

Titre: La mesure des activités collaboratives retrouvées lors d'une réunion de révision technique du processus de génie logiciel
Title:

Auteurs: Patrick D'Astous, & Pierre N. Robillard
Authors:

Date: 1997

Type: Rapport / Report

Référence: D'Astous, P., & Robillard, P. N. (1997). La mesure des activités collaboratives retrouvées lors d'une réunion de révision technique du processus de génie logiciel. (Rapport technique n° EPM-RT-97-30).
Citation: <https://publications.polymtl.ca/9897/>

Document en libre accès dans PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/9897/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Tous droits réservés / All rights reserved
Terms of Use:

Document publié chez l'éditeur officiel

Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: EPM-RT-97-30
Report number:

URL officiel:
Official URL:

Mention légale:
Legal notice:

21 AOUT 1998

ÉCOLE POLYTECHNIQUE
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ET DE GÉNIE INFORMATIQUE

LA MESURE DES ACTIVITÉS COLLABORATIVES
RETROUVÉES LORS D'UNE RÉUNION DE RÉVISION TECHNIQUE
DU PROCESSUS DE GÉNIE LOGICIEL

gratuit

Patrick d'Astous, ing
Étudiant Ph.D.,

Pierre N. Robillard Ph.D., ing.
Directeur de recherche

Laboratoire de recherche en génie logiciel
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-Ville
Montréal, Qc H3C 3A7
dastous@rgl.polymtl.ca
robillard@rgl.polymtl.ca
Tél.: (514) 340-4711 poste 4238
Fax: (514) 340-3240

Novembre 1997

Tous droits réservés. On ne peut reproduire ni diffuser aucune partie du présent ouvrage, sous quelque forme ou par quelque procédé que ce soit, sans avoir obtenu au préalable l'autorisation écrite des auteurs.

Dépôt légal, 11/97
Bibliothèque nationale du Québec
Bibliothèque nationale du Canada

Les aspects de l'échange d'information dans un processus
de génie logiciel (EPM/RT-97/30)

Patrick d'Astous, Pierre N. Robillard (Génie électrique et génie informatique)

Pour se procurer une copie de ce document, s'adresser au:

Service des Éditions
École Polytechnique de Montréal
Case postale 6079, Succursale Centre-ville
Montréal (Québec) H3C 3A7
Téléphone: (514) 340-4473
Télécopie: (514) 340-3734

Compter 0,10 \$ par page et ajouter 3,00 \$ pour la couverture, les frais de poste et la manutention. Régler en dollars canadiens par chèque ou mandat-poste au nom de l'École Polytechnique de Montréal.

Nous n'honorons que les commandes accompagnées d'un paiement, sauf s'il y a eu entente préalable dans le cas d'établissements d'enseignement, de sociétés ou d'organismes canadiens.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	2
TABLE DES ILLUSTRATIONS	3
INTRODUCTION	4
CHAPITRE I: LA RÉUNION DE RÉVISION TECHNIQUE	6
1.1 INTRODUCTION.....	6
1.2 LE PRODUIT LOGICIEL.....	6
1.3 LE CYCLE DE VIE DU LOGICIEL	6
1.4 LA RÉUNION DE RÉVISION DANS LE PROCESSUS	9
CHAPITRE II : DÉMARCHE D'ANALYSE	11
2.1 INTRODUCTION.....	11
2.2 TRADITIONS ET TECHNIQUES	11
2.3 DESCRIPTION DE LA DÉMARCHE	14
2.4 MÉTHODES D'EXPLORATIONS	17
2.5 L'ASPECT TEMPOREL DU PROJET	19
2.6 ADAPTATIONS ET PROBLÈMES POTENTIELS	21
2.7 SOMMAIRE	23
CHAPITRE III: SCHÉMA DE CODAGE	25
3.1 INTRODUCTION.....	25
3.2 OBJECTIFS DU SCHÉMA DE CODAGE	25
3.3 SCHÉMA GÉNÉRIQUE.....	26
3.4 DESCRIPTION DU SCHÉMA DE CODAGE.....	27
3.5 FORMALISATION DU SCHÉMA DE CODAGE	33
3.6 REPRÉSENTATION DU SCHÉMA.....	33
3.7 SOMMAIRE	37
CHAPITRE IV : IDENTIFICATION ET ANALYSE DES DONNÉES	38
4.1 INTRODUCTION.....	38
4.2 PATRONS DE DIALOGUES.....	38
4.3 ANALYSES STATISTIQUES DES CATÉGORIES	41
CONCLUSION	47
RÉFÉRENCES	48

TABLE DES ILLUSTRATIONS

Figure 1 Cycle de vie en cascade	7
Figure 2 Démarche ESDA.....	16
Figure 3 Aspect temporel des données.....	20
Figure 4 Structure du groupe lecture dans le schéma de codage	28
Figure 5 Structure du groupe demande dans le schéma de codage.....	28
Figure 6 Structure du groupe gestion dans le schéma de codage.....	29
Figure 7 Structure du groupe discussion dans le schéma de codage.....	31
Figure 8 Relations entre les activités et les entités du schéma de codage	32
Figure 9 Représentation BNF du schéma de codage.....	33
Figure 10 Distribution en fréquence relative des activités.....	34
Figure 11 Distribution en fréquence relative des activités détaillées de la catégorie discussion ..	35
Figure 12 Distribution temporelle relative des activités	35
Figure 13 Distribution temporelle relative des activités détaillées dans la catégorie de discussion	36
Figure 14 Distribution temporelle relative des entités	36
Figure 15 Distribution temporelle relative des dialogues	40
Figure 16 Distribution temporelle relative de la résolution de conflit.....	40
Figure 17 Patron identifié avec LSA.....	43

INTRODUCTION

Le génie logiciel est une discipline qui tente de structurer le développement de logiciels. Un processus logiciel est en fait un cadre de travail structuré contenant une suite logique de procédures ainsi que des mécanismes précis.

Le génie logiciel est intéressé par le développement de logiciels nouveaux et uniques dans leur implantation. Ces logiciels sont souvent une solution à un problème concret et complexe, demandant des connaissances du domaine d'application et de plusieurs technologies informatiques différentes. Il est donc probable qu'une équipe de développement soit nécessaire pour développer un tel logiciel.

L'industrie s'entend pour dire que la difficulté de trouver une solution à un problème n'est habituellement pas la conséquence d'une insuffisance de la technologie mais plutôt d'une mauvaise application de celle-ci. La communication à l'intérieur de l'équipe de développement semble donc être essentielle au succès d'un projet (Van Scoy, 1992).

Comme toute discipline de génie, le succès du génie logiciel dépend de la connaissance objective des différents facteurs pouvant influencer le produit final. Ainsi, la mesure est utilisée dans l'évaluation du processus, des caractéristiques intrinsèques du logiciel et de son utilité pour une certaine tâche (Boloix et Robillard, 1995). La mesure de l'aspect humain du processus est cependant tout aussi important.

Les activités cognitives individuelles requises pour le développement de logiciels font l'objet d'études empiriques diverses depuis plusieurs années (Curtis, 1980). Plus particulièrement, les aspects de la compréhension individuelle (Detienne et Solloway, 1990 ; Petre, 1995) et du design (Visser, 1990) ont été étudiés. Il est certain que les ingénieurs logiciels usent d'un certain comportement cognitif individuel qui leur permet d'obtenir et d'utiliser toute l'information nécessaire pour effectuer les tâches qui leur sont assignées. Dans le cadre d'un projet de développement de logiciels, cette information proviendra cependant souvent des autres collaborateurs ou du client.

Un des aspects importants du développement logiciel est qu'il gère le travail collaboratif en instaurant plusieurs mécanismes de gestion de l'information. Les documents du processus ainsi que les types divers de réunions en font partie. Les types de réunions (design, révision et autres) sont identifiés et leurs buts expliqués afin d'assurer la conformité.

Il existe un grand nombre de facteurs qui influencent le travail collaboratif (d'Astous, 1997) dont on doit tenir compte pour comprendre et améliorer le processus. La littérature prescrit pourtant plusieurs façons d'organiser et de tenir une réunion qui sont basées uniquement sur l'expérience des auteurs ou sur des références anecdotales à portée très restreinte. Il faut plutôt pouvoir décrire une réunion en fonction des activités collaboratives inhérentes au lieu des résultats attendus de la réunion.

La réunion de révision technique est un de ces mécanismes clés du processus de développement logiciel. L'objectif de l'étude en cours est d'instancier une méthode de révision technique collaborative qui soit basée sur des observations empiriques d'ingénieurs logiciels. Ces réunions sont très utiles pour le bon fonctionnement d'un projet et la production d'un produit de qualité. Elles occupent en fait une proportion importante du temps consacré à un projet et doivent générer des résultats satisfaisants. Il est donc nécessaire de d'identifier quels éléments sont importants dans ces réunions et lesquels font qu'une réunion est plus productive qu'une autre.

Ce document présente la démarche envisagée pour mesurer empiriquement les activités collaboratives retrouvées lors d'une réunion de révision technique du processus logiciel.

CHAPITRE I: LA RÉUNION DE RÉVISION TECHNIQUE

1.1 INTRODUCTION

La conception d'un logiciel se fait toujours en fonction de critères qui lui demandent souvent une grande flexibilité. Il est donc très important de bien définir à priori tous les éléments susceptibles d'influencer le produit logiciel afin de s'assurer qu'il répondra aux besoins. Le Génie logiciel est une discipline qui amène une certaine rigueur dans le développement de logiciels en définissant un cadre de travail permettant de se diriger vers la fabrication d'un produit final dans une suite d'étapes logiques.

Chacune de ces étapes contient un certain nombre de tâches devant être effectuées. L'extrait de chacune de ces étapes est habituellement vérifié lors d'une réunion de révision technique qui requiert la présence de l'équipe de développement pour une période allant de 30 minutes à 2 heures. C'est cette réunion en particulier qui fait l'objet du présent projet. Le présent chapitre explique le contexte dans lequel on retrouve cette réunion.

