, Vol. 11, No 4, 36-45.

PFLEEGER, S. L. (1995) "Experimental Design and Analysis in Software Engineering. Part 2: How to Set Up an Experiment", *ACM SIGSOFT Software Engineering Notes*, Vol. 20, No 1, 22-26.

PFLEEGER, S. L., JEFFERY, R., CURTIS, B., KITCHENHAM, B. (1997) "Status Report on Software Measurement", *IEEE Software*, Vol. 14, No 2, 33-43.

PORTER, A. A., JOHNSON, P. M. (1997) "Assessing Software Review Meetings: Results of a Comparative Analysis of Two Experimental Studies", *IEEE Transactions on Software Engineering*, Vol. 23, No 3, 129-145.

PORTER, A. A., SIY, H., MOCKUS, A., VOTTA, L. G. (1998) "Understanding the Sources of Variation in Software Inspections", *ACM Transactions on Software Engineering and Methodology*, Vol. 7, No 1, 41-79.

PORTER, A. A., SIY, H., TOMAN, C. A., VOTTA, L. G. (1997) "An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development", *IEEE Transactions on Software Engineering*, Vol. 23, No 6, 329-346.

PORTER, A. A., SIY, H., VOTTA, L. G. (1996) "A Review of Software Inspections", ZELKOWITZ, M. (ed.) "Software Process", Academic Press.

POTTS, C. (1993) "Software-Engineering Research Revisited", *IEEE Software*, Vol. 10, No 5, 19-28.

REID, J. B. (1982) "Observer Training in Naturalistic Research", HARTMANN, D. P. (ed.) "Using Observers to Study Behavior", Jossey-Bass, San Francisco, 37-50.

RESNICK, L. B., SALMON, M., ZEITZ, C. M., WATHEN, S. H. (1993) "Reasoning in Conversation", *Cognition and Instruction*, Vol. 11, Nos 3 & 4, 347-364.

ROBILLARD, P. N. (1999) "The Role of Knowledge in Software Development", *Communications of the ACM*, Vol. 42, No 1, 87-92.

ROBILLARD, P. N., d'ASTOUS, P., DÉTIENNE, F., VISSER, W. (1998) "Measuring Cognitive Activities in Software Engineering", présenté à *International Conference on Software Engineering* (ISCE'98), Kyoto, Japon, 19-25 avril.

ROSCOE, J. T., BYARS, J. A. (1971) "An Investigation of the Restraints with Respect to Sample Size Commonly Imposed on the Use of the Chi-Square Statistic", *Journal of the American Statistical Association*, Vol. 66, No 336, 755-759.

ROSENBLUM, L. A. (1978) "The Creation of A Behavioral Taxonomy", SACKETT, G. P. (ed.) "Observing Behavior. Vol.2: Data Collection and Analysis Methods", University Park Press, Baltimore, 15-24.

RUSCH, F. R., WALKER, H. M., GREENWOOD, C. R. (1975) "Experimenter Calculation Errors: A Potential Factor Affecting Interpretation of Results", *Journal of Applied Behavior Analysis*, Vol. 8, 460.

RUST, R. T., COOIL, B. (1994) "Reliability Measures for Qualitative Data: Theory and Implications", *Journal of Marketing Research*, Vol. 31, 1-14.

SACKETT, G. P. (1979) "The Lag Sequential Analysis of Contingency and Cyclicity in Behavioral Interaction Research", OSOFSKY, J. D. (ed.) "Handbook of Infant Development", Wiley-Interscience Publication, 623-649.

SANDERSON, P. M., FISHER, C. (1994) "Exploratory Sequential Data Analysis: Foundations", *Human-Computer Interaction*, Vol. 9, 251-317.

SCHENKEIN, J. N. (1978) "Explanation of Transcription Notation", SCHENKEIN, J. N. (ed.) "Studies in the Organization of Conversational Interaction", Academic Press, New York, xi-xvi.

SCHERRER, B. (1984) "Biostatistique", Gaëtan Morin éditeur, Boucherville, 850 p.