1.2 LE PRODUIT LOGICIEL

Un produit logiciel est un ensemble de procédures ou de règles, de données pertinentes à ces procédures et la documentation nécessaires pour répondre à un besoin spécifique.

Les caractéristiques de produit logiciel dont il est question dans ce document sont les suivantes:

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

1.3 LE CYCLE DE VIE DU LOGICIEL

La suite d'étapes logiques a reçu le nom de cycle de vie du logiciel. La vie du logiciel commence donc par l'élaboration d'un certain besoin pouvant être rempli par un produit logiciel

pour se terminer par l'installation de ce logiciel dans son environnement de travail et la maintenance de celui-ci. Les étapes intermédiaires comprennent tout le travail de design, de réalisation et de tests nécessaires à l'implantation du logiciel. La suite logique des ces étapes diffèrent en fonction du genre de cycle de vie utilisé. Le cycle de vie en cascade est l'un des processus de développement les plus utilisés dans l'industrie en ce moment. La Figure 1

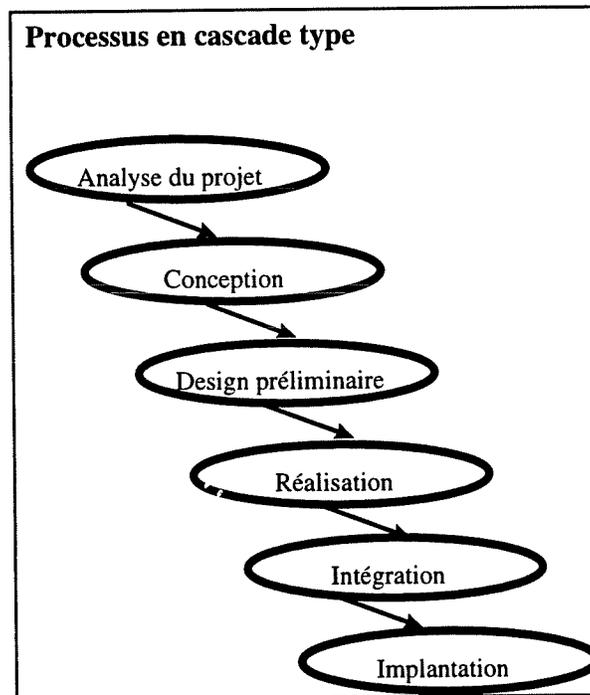


Figure 1 Cycle de vie en cascade

représente un processus en cascade type. Il consiste en une suite d'étapes devant être satisfaites individuellement avant de pouvoir passer à l'étape suivante.

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

La phase de conception est la suivante. C'est durant cette période du projet de développement que les besoins sont décrits en tant que fonctionnalités du produit. Comme son nom l'indique, on identifie les concepts opérationnels qui seront à la base des phases subséquentes. On doit aussi

identifier précisément quelle technologie sera utilisée pour réaliser le logiciel. Cette phase résulte donc en un modèle simplifié de l'environnement contenant ses contraintes de performance et d'implantations.

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

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

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

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

Le cycle de vie décrit précédemment n'est qu'un exemple concret des différentes étapes de développement qui peuvent être utilisées pour développer un logiciel. Chacune de ces phases produit un certain nombre de documents qui permettent de suivre le cheminement cognitif de l'équipe de développement. De plus, ces documents permettent de connaître le rationnel des décisions faites durant le projet et ainsi de comprendre les différentes raisons derrière le choix des différentes implantations des fonctionnalités du produit logiciel.

1.4 LA RÉUNION DE RÉVISION DANS LE PROCESSUS

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

L'une de ces activités de validation est la réunion de révision technique (RRT) qui consiste à vérifier l'état courant du projet de développement et de valider les spécifications des tâches subséquentes. Une RRT requiert la présence de plusieurs réviseurs pour une certaine période et doit se dérouler selon des procédures établies (IEEE, 1993; Bell, 1987).

Il existe présentement plusieurs façons de décrire ce qu'est une réunion de révision technique. Inspection, walkthrough et revue sont des termes souvent utilisés pour identifier une réunion de révision technique. Voici d'ailleurs quelques définitions:

- Une inspection permet de détecter et résoudre des erreurs le plus tôt possible lors du processus de développement logiciel et assure que les erreurs ne sont pas propagées d'une phase de développement à l'autre (Fagan, 1976).
- Une inspection est une rencontre structurée durant laquelle du personnel technique analyse un artefact de façon systématique afin de produire un artefact de meilleur qualité (Johnson, 1996).
- Un walkthrough est considérée comme une revue informelle effectuée par des pairs où il n'est pas nécessaire de se préparer et les suivis de corrections sont inexistantes (Brykczynski et al., 1994).

La réunion de révision technique observée dans ce travail réunit donc des caractéristiques tirées des trois définitions retrouvées auparavant. Il est donc possible de définir les réunions de révision techniques:

Une rencontre formelle durant laquelle un groupe de pairs effectuent l'analyse systématique d'un artefact afin d'y déceler des erreurs, en fonction de barèmes existants, et de les corriger. Le produit d'une telle réunion est un nouvel artefact utilisable lors de la phase subséquente de développement.

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

CHAPITRE II : DÉMARCHE D'ANALYSE

2.1 INTRODUCTION

L'objectif de cette étude est de mieux comprendre les différentes activités collaboratives présentes à l'intérieur d'une réunion de révision technique afin de définir une procédure à suivre lors de ces réunions.

L'optique de cette approche n'est pas de déterminer un modèle théorique et de vérifier son fonctionnement dans une organisation mais plutôt d'analyser le travail d'une équipe d'ingénieurs logiciels lors d'un projet de développement de logiciels industriels. Ce genre de démarche est connu, en anglais, sous le nom "Exploratory Sequential Data Analysis" (Analyse exploratoire et séquentielle des données) (ESDA) (Sanderson et Fisher, 1994). Cette approche favorise l'analyse de systèmes, d'environnements ou de données comportementales dont l'intégrité séquentielle a été préservée. L'analyse de ces données doit correspondre à des questions de recherches, être guidée par une méthodologie et des pratiques et être considérée comme un travail d'exploration.

2.2 TRADITIONS ET TECHNIQUES

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

2.2.1 Tradition comportementale

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

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

Les théories d'échantillonnement guident le choix des données brutes en fonction des sujets et des situations. La verbalisation utilisée dans ce genre d'observation consiste en activités faites naturellement dans le cadre d'interactions de groupe lors de l'accomplissement d'une tâche quelconque. Dans la plupart de cas, le codage de données brutes doit être fait par un humain.

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

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

2.2.2 Tradition cognitive

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

L'approche cognitive dépend grandement de la verbalisation parce qu'elle permet aux analystes d'inférer les activités cognitives. Les transcriptions de verbalisations tendent à se concentrer sur le contenu de l'énoncé plutôt que la raison de son apparition. Elles sont habituellement divisées

en segments (itérations) qui reflètent des éléments du processus cognitif. Dans plusieurs cas, la suite séquentielle est plus importante que le temps réel, il n'est donc pas toujours important de savoir à quel moment absolu une donnée est recueillie.

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

2.2.3 Tradition sociale

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

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

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

2.2.4 La démarche actuelle

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

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

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

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

2.3 DESCRIPTION DE LA DÉMARCHE

2.3.1 Définition de la démarche

Une démarche ESDA demande beaucoup d'explorations, de rétroactions et d'itérations. Le produit d'une étape de la démarche est souvent revu et transformé en fonction de nouvelles informations découvertes. Le processus de transformation des données brutes en produits plus détaillés implique souvent plusieurs itérations qui pourront quelquefois changer les résultats attendus. Un chercheur doit quelquefois adapter ou tout simplement changer complètement de technique en fonction des circonstances.

La démarche préconisée pour ce projet fait bien parti de la famille des études ESDA parce qu'elle contient les caractéristiques suivantes:

1. Les données nécessaires à l'étude ne peuvent être obtenues d'aucune autre façon parce qu'elles sont liées à la séquence et à l'organisation temporelle des activités observées;
2. La technique utilisée pour obtenir les résultats escomptés à partir de la question de recherche en fonction du codage et des analyses est une approche viable et efficace; et
3. Les bénéfices conceptuels apportés par l'ESDA dans ce projet valent le temps additionnel nécessaire pour l'utilisation de cette approche.

2.3.2 Définition et vue globale de la démarche

La Figure 2 présente une vue globale du projet de recherche en fonction des étapes du ESDA. Le premier cercle identifie la question de recherche que nous tentons d'élucider dans le cadre du projet. Les boîtes qui suivent représentent les données brutes qui sont disponibles pour analyses. Ces réunions ont des durées qui varient entre trente minutes et deux heures. Chacune des ces réunions comprend la révision de un ou plusieurs documents conçus dans le processus de développement. Ces documents sont des produits de différentes phases du cycle de vie du logiciel.

Les enregistrements vidéos de ces réunions permettent de produire des transcriptions détaillées des discussions. Ces transcriptions sont le plus fidèles possibles à la réalité, c'est-à-dire qu'elles doivent refléter les différentes activités faites par les participants aux réunions. L'étape suivante du processus consiste à coder les transcriptions dans un langage plus facilement analysable. Pour ce faire, un schéma contenant une liste des activités cognitives retrouvées à l'intérieur d'une réunion est conçue et testée sur des transcriptions afin de s'assurer de sa validité. Une fois cela accompli, les données brutes sont converties dans un langage structuré ayant un nombre fini de possibilité.

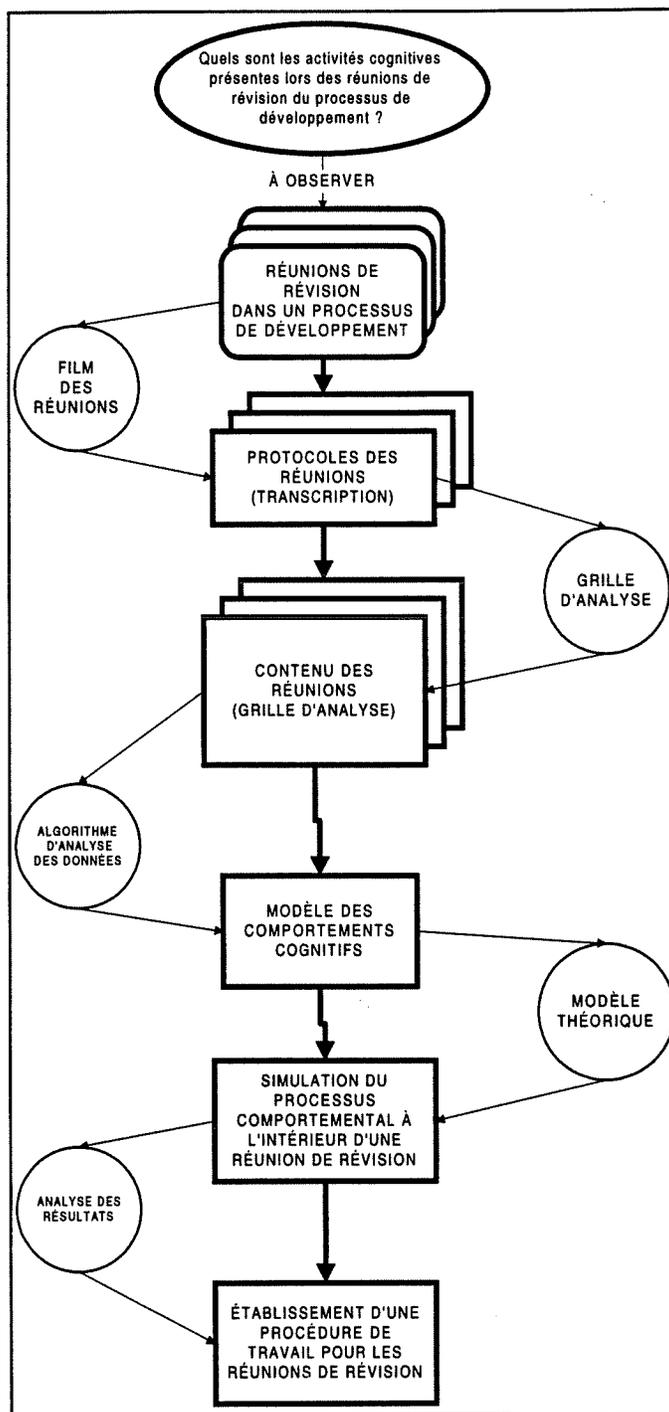


Figure 2 Démarche ESDA

Ces codages de réunion sont alors analysés à l'aide de différentes techniques mathématiques et de solutions de patrons afin de déceler et identifier les comportements importants à l'intérieur

d'une réunion. Un certain nombre d'algorithmes pourront être conçus et utilisés à cette tâche. Le but de cette étape est de concevoir un modèle des comportements cognitifs des participants à une réunion de révision. Un modèle théorique en découlera qui pourra être simulé grâce à un outil de simulation développé au Laboratoire RGL de l'École Polytechnique.

L'analyse de ces résultats devrait permettre d'établir une procédure de travail à suivre par les ingénieurs logiciels lors de leurs réunions de révision.

2.4 MÉTHODES D'EXPLORATIONS

2.4.1 Principes fondamentaux

L'EDA est une méthode simple, visuelle et quantitative d'analyse de données qui permet à un chercheur d'avoir une compréhension plus vaste en "observant les données afin de comprendre ce qu'elles disent" (Tukey, 1977).

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

1. **Ouverture d'esprit et itération:** Il ne faut pas imposer un modèle a priori, plutôt s'en tenir à des idées générales pour ensuite les détailler de plus en plus.
 - a) Le schéma de codage permet de trouver des patrons dans les comportements observés. Ce schéma est conçu initialement à partir de notre expérience du développement de logiciels pour être amélioré en fonction des observations que nous ferons. Les versions subséquentes du schéma illustrent une meilleure compréhension des activités cognitives présentes dans une réunion;
 - b) La simulation d'une réunion de révision technique est un moyen de s'assurer que ce modèle est faisable afin d'établir une procédure de travail utile.
2. **Scepticisme:** Le chercheur doit avoir une approche soupçonneuse dans le cadre de ses observations. Les données numériques doivent être vérifiées et revérifiées pour assurer une modélisation stricte.

- a) Il faut tenter de comprendre et certainement expliquer pourquoi certaines données n'entrent pas dans le modèle théorique; et
 - b) Utiliser des analyses quantitatives qui donnent des résultats justifiés.
3. **Exploration plutôt que confirmation:** Le but de cette démarche n'est pas de tenter de confirmer un modèle théorique par l'analyse de données mais plutôt d'utiliser ces données pour explorer le sujet afin de créer un modèle.

2.4.2 Les cinq opérations de transformation

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

1. **Paquets:** Les paquets correspondent à toutes les épisodes contenues dans une réunion. Il s'agit pour nous de trouver un modèle qui transforme ces morceaux disjoints en un modèle cognitif cohérent.
2. **Commentaires:** Les commentaires sont des annotations utilisées pour ajouter de l'information considérée importante à un moment donné lors des différentes transformations. Lors du codage des transcriptions, par exemple, plusieurs commentaires seront ajoutés pour relier le contenu de la réunion (à partir d'un document) à la transformation.
3. **Codes:** Les codes sont le produit du codage d'une transcription brute en utilisant un schéma contenant les catégories de comportements cognitifs. Le processus de codage consiste à accoler une étiquette à chacun des épisodes.
4. **Interactions:** Le codage des épisodes traduit les données brutes en une série de codes structurés, appartenant au schéma de codage. L'analyse des interactions entre chacun des épisodes donne un aperçu dynamique du comportement cognitif des participants à une réunion. Ces interactions peuvent être mesurées afin d'établir des patrons de comportements cognitifs.

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

2.5 L'ASPECT TEMPOREL DU PROJET

2.5.1 Les données

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

1. **Processus cognitifs inférés:** Ces gestes correspondent à des activités qui ne sont pas directement observées, mais plutôt qui sont implicites à une certaine conversation. Les données brutes contiennent quelques activités de ce genre, telles que les fins de conversations, les coupures de paroles et les moments de pensées.
2. **Vocalisations:** Comportements des participants aux réunions qui sont directement observables à partir des vidéos et qui peuvent être facilement reproduits dans une transcription écrite de la réunion.
3. **Réunions:** Il se peut qu'il soit nécessaire de faire un lien entre la durée de la réunion et le comportement des participants. De plus, il faut vérifier ces comportements à différents moments de la réunion.
4. **Projet:** Il faudra aussi observer l'incidence du moment de la réunion dans les phases du cycle de vie du logiciel ainsi que dans le projet lui-même sur la nature des données recueillies afin de s'assurer que le modèle résultant inclue toutes ces possibilités.

La Figure 3 représente ces quatre importants critères de l'aspect temporel des données dans le projet.

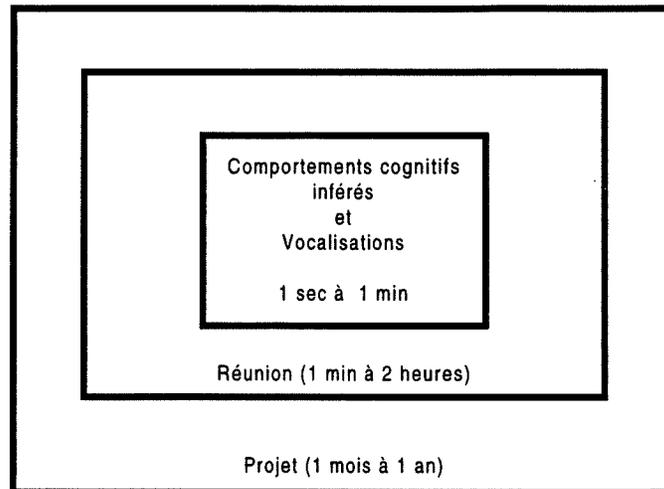


Figure 3 Aspect temporel des données

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

2.5.2 Le ration TA:TS

Le ratio TA:TS correspond au temps nécessaire à l'analyse des données (TA) par rapport au temps des séquences observées (TS). Le temps d'analyse devra être diminué en identifiant bien les facteurs qui déterminent sa longueur et en utilisant des mécanismes qui le permettront. En particulier, il faudra identifier le nombre de réunions nécessaires pour assurer l'obtention de résultats significatifs. Voici quelques facteurs qui auront une influence sur le ratio TA:TS:

1. Préparation des données:
 - a) Transcription des enregistrements vidéos; et
 - b) Codage des transcriptions grâce au schéma de codage;
2. Détermination de la fréquence des comportements à utiliser:
 - a) La façon de coder les comportements cognitifs et les vocalisations aura un impact sur le temps d'analyse. Les épisodes sont donc regroupés selon leurs codes respectifs et l'auteur de la vocalisation;

- b) Afin de s'assurer que le modèle résultant des analyses est conforme à la majorité des réunions du processus de développement, il sera nécessaire d'évaluer des réunions faisant parties d'au moins deux phases différentes. Dans le présent projet, les phases d'ingénierie préliminaire et de réalisation seront observées; et
- c) Il sera important de mesurer l'impact du moment où la réunion a lieu dans le projet. Ainsi, nous pourrons mesurer si les comportements changent à l'intérieur de réunions faisant parties de la même phase du processus de développement mais à différents moments dans le projet.