SCHMIDT, F. L., HUNTER, J. E. (1989) "Interrater Reliability Coefficients Cannot Be Computed When Only One Stimulus Is Rated", *Journal of Applied Psychology*, Vol. 74, No 2, 368-370.

SCOTT, W. A. (1955) "Reliability of Content Analysis: The Case of Nominal Scale Coding", *The Public Opinion Quarterly*, Vol. 19, 321-325.

SEAMAN, C. B. (1996) "Organizational Issues in Software Development: An Empirical Study of Communication", thèse de doctorat, University of Maryland at College Park, 123 p.

SEAMAN, C. B. (1999a) "Qualitative Methods in Software Engineering Research", Séminaire *Measuring Success: Empirical Studies of Software Engineering*, 4-5 mars, NRC, Ottawa.

SEAMAN, C. B. (1999b) Discussions informelles, Séminaire *Measuring Success: Empirical Studies of Software Engineering*, 4-5 mars, NRC, Ottawa.

SEAMAN, C. B., BASILI, V. R. (1997) "An Empirical Study of Communication in Code Inspections", *Proceedings of the ICSE '97*, Boston, 96-106.

SEAMAN, C. B., BASILI, V. R. (1998) "Communication and Organization: An Empirical Study of Discussion in Inspection Meetings", *IEEE Transactions on Software Engineering*, Vol. 24, No 7, 559-572.

SHAPIRO, D. (1994) "The Limits of Ethnography: Combining Social Sciences for CSCW", *CSCW' 94, Proceedings of the Conference on Computer Supported Cooperative Work*, Chapel Hill, North Carolina, 417-429.

SINGER, J., LETHBRIDGE, T. C., VINSON, N., ANQUETIL, N. (1997) "An Examination of Software Engineering Work Practices", *CASCON '97*, Toronto, 209-223.

STEVENS, S. S. (1946) "On the Theory of Scales of Measurement", *Science*, Vol. 103, 677-680.

STREET, R. L. Jr, CAPPELLA, J. N. (1985) "Sequence and Pattern in Communicative Behaviour: A Model and Commentary", STREET, R. L. Jr, CAPPELLA, J. N. (eds.) "Sequence and Pattern in Communicative Behaviour", Edward Arnodl, London, 243-276.

SUPPES, P., ATKINSON, R. C. (1960) "Markov Learning Models for Multiperson Interactions", Stanford University Press, Stanford, 296 p.

TICHY, W. F. (1998) "Should Computer Scientists Experiment More?", *Computer*, Vol. 31, No 5, 32-40.

TICHY, W. F., LUKOWICZ, P., PRECHELT, L., HEINZ, E. A. (1995) "Experimental Evaluation in Computer Science: A Quantitative Study", *Journal of Systems and Software*, Vol. 28, No 1, 9-18.

TJAHJONO, D. (1996) "Exploring the Effectiveness of Formal Technical Review Factors with CSRS, a Collaborative Software Review System", thèse de doctorat, University of Hawaii, 232 p. ftp://ftp.ics.hawaii.edu/pub/tr/ics-tr-95-08.ps.Z

TROCHIM, W. M. K. (1999) "The Research Methods Knowledge Base, 2nd edition". http://trochim.human.cornell.edu/kb/

TRUCHON, M. (1996) "Application des modèles de compréhension des programmes au pseudocode shématique", mémoire de maîtrise, École Polytechnique de Montréal, 22-55.

UEBERSAX, J. S. (1982) "A generalized kappa coefficient", *Educational and Psychological Measurement*, Vol. 42, 181-183.

VON MAYRHAUSER, A., VANS, A. M. (1993) "From Program Comprehension to Tool Requirements for an Industrial Environment", *Proceedings of 2nd Workshop on Program Comprehension*, Capri, Italy, 78-86.

VORTAC, O. U., EDWARDS, M. B., MANNING, C. A. (1994) "Sequences of Actions for Individual and Teams of Air Traffic Controllers", *Human-Computer Interactions*, Vol. 9, 319-343.