3. Complexité des concepts

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

Le ratio TA:TS devra être minimisé pour permettre une recherche avec des échéanciers et des résultats acceptables.

2.6 ADAPTATIONS ET PROBLÈMES POTENTIELS

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

2.6.1 Manque d'expertise

Connaissance du domaine d'application. Le travail de recherche en cours demande de bonnes connaissances dans le domaine de l'ergonomie cognitive ainsi que dans l'étude de comportements chez l'individu. Ces connaissances permettent de préciser les détails à observer

pour obtenir des résultats significatifs et utilisables. Il faut donc s'assurer que les analyses effectuées permettent de tester des hypothèses en profondeur et de générer des résultats informatifs. Le défi le plus important qui attend le projet est sans aucun doute le codage des transcriptions de réunions à l'aide d'un schéma de codage. En effet, une expertise en psychologie est nécessaire pour former une grille qui soit significative. Cependant, cette expertise sera gaspillée s'il n'existe pas des codeurs capables de bien comprendre le domaine dans lequel cette grille doit être appliquée. Jusqu'à maintenant, il est apparu que les codeurs devaient absolument être des experts en informatique pour pouvoir effectuer un travail adéquat.

Non correspondance d'une technique d'analyse avec le problème. La démarche ESDA propose un grand nombre de techniques permettant d'analyser les données. Ces techniques doivent être en mesure de bien répondre aux objectifs de la recherche. Après que les questions de recherche sont identifiées, il faut donc faire une recherche des différentes techniques et décider si elles sont appropriées dans le cadre du projet actuel. Il sera donc nécessaire de rechercher les techniques les plus susceptibles de nous aider dans ce projet et d'évaluer la faisabilité de les employer. Ce travail demande un minimum de connaissance de l'ESDA et de son application. Il est intéressant de comparer le présent travail avec celui fait au préalable par Olson et als. (1992) afin de déterminer si les techniques qu'ils ont employées pourraient être utilisées.

2.6.2 Gestion du temps: Ratio TA:TS

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

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

comportements possibles des participants et en minimisant l'interprétation différente des données par plusieurs codeurs. Une formation adéquate de chacun des codeurs devrait réduire encore plus la chance d'erreur.

2.6.3 Problèmes reliés aux données

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

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

2.7 SOMMAIRE

Ce chapitre a permis de bien comprendre la démarche de recherche utilisée en fonction d'une approche typique ESDA. Il apparaît donc que la présente recherche est une adaptation des traditions comportementales et cognitives de l'ESDA. Il est donc maintenant possible d'établir avec certitude dans quelle direction les recherches de techniques d'analyses doit être conduite. La décision d'utiliser ou non des techniques disponibles devra se faire en tenant compte des caractéristiques du projet, du temps disponible à son application et surtout, des avantages et désavantages qu'elles apporteront au projet.

L'adaptation de ces techniques au projet actuel constitue une autre difficulté inhérente au projet. Il s'agit de trouver les techniques d'analyses qui nous permettront de mener à bien ce projet en tenant compte de contraintes d'échéancier et de qualité. Une expertise dans l'ESDA nous est donc nécessaire. Les spécialistes de l'INRIA apporteront certainement une aide importante dans

le choix de ces techniques, mais leur application dans l'analyse des données disponibles ne pourra se faire que par des spécialistes en génie logiciel.

Le développement d'un schéma de codage adéquat est la première et définitivement la plus importante étape du projet. Comme il a été souvent discuté auparavant, il constitue la pierre angulaire de tout le travail d'analyse ultérieur. Le codage des données brutes dans un vocabulaire précis permettra d'identifier des patrons types et d'obtenir des mesures statistiques fiables. Le succès de la démarche d'analyse sera dépendant en grande partie de la précision de ce schéma.

CHAPITRE III: SCHÉMA DE CODAGE

3.1 INTRODUCTION

Le schéma de codage présenté dans ce chapitre est le résultat d'une démarche évolutive et itérative qui a permis de faire ressortir les détails importants retrouvés lors d'une réunion de révision technique (d'Astous et al., 1997). En premier lieu, un schéma de codage générique a été identifié qui peut être adapté à plusieurs types de réunion dans le processus de génie logiciel. Dans un deuxième temps, ce schéma générique est complété afin de s'appliquer plus spécifiquement aux réunions de révision technique.

Le présent chapitre présente la dernière version du schéma de codage et donc celle utilisée pour recueillir les données présentées dans la suite de ce document.

3.2 OBJECTIFS DU SCHÉMA DE CODAGE

Bien que ce travail soit exploratoire parce que nous ne connaissons pas exactement ce que nous recherchons, nous savons que le travail consiste à identifier, analyser et comprendre les comportements cognitifs importants qui rendent une réunion de révision utile au processus de développement de logiciels. Le schéma de codage est l'outil de prise de données qui servira de base pour le reste du travail, il doit donc être en mesure de bien représenter les protocoles. Le schéma doit répondre aux critères suivants:

3.2.1 Fiabilité

La fiabilité de toute mesure est souvent représentée par la reproduction des résultats obtenus. Les catégories formant le schéma de codage doivent donc être suffisamment précises et orthogonales pour permettre d'éliminer le plus possible les interprétations personnelles. La description des catégories doit enlever toute ambiguïté et réduire l'introduction de "bruits" dans le codage. La fiabilité ne pourra être mesurée qu'en soumettant un protocole à plusieurs codeurs différents utilisant le même schéma de codage afin de mesurer le degré d'homogénéité des données obtenues.

3.2.2 Formalisme

Il doit être possible de représenter le schéma de codage dans un formalisme afin de s'assurer de sa validité et de sa complétude. Puisque les catégories identifiées dans le schéma de codage sont le fruit d'une certaine démarche scientifique et théorique, leur formalisation devrait assurer l'existence d'une rigueur intellectuelle dans leur choix et leur utilisation. La forme normale de Backus (BNF) est employée dans le cadre de ce projet pour formaliser le schéma de codage.

3.2.3 Représentation

Le schéma de codage est l'outil qui permet de transformer les données brutes en un ensemble de données qui pourront ultérieurement être analysées. Il est donc important que les catégories représentent le plus précisément possible les comportements retrouvés dans une réunion de révision. Il faut aussi que la représentation des catégories soit relativement proportionnelle, c'est-à-dire qu'il n'y ait pas de catégories qui caractérisent la majeure partie du protocole contre d'autres qui ne caractérisent qu'une infime partie.

De plus, afin de valider les catégories utilisées pour coder les comportements, il est nécessaire d'avoir un nombre suffisant d'échantillons. Une règle retrouvée dans la littérature (Fagen et Young, 1978) stipule que le nombre minimum d'échantillon devrait être de dix fois le carré du nombre de catégories dans le schéma de codage. Il est donc important de s'assurer que nous avons un nombre de catégories qu'il soit possible de valider avec les échantillons disponibles.

3.2.4 Analyse

Les catégories doivent permettre une analyse simple et relativement rapide afin de réduire le TA. Il est certain qu'un plus grand nombre de catégories permettra de coder plus en détails les comportements cognitifs, mais cela comportera des risques d'erreurs plus grands. Il s'agit de trouver l'équilibre entre la simplicité d'analyse et le détail de la couverture de nos catégories.

3.3 SCHÉMA GÉNÉRIQUE

Le schéma de codage générique permet de coder les différents épisodes en utilisant une structure syntactique très précise composée de catégories. Chacune des interventions retrouvées dans la réunion est décomposée en épisodes selon les règles suivantes:

1. un nouvel épisode commence lorsque l'activité incluse dans une intervention ne peut plus être décrite par la même catégorie ; et
2. l'auteur de l'intervention change.

Le schéma de codage générique est composé de trois niveaux afin de faciliter l'identification des épisodes et créer une base cohérente pour les analyses subséquentes:

1. le niveau 1 identifie le type d'activité effectué par un individu;
2. le niveau 2 définit l'entité sur laquelle l'activité est effectuée; et
3. le niveau 3, optionnel, est un critère qui complète l'entité.

Par exemple, les participants à une réunion peuvent discuter (activité) la version x d'un document (entité) de design en se référant au format utilisé pour décrire le design (critère de forme).

Bien que la structure de ce schéma de codage soit générale, il est malgré tout nécessaire de l'adapter au contexte d'une réunion de révision technique en spécifiant les types d'activités, la nature des entités ainsi que les critères susceptibles d'être identifiés lors de ces réunions.

3.4 DESCRIPTION DU SCHÉMA DE CODAGE

Le schéma de codage adapté aux réunions de révision technique est formé de quatre groupes d'activités distincts: lecture, demande, gestion et discussion. Il existe aussi un nombre d'entités qui permettent de bien préciser l'activité: messages, artefacts et tâches. Le Tableau 1 illustre les activités, entités et critères utilisés pour adapter le schéma générique aux réunions de révision technique.

ACTIVITÉ	ENTITÉ	CRITÈRE
Lecture	Artefact	
Demande	Objet (Artefact et tâche)	Forme et Contenu
Gestion	Tâche	
Discussion	Objet (Artefact et tâche)	Forme et Contenu

Tableau 1 Adaptation du schéma générique pour les RRT

3.4.1 Activités de lecture

La progression de la réunion est gérée par la lecture de chacune des sections d'un document (artefact) qui sont utilisées comme références. Cette coupure est discernable facilement. L'activité de lecture identifie tous les épisodes où un participant à la réunion présentera un artefact afin de le soumettre à l'évaluation des autres participants. La Figure 4 illustre la composition du groupe lecture.

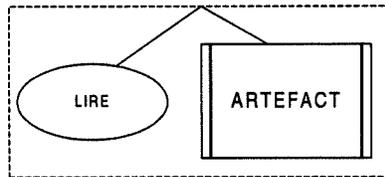


Figure 4 Structure du groupe lecture dans le schéma de codage

3.4.2 Activités de demande

La demande (DEM) est une activité où un participant sollicite un comportement précis de la part d'un autre participant à la réunion, ce comportement étant spécifié par l'objet de cette demande. La demande sera donc spécifiée par une activité afin de la préciser.