VOTTA, L. G. (1993) "Does every Inspection Need a Meeting?", *Proceedings of ACM SIGSOFT '93*, Los Angeles, 107-114.

WALZ, D. B., ELAM, J. J., CURTIS, B. (1993) "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration", *Communications of the ACM*, Vol. 36, No 10, 63-77.

WALZ, D. B., ELAM, J. J., KRASNER, H., CURTIS, B. (1987) "A Methodology for Studying Software Design Teams: An Investigation of Conflict Behaviors in the Requirements Definition Phase", OLSON, G. M., SHEPPARD, S., SOLOWAY, E. (eds.) "Empirical Studies of Programmers: Second Workshop", Ablex Publishing Corp., Norwood, New Jersey, 83-99.

WEICK, K. E. (1968) "Systematic Observational Methods", LINDSEY, G., ARONSON, E. (eds.) "The Handbook of Social Psychology, 2nd edition", Vol. 2, Addison-Welsey, Reading Massachusetts, 357-451.

WELLER, E. F. (1993) "Lessons from Three Years of Inspection Data", IEEE Software, Vol. 10, No 5, 38-45.

WHEELER, D. A., BRYKCZYNSKI, B., MEESON, R. N. Jr. (1996a) "Introduction to Software Inspection", WHEELER, D. A., BRYKCZYNSKI, B., MEESON, R. N. Jr. (eds.) "Software Inspection: An Industry Best Practice", IEEE Computer Society Press, Los Alamitos, California, 293 p.

WHEELER, D. A., BRYKCZYNSKI, B., MEESON, R. N. Jr. (1996b) "Peer Review Processes Similar to Inspection", WHEELER, D. A., BRYKCZYNSKI, B., MEESON, R. N. Jr. (eds.) "Software Inspection: An Industry Best Practice", IEEE Computer Society Press, Los Alamitos, California, 293 p.

WIENBERG, G. M. (1971) "The Psychology of Computer Programming", Van Nostrand Reinhold, New York, 288 p.

WILDMAN, B. G., ERICKSON, M. T., KENT, R. N. (1975) "The Effect of Two Training Procedures on Observer Agreement and Variability of Behavior Ratings", *Child Development*, Vol. 46, 520-524.

WINOGRAD, T. (1987) "A Language/Action Perspective on the Design of Cooperative Work", Rapport no CSLI-87-98, Center for the study of language and information, 32 p.

WINOGRAD, T., FLORES, F. (1986) "Understanding Computers and Cognition", Addison-Wesley, Reading, Massachusetts, 207 p.

WOOD, M., ROPER, M., BROOKS, A., MILLER, J. (1997) "Comparing and Combining Software Defect Detection Techniques: A Replicated Empirical Study", *ACM SIGSOFT Software Engineering Notes*, Vol. 22, No 6, 262-277.

YELTON, A. R. (1979) "Reliability in the Context of the Experiment: A Commentary on Two Articles by Birkimer and Brown", *Journal of Applied Behavior Analysis*, Vol. 12, No 4, 565-569.

ZELKOWITZ, M. V., WALLACE, D. (1997) "Experimental Validation in Software Engineering", *Information and Software Technology*, Vol. 39, 735-743.

ANNEXE A - DURÉE DES INTERVENTIONS

Tableau A.1 Durées moyennes des interventions pour chaque couple de codes

	Version 3					
Version 1	DEV	EVA	JUS	HYP	INF	INT
DEV	21.3	8		19	10.33	
EVA		10.43		9.5	5	
JUS		20	20.6	5	8.5	
HYP	8	4		15.74	19.67	
INF			9.5	3	15.08	
INT						22.59

Tableau A.2 Écart types des interventions pour chaque couple de codes

	Version 3					
Version 1	DEV	EVA	JUS	HYP	INF	INT
DEV	11.8	1.73		5.66	6.1	
EVA		6.73		3.53	2.94	
JUS		5*	13.34	3*	4.43	
HYP	3.24	3*		10.69	16.17	
INF			4.95	2*	12.86	
INT						27.55

Une * signifie que l'écart type pour le couple n'est pas calculable, puisqu'une seule intervention est dans ce couple. La valeur utilisée est une estimation non scientifique, mais proportionnelle à la valeur de la moyenne associée, de l'écart type.

ANNEXE B - EXEMPLE DE SIMULATION DE DONNÉES

L'exemple suivant permettra de bien illustrer le processus de génération de données. Cet exemple est fictif, il n'est pas tiré des données. Comme point de départ, la version 1 a 6 interventions codées DEV, tel que le présente le tableau B.1.

Tableau B.1 Interventions DEV dont on veut simuler des données

ID	Code	Durée
1	DEV	15
2	DEV	4
3	DEV	37
4	DEV	12
5	DEV	20
6	DEV	8

a) Six nombres aléatoires sont générés, suivant une distribution uniforme U(0,1), un pour chaque intervention à coder qui sont codées DEV dans la version 1. Ces six nombres sont:

Tableau B.2 Nombres aléatoires pour la distribution des codes

0.68
0.13
0.57
0.22
0.4
0.49

Les nombres du tableau B.2 sont ensuite comparés aux valeurs du tableau 4.2, au niveau de la ligne pour le code DEV. Ceci montre qu'il y aura 4 codes DEV et 2 codes EVA dans la nouvelle version, puisque 4 des nombres du tableau B.2 sont inférieurs à 0.55, la limite pour DEV, et 2 sont compris entre 0.55 et 0.7, les limites pour EVA.

Il ne reste plus qu'à identifier quel code sera affecté à quelle intervention.

b) La durée moyenne et l'écart type des associations entre les codes DEV et DEV, et entre les codes DEV et EVA sont notés à partir des tableaux A.1 et A.2. Ces valeurs sont 21.3 et 11.8 pour DEV, et 8 et 1.73 pour EVA.

c) Deux séries de nombres aléatoires sont générés, une pour chaque code qui sera associé à une intervention codée DEV dans la version 1. Les nombres aléatoires seront générés de telle manière à suivre une distribution normale N(21.3, 11.8) et N(8, 1.73). Les deux séries sont:

Tableau B.3 Nombres aléatoires pour la distribution des durées

Code	Durée simulée
DEV	9.0878
DEV	17.21533
DEV	25.11824
DEV	28.76453
EVA	7.549516
EVA	10.17305

Il est à noter que ces données ne respectent pas exactement les distributions demandées, mais s'en approche et y tende quand la quantité de nombres générés est grande.

d) Les deux séries de nombres du tableau B.3 sont combinés et triés en une seule série, suivant l'ordre croissant des durées. Ceci donne le tableau B.4.

Tableau B.4 Série combinée triée des nombres aléatoires

Code	Durée simulée
EVA	7.549516
DEV	9.0878
EVA	10.17305
DEV	17.21533
DEV	25.11824
DEV	28.76453

e) Les interventions sont ensuite triées par code et par durée croissante, comme dans le tableau B.5.

Tableau B.5 Interventions triées

ID	Code	Durée
2	DEV	4
6	DEV	8
4	DEV	12
1	DEV	15
5	DEV	20
3	DEV	37

f) La série combinée de nombre aléatoire est associée aux interventions, en respectant l'ordre des durées.

Tableau B.6 Association des nouveaux codes aux interventions

ID	Code	Durée	Nouveau code	Durée simulée
2	DEV	4	EVA	7.549516
6	DEV	8	DEV	9.0878
4	DEV	12	EVA	10.17305
1	DEV	15	DEV	17.21533
5	DEV	20	DEV	25.11824
3	DEV	37	DEV	28.76453

g) Le tableau d'interventions est enfin trié par numéro d'intervention, et les durées simulées sont retirées. Le tableau B.7 présente le résultat final, montrant la version originale du code et la nouvelle version simulée.