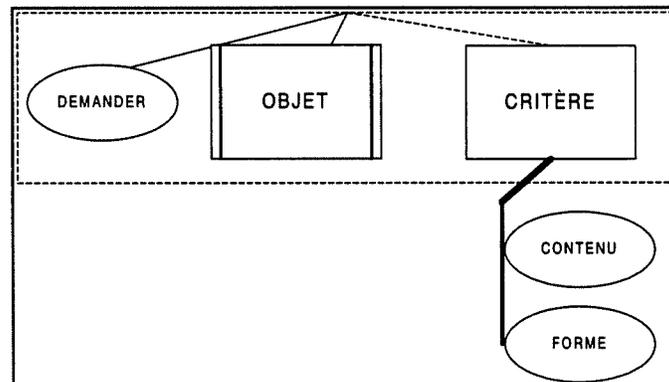


Figure 5 Structure du groupe demande dans le schéma de codage

3.4.3 Activités de gestion

Les activités de gestion (GEST) sont les actions qui déterminent le comportement que doivent prendre les participants à une réunion. Il existe deux types d'activités de gestion: la coordination et la planification. Les activités de gestion se réfèrent aux besoins de s'assurer que la réunion suit son cours ou qu'une certaine activité peut avoir de l'impact sur le projet.

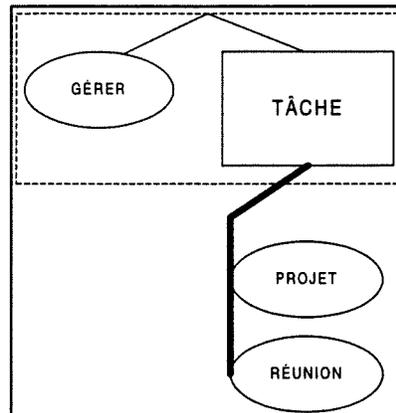


Figure 6 Structure du groupe gestion dans le schéma de codage

3.4.4 Activités de discussion

La discussion est le groupe d'activités principal d'une réunion de révision technique. Ce groupe est divisé en dix activités qui permettent d'identifier très précisément un épisode particulier:

1. **Justifier:** Une justification est l'action d'argumenter ou de prouver le bien-fondé du POURQUOI on a fait un certain choix de solution ou de critère. Il est souvent nécessaire de répondre à des évaluations par une justification de l'approche prise. On peut justifier plusieurs objets.
2. **Informier:** Action de porter des renseignements à la connaissance d'une ou de plusieurs personnes au sujet de la nature d'une solution ou d'un critère.
3. **Évaluer:** Action de porter un jugement sur la valeur d'un objet. On émet une opinion sur un fait quelconque. Cette évaluation peut être négative, positive ou neutre.
4. **Accepter:** Action de considérer un objet comme valide.
5. **Rejeter:** Refuser ou écarter un objet comme non valide.
6. **Développer:** Action d'exposer en détail une idée nouvelle au sujet d'une solution ou d'un critère. Il s'agit donc d'une activité créatrice.

7. **Faire une hypothèse** : Présenter la représentation que l'on se fait d'une solution ou d'un critère. Cette représentation a le statut cognitif d'hypothèse ce qui s'exprime verbalement par des expressions comme "je crois que", "je pense que...", "...peut-être".

3.4.5 Entités

Il existe deux type d'objets dans une conversation. Premièrement, il y a le contenu des documents qui sont utilisés à l'intérieur de la réunion. Il s'agit ici d'idées élaborées préalablement qui servent souvent de base à la discussion ou à amener des compléments d'information aux participants. Ces objets sont appelés "artefact". Dans le cas d'une réunion de révision, il peut s'agir de solutions et de critères.

Dans un deuxième temps, les activités cognitives identifiées à l'intérieur d'une réunion produisent elles-mêmes des objets pouvant être utilisés **ultérieurement** par d'autres activités cognitives. Une activité se fait donc toujours sur un objet déjà existant. Ainsi les activités cognitives produisent les objets suivants (appelés "messages"):

• Évaluer	→	Évaluation
• Justifier	→	Justification
• Accepter	→	Acceptation
• Rejeter	→	Rejet
• Faire hypothèse	→	Hypothèse
• Développer	→	Développement
• Informer	→	Information

La Figure 7 illustre la composition du groupe *discussion* dans le schéma de codage.

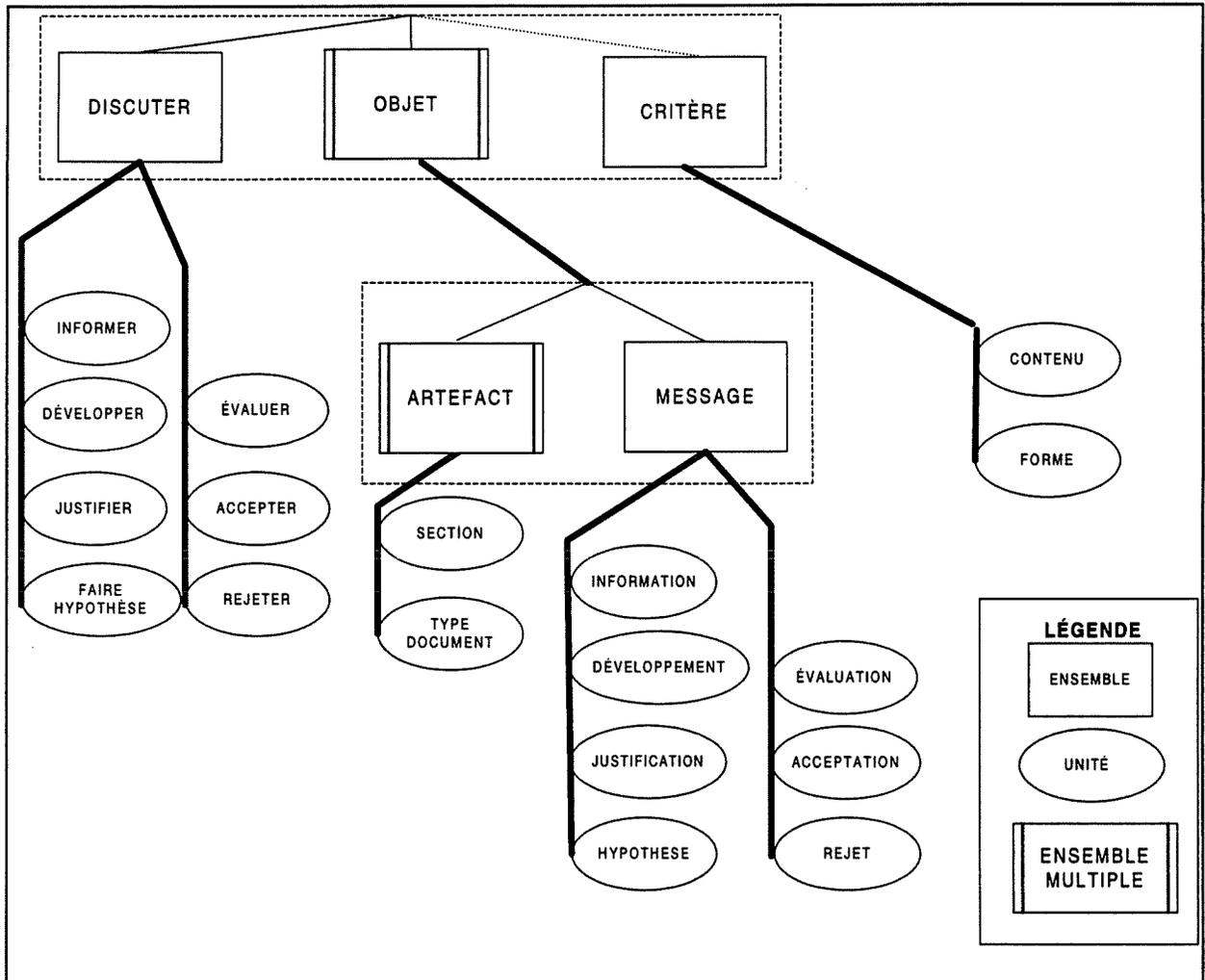


Figure 7 Structure du groupe discussion dans le schéma de codage

Voici quelques exemples:

- x ÉVAL/SOLn/CRIT.C Évaluer la solution n par rapport à un critère de contenu
 x+1 JUSTIF/ÉVALx/CRIT.F Justifier l'évaluation x à l'aide d'un critère de forme

Chacun des objets (artefact ou message) est suivi d'une expression alphanumérique qui permet d'identifier clairement celui-ci dans les épisodes suivants. Ainsi, l'artefact SOLn est évalué dans l'épisode x, donnant comme résultat l'ÉVALx qui est elle-même justifié lors de l'épisode x+1.

Les bénéfices d'une telle structure sont de deux natures: l'activité est représentée par un verbe qui est précis et qui permet de comprendre directement le genre d'activités cognitives auquel l'on

fait face. L'utilisation d'un objet produit par la discussion permet de préciser encore plus une activité cognitive subséquente.

Cette structure est intéressante parce qu'il est conceptuellement possible d'avoir n'importe quelle permutation "activité/objet" du schéma. Toutefois, l'expérience et le type de réunion évalué aura une incidence sur les permutations effectivement présentes.

Une demande se fait elle aussi sur un objet (artefact ou message) tandis qu'une lecture s'effectue sur un artefact seulement. Enfin, la gestion se fait sur des tâches (réunion ou projet). La Figure 8 représente les différentes relations entre les activités et les entités du schéma de codage.