Tableau B.7 Nouvelles données simulées

ID	Code	Durée	Nouveau code
1	DEV	15	DEV
2	DEV	4	EVA
3	DEV	37	DEV
4	DEV	12	EVA
5	DEV	20	DEV
6	DEV	8	DEV

ANNEXE C - RÉSULTATS DU LSA

Les 10 tableaux suivants présentent les résultats du LSA sur les différentes simulations de données de la version 1. Les cases ayant un fond gris sont significatives et se traduisent par une flèche sur la figure correspondante.

Tableau C.1 LSA, version 1, semence 169

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,42	**2,63**	0,42	0,27	-1,78	-5,74
EVA	**6,16**	-6,13	**2,81**	-1,84	-1,49	-6,72
JUS	-0,15	1,89	-2,70	-0,26	**2,91**	-4,14
HYP	**2,63**	-1,41	1,70	-5,16	**5,63**	-5,82
INF	-2,10	-0,11	-1,61	**3,49**	-6,24	-7,14
INT	**2,15**	**8,27**	-0,16	**2,15**	**3,16**	-4,96

Tableau C.2 LSA, version 1, semence 65432

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,71	**4,17**	0,81	-0,70	-2,04	-5,98
EVA	**5,51**	-5,83	**2,08**	-1,22	-1,38	-6,62
JUS	0,52	-0,18	-2,57	0,30	**3,05**	-4,11
HYP	0,73	-2,14	1,30	-4,54	**6,35**	-5,56
INF	-0,94	-0,41	-1,16	**4,37**	-6,58	-7,30
INT	**3,59**	**8,48**	0,88	1,03	**2,86**	-4,96

Tableau C.3 LSA, version 1, semence 11

	DEV	EVA	JUS	HYP	INF	INT
DEV	-5,38	**3,03**	1,10	-1,27	-1,81	-6,51
EVA	**4,86**	-5,53	**2,93**	-1,03	-1,32	-6,32
JUS	0,28	0,58	-2,83	1,28	1,81	-4,30
HYP	1,10	-1,66	1,43	-4,52	**6,62**	-5,43
INF	0,20	-0,88	-1,56	**3,85**	-6,20	-7,00
INT	**4,06**	**8,56**	-0,21	0,55	**3,04**	-4,96

Tableau C.4 LSA, version 1, semence 4096

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,73	**3,80**	0,18	-0,64	-2,00	-6,06
EVA	**4,44**	-6,13	**2,36**	-0,84	-1,02	-6,83
JUS	0,64	0,13	-2,48	0,66	1,94	-4,06
HYP	0,94	-1,59	**2,73**	-4,51	**6,00**	-5,45
INF	-1,48	0,28	-1,19	**3,92**	-6,34	-7,16
INT	**4,98**	**7,93**	-0,19	0,29	**3,63**	-4,96

Tableau C.5 LSA, version 1, semence 20807

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,64	**2,70**	1,68	-0,08	-1,52	-5,84
EVA	**5,39**	-5,64	1,76	-1,96	-1,09	-6,48
JUS	0,40	1,65	-3,26	0,11	**2,07**	-4,64
HYP	1,35	-2,83	**2,16**	-4,55	**6,08**	-5,50
INF	-1,01	-0,21	-1,54	**4,30**	-6,31	-7,08
INT	**3,20**	**8,35**	0,49	1,55	**2,68**	-4,96

Tableau C.6 LSA, version 1, semence 19223

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,56	**3,00**	1,47	0,31	-1,98	-5,79
EVA	**4,13**	-5,67	**2,27**	-0,92	-0,85	-6,48
JUS	0,20	0,53	-2,79	-0,13	**2,37**	-4,33
HYP	1,87	-0,77	1,47	-5,07	**5,45**	-5,79
INF	-1,80	0,08	-1,26	**3,85**	-6,38	-7,16
INT	**4,25**	**7,66**	-0,10	1,06	**3,27**	-4,96