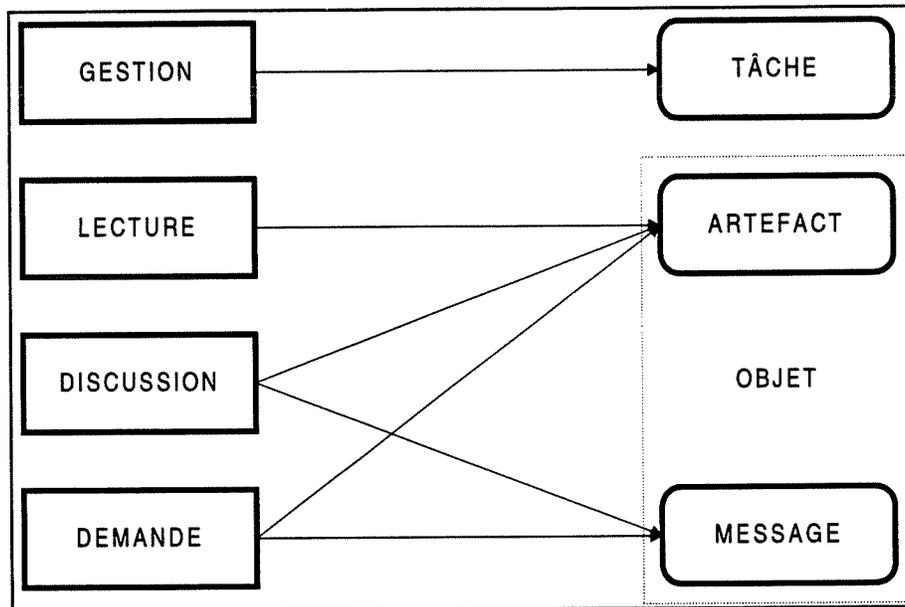


Figure 8 Relations entre les activités et les entités du schéma de codage

3.4.6 Critères de contenu et de forme

Une liste des critères de forme et de contenu à utiliser lors des codages est utilisée et tenue à jours. Les critères de contenu proviennent de la norme ISO 9126 tandis que les critères de forme sont issus du cahier des pratiques utilisés par les membres de l'équipe de développement. Voici la liste courante:

Critères de forme

- a. Nomenclature
- b. Forme des algorithmes
- c. Documentation

Critères de contenu

- (ISO 9126)
- a. Fonctionnalité
 - b. Réutilisation

- | | |
|------------------------------|------------------------------|
| d. Fonctions | c. Portabilité |
| e. Fichiers | d. Fiabilité |
| f. Types de données | e. Maintenabilité |
| g. Schémacode (Éditeur) | f. Efficacité |
| h. Déclaration des variables | g. Simplicité d'implantation |
| i. Variables globales | |
| j. Structure du document | |
| k. Sémantique | |
| l. Niveau de description | |

3.5 FORMALISATION DU SCHEMA DE CODAGE

La notation BNF (Naur, 1960) a été utilisée pour formaliser la schéma de codage et est présentée dans la Figure 9. Cette représentation permet de valider la sémantique du schéma et d'illustrer sa flexibilité.

Définition des symboles:

/	séparateur	< >	catégorie non terminale
::=	signifie "est défini comme"	{ } ⁺	au moins un item doit exister
	signifie "ou exclusif"	{ } ⁰	l'item est optionnel

<Codage> ::=	{ {gérer / <tâche> } ⁰ {lire / <artefact>} ⁰ {demander / <objet> {/ <critère>} ⁰ {<discuter> / <objet> {/ <critère>} ⁰ } ⁺
<discuter> ::=	accepter développer évaluer faire hypothèse informer justifier rejeter
<tâche> ::=	projet réunion
<objet> ::=	<artefact> <message>
<artefact> ::=	<document type> <section>
<message> ::=	acceptation développement évaluation hypothèse information justification rejet
<critère> ::=	contenu forme

Figure 9 Représentation BNF du schéma de codage

Un codage est composé d'au moins un des quatre groupes disponibles dans le schéma de codage. Chacun des épisodes est donc décrit par une activité, un symbole de séparation (/), l'entité approprié et un critère optionnel, dans le cas de l'activité de discussion.

3.6 REPRÉSENTATION DU SCHEMA

Ce chapitre présente les données recueillies dans le but de mesurer la représentation du schéma de codage pour l'analyse de réunions de révision technique. Les données ont été obtenues à partir de l'analyse d'une RRT effectuée par quatre individus. Cette réunion a duré vingt-cinq

minutes et contient 256 épisodes codés. Les données résultantes peuvent être analysées selon trois points de vues : une distribution en fréquences et une distribution temporelle des catégories, ainsi qu'un regroupement de catégories afin de former des patrons de dialogues.

Bien que ces données aient été prises à partir de l'observation d'une seule réunion de révision technique et qu'elles ne peuvent servir de base pour une argumentation solide, elles démontrent très bien l'étendue de l'utilisation du schéma de codage.

3.6.1 Distribution en fréquence des catégories

La distribution en fréquence démontre l'importance relative des épisodes codés et permet de démontrer la représentation du schéma de codage pour une réunion de révision technique. Par exemple, une catégorie fréquemment utilisée signifie que la plupart des épisodes sont codés dans celle-ci. Le codage offrirait donc une faible sensibilité à ce type d'activité. La Figure 10 illustre l'importance relative des fréquences des groupes d'activités de gestion (GEST), lecture (LECT), demande (DEM) et discussion (DISC).

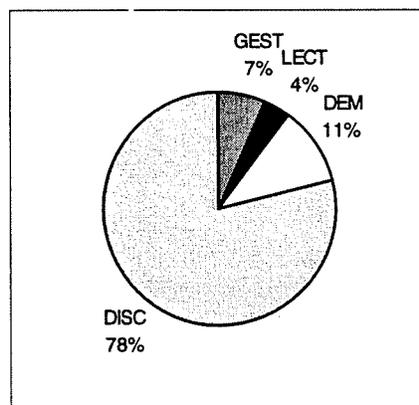


Figure 10 Distribution en fréquence relative des activités

Étant donné la domination de l'activité de discussion (78%), il est nécessaire de la subdiviser en huit activités plus détaillées (les trois activités de faire une hypothèse ayant été généralisées à une dans les calculs de fréquence). La Figure 11 illustre la distribution en fréquence de ces activités détaillées. La plupart des activités sont distribuées proportionnellement. L'observation de ces résultats permet de valider la sensibilité et la représentation du schéma de codage pour les réunions de révision technique.

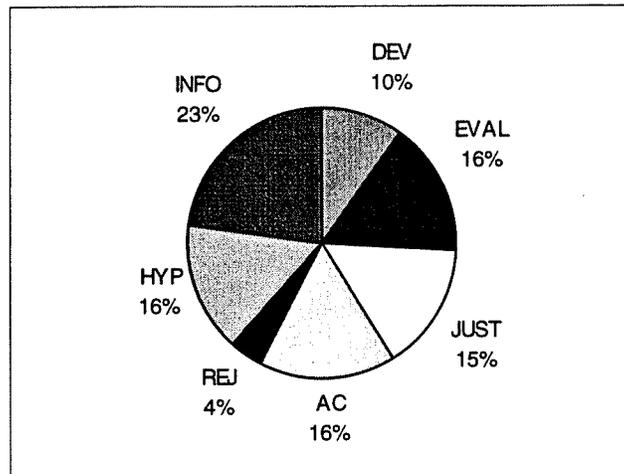


Figure 11 Distribution en fréquence relative des activités détaillées de la catégorie discussion

3.6.2 Distribution temporelle des activités

L'analyse de la distribution temporelle des catégories illustrée à la Figure 12 démontre combien de temps est utilisée par chacune des catégories. L'activité de lecture exemplifie la relation entre la fréquence et le temps. Malgré le fait que la fréquence de l'activité de lecture soit non significative (<5%), le temps passé dans cette activité l'est (>15%).

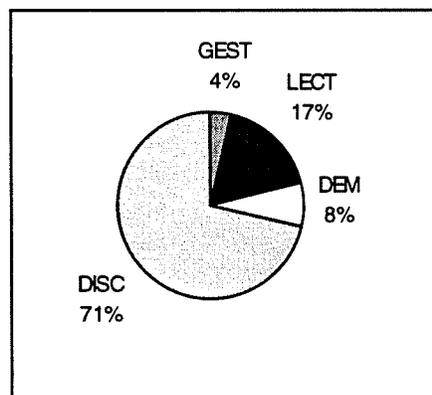


Figure 12 Distribution temporelle relative des activités

La Figure 13 démontre que les activités explication, développement et justification requièrent une portion importante de la réunion de révision. L'activité d'acceptation, qui est en fait le but de la réunion, prend seulement 5% du temps de discussion bien qu'elle soit présente dans 16% des épisodes. Cela signifie que les épisodes d'acceptation ne demandent que très peu de temps.

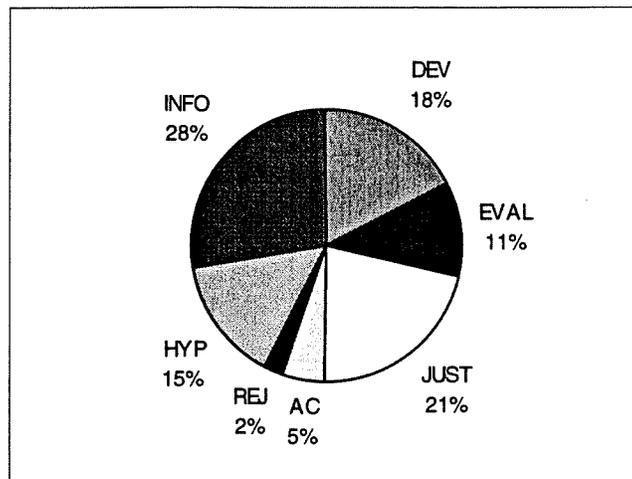


Figure 13 Distribution temporelle relative des activités détaillées dans la catégorie de discussion

3.6.3 Distribution temporelle et en fréquence des entités

Grâce à la structure du schéma de codage, il est aussi possible d'effectuer des analyses de distribution temporelle ou en fréquence des entités. Ainsi, la Figure 14 illustre le temps relatif passé à discuter la solution initiale (SOL. INI.) présentée dans le document révisé, les solutions alternatives (SOL. ALT.) proposées, les critères d'évaluation (CRIT) et les autres (AUT).

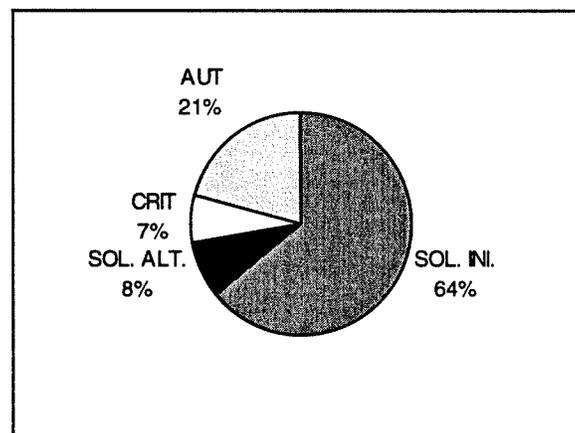


Figure 14 Distribution temporelle relative des entités

La Figure 14 démontre clairement que malgré qu'une réunion de révision type doit se pencher uniquement sur la révision d'un document, 35% du temps de réunion est utilisé pour discuter de d'autres sujets.

3.7 SOMMAIRE

Le schéma de codage présenté dans ce chapitre permet de coder beaucoup d'information directement en prenant connaissance du verbe spécifiant l'activité cognitive tout en permettant d'utiliser une plus grande liste d'objets pour préciser encore plus ces activités.

Cette structure répond à l'objectif de formalisme évoqué auparavant et les données recueillies initialement tendent à démontrer que le schéma représente bien les activités retrouvées dans une réunion de révision technique. Les objectifs de fiabilité et d'analyse seront vérifiés par l'utilisation de ce nouveau schéma de codage.

CHAPITRE IV : IDENTIFICATION ET ANALYSE DES DONNÉES

4.1 INTRODUCTION

Les données recueillies dans le but de vérifier la représentation du schéma de codage ont permis de vérifier sa validité. L'information disponible à partir de ce genre de travail est utile et informatif, mais d'une application limitée pour le génie logiciel. La réunion de révision technique est constituée d'un ensemble d'activités cognitives, reliées dans un réseau dynamique complexe. L'intérêt principal de ce genre de recherche se situe surtout dans la possibilité d'observer et d'identifier des régularités (ou patrons) dans les données recueillies. Ce genre d'analyse implique l'utilisation d'une séquence de données comportementales.

4.2 PATRONS DE DIALOGUES

L'identification de patrons dans les dialogues requiert un grand nombre de données, une série d'hypothèses à vérifier et une méthode d'analyse robuste. Un nombre $5 R^2$ de données (R étant le nombre total de catégories) est nécessaire (Fagen et Young, 1978) pour obtenir des résultats significatifs. Le cas de l'analyse des activités nécessite donc un minimum de 605 données (11 catégories – lire, demander, gérer, développer, justifier, informer, faire hypothèse, expliquer, accepter, rejeter, évaluer). Chacune des réunions disponibles pour fin d'analyses contient en moyenne 400 épisodes, ce qui rencontre parfaitement le critère.

Les différentes méthodes d'analyse seront discutées plus tard. Pour l'instant, concentrons la discussion sur le besoin de formuler des hypothèses valables afin de faciliter l'identification de dialogues. Ces hypothèses doivent être fondées sur une connaissance des comportements attendus ainsi que sur les fondements théoriques de l'ergonomie cognitive. La première étape de la démarche d'analyse consiste donc à identifier des patrons de comportements (dialogues) de façon heuristique, en observant les données qualitativement. Ce genre de travail permet de sélectionner les bornes et les critères qui seront utilisés ultérieurement lors des analyses statistiques.

Cinq dialogues ont été observés à partir de la même réunion que précédemment :

1. **Synchronisation cognitive (Falzon, 1996):** Permet aux participants de s'assurer qu'ils partagent une représentation commune de l'état d'un artefact, d'un message ou d'un critère (contenu ou forme).
2. **Révision:** Les participants jugent la valeur ou donnent leur opinion d'une alternative ou d'un critère utilisé pour évaluer une solution. La révision peut être négative ou positive. La révision peut être implicite.
3. **Résolution de conflit:** Arguments entre deux ou plusieurs participants quant à la divergence d'opinion sur un critère ou l'évaluation d'une solution. Il peut aussi y avoir résolution de conflit lorsqu'il existe plusieurs représentations cognitives d'une même solution ou critère.
4. **Élaboration d'alternatives:** Conception et analyse de solution qui ne font pas partie du document original (artefact).
5. **Gestion:** Coordination et planification de différentes tâches.

Chacun de ces dialogues est composé d'une ou plusieurs activités cognitives faisant parties du schéma de codage. La particularité d'une séquence ainsi que sa fréquence permet de définir et de différencier un dialogue d'un autre.

La Figure 15 montre distribution relative temporelle des trois dialogues principaux. Il est intéressant de noter que la moitié du temps de la réunion est utilisé pour la synchronisation cognitive (SYNC) tandis que l'autre moitié est séparé également entre la révision (REV) et l'élaboration de solutions alternatives (ALT).

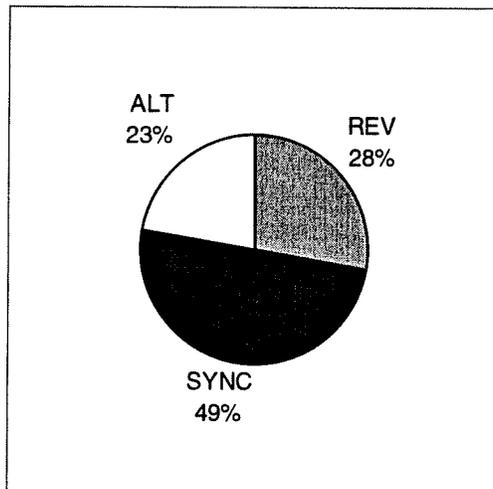


Figure 15 Distribution temporelle relative des dialogues

La figure précédente n'inclue pas la résolution de conflit parce que celle-ci semble constitué un sous dialogue présent à l'intérieur des autres dialogues. La Figure 16 démontre la résolution de conflit occupe au moins 10% du temps dans les autres dialogues et jusqu'à 20% dans le cas de la synchronisation cognitive.

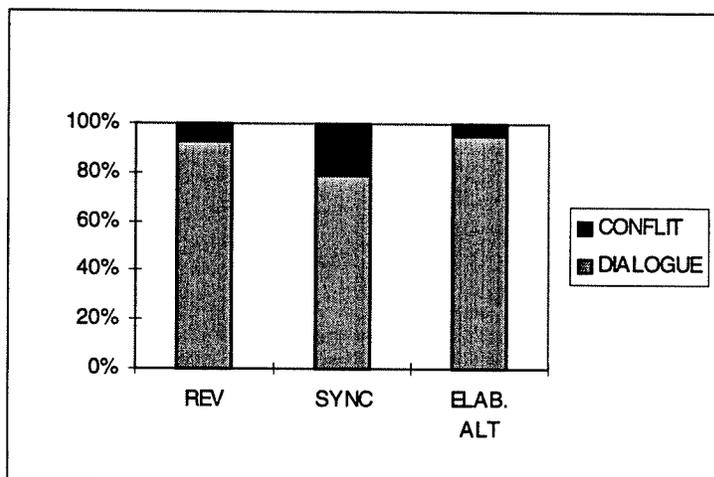


Figure 16 Distribution temporelle relative de la résolution de conflit

Les cinq dialogues identifiés auparavant sont donc retrouvés dans une réunion comprenant environ 250 épisodes. Bien qu'ils ne soient donc pas significatifs selon le critère décrit dans ce chapitre, ils permettent d'avoir une bonne idée des résultats attendus. Les hypothèses

nécessaires pour l'utilisation de méthodes statistiques formelles peuvent donc être envisagés à partir des dialogues.

4.3 ANALYSES STATISTIQUES DES CATÉGORIES

Plusieurs méthodes d'analyses statistiques peuvent être utilisées afin de déterminer les types de dialogues. Les méthodes statistiques permettent de savoir qu'elles activités forment une séquence (un dialogue) selon un certain niveau de confiance. La modélisation permet d'estimer les paramètres qui caractérisent une certaine situation à partir des données recueillies.

Deux méthodes sont considérées pour cette partie du travail : la modélisation log-linéaire et le *lag sequential analysis* (LSA). Chacune de ces méthodes provient d'une base formelle et peut s'appliquer au présent travail.

4.3.1 Lag Sequential Analysis (LSA)

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

Lorsque l'activité antécédente est étudiée en relation avec l'activité conséquente décalée de k , toutes les activités intermédiaires sont ignorées. Si l'antécédent varie, un profil sera alors calculé pour chaque conséquent. Ces profils sont utilisés pour générer des hypothèses quant aux séquences probables dans les données.