Tableau C.7 LSA, version 1, semence 95034

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,39	**2,42**	1,35	-0,37	-1,19	-5,74
EVA	**3,73**	-5,86	**2,95**	-1,04	-0,30	-6,62
JUS	0,62	0,72	-2,63	0,84	1,53	-4,17
HYP	1,80	-1,45	1,70	-5,25	**5,01**	-6,03
INF	-0,84	-0,29	-2,21	**4,22**	-6,08	-7,00
INT	**3,89**	**8,99**	-0,37	1,71	**2,32**	-4,96

Tableau C.8 LSA, version 1, semence 5756

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,27	**3,71**	1,45	-1,31	-1,82	-5,66
EVA	**5,54**	-5,71	**2,15**	-0,81	-1,09	-6,48
JUS	0,18	0,24	-2,70	1,05	1,49	-4,27
HYP	1,47	-2,09	1,45	-5,15	**6,16**	-5,95
INF	-1,71	0,44	-1,44	**3,97**	-6,41	-7,19
INT	**3,37**	**7,58**	0,02	**2,37**	**3,27**	-4,96

Tableau C.9 LSA, version 1, semence 28713

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,79	**5,01**	0,57	-0,89	-1,75	-5,95
EVA	**4,12**	-5,85	**2,15**	-0,72	-1,18	-6,64
JUS	0,09	0,25	-2,51	0,58	**2,49**	-4,06
HYP	1,61	-1,97	1,29	-4,72	**5,90**	-5,63
INF	-1,61	0,33	-1,36	**4,04**	-6,56	-7,27
INT	**4,35**	**7,58**	0,95	0,81	**2,86**	-4,96

Tableau C.10 LSA, version 1, semence 96409

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4,56	**3,93**	0,51	0,09	-2,12	-5,79
EVA	**5,17**	-5,77	**2,01**	-0,65	-1,54	-6,51
JUS	-0,85	0,49	-2,75	-0,05	**2,93**	-4,30
HYP	1,28	-2,47	**2,95**	-5,04	**6,00**	-5,79
INF	-1,29	-0,28	-0,58	**3,93**	-6,50	-7,16
INT	**4,25**	**7,80**	-0,04	1,11	**2,80**	-4,96

Les 10 tableaux suivants présentent les résultats du LSA pour la version 3.

Tableau C.11 LSA, version 3, semence 169

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.49	**5.20**	-1.56	-0.63	-2.41	-5.87
EVA	**4.12**	-6.47	**2.45**	-3.69	-0.23	-7.22
JUS	-0.23	1.10	-1.79	0.51	**2.55**	-3.31
HYP	**2.32**	-1.45	0.76	-3.73	**7.78**	-4.69
INF	-1.10	-0.36	-0.44	**4.70**	-8.76	-8.58
INT	**4.56**	**8.52**	0.04	-1.34	**4.07**	-4.96

Tableau C.12 LSA, version 3, semence 65432

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.58	**4.57**	-0.75	-0.80	-1.46	-5.82
EVA	**3.56**	-6.53	1.78	-2.31	-0.47	-7.16
JUS	0.32	1.49	-2.02	-0.55	**3.54**	-3.45
HYP	1.91	-2.04	0.45	-3.62	**8.54**	-4.59
INF	-2.44	0.93	-0.27	**3.53**	-8.94	-8.68
INT	**5.23**	**7.69**	0.06	-0.97	**3.16**	-4.96

Tableau C.13 LSA, version 3, semence 11

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.11	**4.28**	-0.63	-1.46	-1.86	-5.63
EVA	**3.40**	-6.66	1.63	-1.86	1.14	-7.22
JUS	0.99	0.98	-1.80	-0.77	**3.57**	-3.28
HYP	1.24	-1.12	0.86	-3.91	**6.81**	-4.96
INF	-1.81	0.08	0.07	**3.43**	-8.54	-8.58
INT	**5.41**	**8.40**	-0.48	0.01	**2.99**	-4.96

Tableau C.14 LSA, version 3, semence 4096

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.83	**4.28**	-1.05	-1.02	-0.48	-6.00
EVA	**2.99**	-5.97	1.89	-2.78	-0.30	-6.86
JUS	-0.26	1.75	-2.08	0.71	**3.54**	-3.48
HYP	**3.51**	-1.30	0.18	-3.73	**6.04**	-4.72
INF	-1.92	-0.22	0.43	**3.85**	-8.79	-8.61
INT	**4.35**	**8.65**	-0.48	-0.32	**3.27**	-4.96

Tableau C.15 LSA, version 3, semence 20807

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.31	**5.46**	-0.91	-0.86	-2.67	-5.71
EVA	**2.86**	-6.43	1.70	-1.90	0.33	-7.14
JUS	-0.77	0.45	-1.95	-0.89	**5.50**	-3.42
HYP	**2.45**	-0.88	0.02	-3.65	**6.67**	-4.69
INF	-1.33	0.19	-0.10	**3.43**	-8.92	-8.74
INT	**5.06**	**6.97**	0.20	0.11	**3.80**	-4.96

Tableau C.16 LSA, version 3, semence 19223

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.96	**4.47**	-1.05	-0.98	-0.67	-6.16
EVA	**3.94**	-5.76	1.21	-2.45	-0.44	-6.75
JUS	0.04	0.52	-2.09	-0.13	**4.28**	-3.56
HYP	**2.71**	-1.37	**2.21**	-3.54	**5.54**	-4.62
INF	-1.43	-0.15	-0.49	**3.63**	-8.50	-8.58
INT	**4.35**	**8.00**	0.06	-0.12	**4.14**	-4.96

Tableau C.17 LSA, version 3, semence 95034

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.40	**4.33**	-0.88	-1.65	-1.74	-5.84
EVA	**4.40**	-6.35	1.73	-2.34	0.22	-7.00
JUS	-0.44	0.48	-1.76	0.82	**4.77**	-3.16
HYP	1.31	-1.90	1.93	-3.72	**7.22**	-4.75
INF	-1.79	0.83	-0.95	**3.26**	-9.23	-8.95
INT	**5.40**	**7.77**	-0.54	-0.60	**3.69**	-4.96

Tableau C.18 LSA, version 3, semence 5756

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.77	**4.20**	-0.80	0.38	-0.34	-5.92
EVA	**3.45**	-6.39	1.40	-2.47	-0.04	-7.22
JUS	0.17	**2.25**	-1.73	-0.45	**3.24**	-3.19
HYP	1.74	-1.87	1.36	-3.44	**7.04**	-4.59
INF	-1.41	0.18	-0.90	**2.06**	-8.49	-8.77
INT	**4.35**	**7.80**	0.04	0.04	**4.38**	-4.96

Tableau C.19 LSA, version 3, semence 28713

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.42	**4.57**	-0.52	-1.00	-2.25	-5.82
EVA	**4.25**	-6.26	**2.22**	-1.84	0.47	-6.86
JUS	-0.31	**2.01**	-2.03	-0.24	**4.96**	-3.39
HYP	0.86	-1.66	0.38	-3.40	**7.83**	-4.54
INF	-1.37	-0.32	-1.45	**2.77**	-9.24	-9.14
INT	**4.98**	**6.53**	-0.07	0.38	**4.18**	-4.96

Tableau C.20 LSA, version 3, semence 96409

	DEV	EVA	JUS	HYP	INF	INT
DEV	-4.51	**4.29**	-0.59	-0.91	-2.17	-5.90
EVA	**3.76**	-6.26	1.80	-2.31	0.49	-6.94
JUS	-0.02	0.74	-2.08	1.36	**3.94**	-3.48
HYP	**2.10**	-1.69	1.17	-3.51	**6.39**	-4.62
INF	-1.53	1.59	-1.15	**2.32**	-8.90	-8.77
INT	**4.89**	**6.09**	-0.07	0.60	**4.40**	-4.96