Le LSA est donc une mesure de la dépendance d'une activité sur une autre. On utilise la cote- z comme mesure de dépendance. Lorsqu'on veut évaluer les probabilités en utilisant une distribution normale, il est nécessaire d'évaluer les régions sous la courbe (normale). Il existe cependant un nombre infini de distributions normales correspondants aux μ (moyenne) et à σ^2 (écart-type). La cote- z mesure le nombre d'écart-types de distance d'une observation par rapport à la moyenne (la chance). La variable "normale" z permet d'unifier ces deux valeurs comme suit:

$$z = \frac{X - \mu_x}{\sigma_x}$$

Bien que la puissance du LSA et de la cote-z pour ce genre de travaille ne fasse aucun doute, il existe plusieurs différents quant à la façon la plus appropriée de calculer cette statistique. Allison et Liker (1982) ont proposé une telle statistique qui est encore aujourd'hui la plus utilisée. Elle se sert de méthodes développées pour l'analyse des tables de contingences.

$$z = \frac{P_{B|A} - P_B}{\sqrt{\frac{P_B(1-P_B)(1-P_A)}{(n-k)P_A}}}$$

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

	DÉVELOPPER	ÉVALUER	DEMANDER	EXPLIQUER	JUSTIFIER	ACCEPTER	REJETER	HYP (INFO)	HYP (JUST)	HYP (EXPL)	INFORMER	LIRE	GÉRER
ACCEPTER	0.29	-1.76	2.62	-0.22	2.28	-1.25	-1.10	-0.07	-0.02	-0.02	-0.40	-0.16	-0.89
	-1.78	2.18	-0.96	1.01	-0.57	-1.80	2.10	1.05	-0.02	-0.02	-1.09	1.86	-0.89
	-0.40	1.05	2.01	-0.83	1.14	-0.69	-1.10	-0.63	-0.02	-0.02	-0.40	-1.17	-0.14

Tableau 2 LSA des activités avec ACCEPTER comme antécédents

Le Tableau 2 suivant illustre les résultats obtenus en utilisant le LSA sur les données de la réunion initiale. Attention, il s'agit du calcul de la dépendance de toutes les activités en fonction de l'activité accepter comme antécédent seulement. On se rend compte qu'accepter a des liens significatifs avec évaluer (intervalle 2), demander (intervalle 1 et 3), justifier (intervalle 1) et rejeter (intervalle 2). À partir de tels résultats, il est possible de déterminer plusieurs patrons significatifs. Par exemple, la Figure 17 représente plusieurs interactions significatives qui ne sont pas sans ressembler au dialogue de synchronisation cognitive discute précédemment. Ce

diagramme a été conçu grâce à la mise en commun de plusieurs grilles similaires à celle illustrée au Tableau 2.

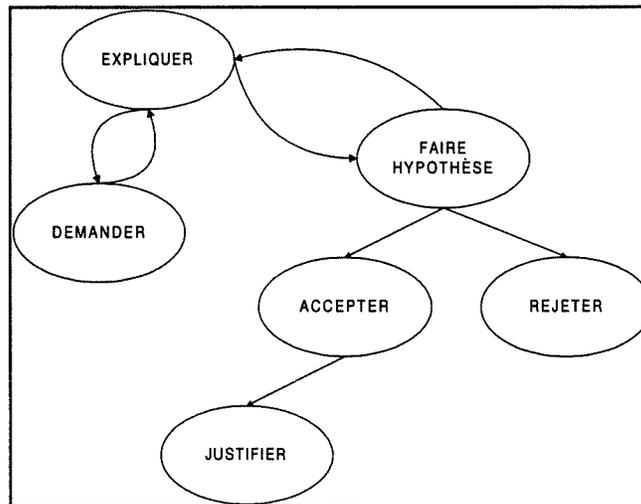


Figure 17 Patron identifié avec LSA

4.3.2 Modèles log-linéaires

Les modèles log-linéaires (MLL) permettent d'établir une combinaison linéaire des valeurs attendues d'un nombre de paramètres. L'ajustement d'un MLL correspond à valider certaines hypothèses faites sur les données. Il s'agit donc d'utiliser les techniques de validation d'hypothèse: estimation des fréquences attendues et comparaison avec les valeurs observées. Ce genre de modèle permet l'identification de patrons dans les données qui serviront à valider les hypothèses.

Un MLL se base sur cinq principes fondamentaux:

1. le principe de hiérarchie;
2. l'établissement d'hypothèses basées sur des concepts statistiques d'indépendance entre variables (ces hypothèses sont différentes de celles basées sur une quelconque théorie);
3. la distinction entre les modèles saturés et non-saturés;

4. le calcul du degré de liberté utilisé pour vérifier les MLL alternatifs; et
5. ajustement de modèles et vérification d'hypothèses.

Le principe de hiérarchie consiste à ordonner les hypothèses de façon à savoir laquelle sera vérifiée en premier et lesquelles seront vérifiées en sachant qu'une hypothèse antérieure est vérifiée.

La distinction entre des modèles saturés et non-saturés est très importante. Un modèle saturé ne contient aucun degré de liberté. Un modèle ne contient aucun degré de liberté lorsque le nombre de paramètres indépendants à estimer est égal au nombre de cellules de la table de contingence. Un tel modèle est parfaitement mais trivialement ajusté aux données puisque les valeurs attendues sont égales aux valeurs observées. L'objectif du MLL est de tenter d'éliminer les modèles saturés pour plutôt identifier des modèles non-saturés qui sont adéquatement ajustés aux données observées.

Le nombre total de degrés de liberté disponible pour estimer un modèle log-linéaire est égal au nombre de cellules contenues dans une table de contingence. Il faut soustraire à partir de ce total, le nombre de paramètres indépendants nécessaires pour vérifier un certain MLL qui est ajusté aux données. La différence équivaut au degré de liberté utilisé pour vérifier un modèle théorique.

La précision d'un modèle est vérifiée en calculant la différence entre les valeurs attendues obtenues à partir d'un modèle théorique quelconque et les données observées. La statistique chi-carré du ratio de la vraisemblance est utilisée pour calculer cette différence:

$$G^2 = 2 \sum \text{observe} * \ln \left(\frac{\text{observe}}{\text{attendu}} \right)$$

Les analyses log-linéaires sont supérieures à l'utilisation des méthodes statistiques traditionnelles parce qu'elles mesurent aussi les liens partiels. Le MLL permet de décider laquelle de plusieurs alternatives théoriques est le meilleur modèle pour représenter la structure d'une suite de données. Les MLL sont cependant moins utiles lorsqu'ils sont utilisés de façon exploratoire à

cause du grand nombre d'hypothèses à considérer. Sans la présence d'une base théorique solide, il est difficile de sélectionner un modèle dans un ensemble d'alternatives diverses.

4.3.3 Stratégies à utiliser

Les deux méthodes d'analyses statistiques présentées dans le présent chapitre ont leurs avantages et désavantages. Il est cependant possible d'utiliser chacune des méthodes là où elles peuvent le mieux aider dans l'identification de séquences significatives dans les données.

Le LSA permet de déceler des connexions séquentielles significatives dans les données. Cette méthode est donc extrêmement utile pour valider l'identification des dialogues et leurs interactions statistiquement. Cette approche permettra de valider les cinq dialogues trouvés et peut-être plus. L'identification et la validation de ces dialogues à l'aide de techniques probabilistiques facilitera l'analyse et la compréhension des réunions de révision technique.

Le MLL se concentre sur les liens de premier niveau entre plusieurs facteurs. Bien que les dialogues caractérisent les comportements retrouvés à l'intérieur d'une réunion, ils ne permettent pas de représenter tous les facteurs qui l'influencent. Ainsi, est-il important de définir plus en détail une réunion de révision technique :

1. Le rôle (responsabilité) des participants influence-t-il les comportements à l'intérieur de la réunion ?
2. Est-ce que toutes les réunions sont semblables, peut importe le moment de la réunion dans le projet de développement ?
3. Est-ce que le type de solution révisé (Pennington, 1987) est important ?
4. L'expérience des participants a-t-elle un effet sur la réunion ?
5. Le paradigme de programmation (structurel vs orienté objet) change-t-il la façon de réviser ?

Le MLL peut donc être utilisé pour identifier les relations entre les différents facteurs indiqués auparavant. Les résultats obtenus donneront ainsi une image sans équivoque des réunions de révision technique observés. Ces résultats seront validés statistiquement grâce au MLL.

CONCLUSION

Le développement de logiciels industriels est un travail d'équipe. Le processus logiciel facilite le développement en instaurant un cadre structuré de travail collaboratif composé d'objectifs très précis de qualité et de productivité. L'information pertinente au projet est conservée dans la documentation et partagée à l'aide de réunions de toutes sortes. L'un de ces types de réunion est la réunion de révision technique. Plusieurs descriptions de ce type de réunion existent et souvent se contredisent.

Le présent document a présenté une démarche qui devrait permettre de faire la lumière sur les activités cognitives et collaboratives retrouvées lors d'une réunion de révision technique. Cette démarche est exploratoire, c'est-à-dire que chacune des étapes de la recherche ne sont pas toutes connues pour l'instant. L'objectif est cependant très claire : la prescription d'une procédure de révision collaborative qui tient compte du résultat recherché et qui est ergonomique.

Les résultats présentés démontrent la facilité avec laquelle la structure à trois niveaux du schéma de codage permet d'extraire des données à partir du protocole d'une réunion. L'utilisation de méthodes statistiques pour identifier et modéliser les comportements validera encore plus les résultats. La modélisation des comportements collaboratifs selon plusieurs facteurs permettra de bien comprendre les caractéristiques qui influencent la productivité et la qualité du produit d'une réunion de révision technique.

RÉFÉRENCES

ALLISON, P.D., LIKER, J.K. (1982). Analyzing sequential categorical data on dyadic interaction: A comment on Gottman. Psychological Bulletin, 91, 393-403.

BELL, D., MORREY, I., PUGH, J. (1987). Software Engineering: A Programming Approach. Prentice-Hall International, Chapter 14.

BOLOIX, G., ROBILLARD, P.N. (1994). A Software System Evaluation Framework. Computer, 28, 17-26.

BRYCZYNSKI, B., MEESON, R.N., WHEELER, D.A. (1994). Software Inspection: Eliminating Software Defects. Proceedings of the sixth annual software technology conference.

CURTIS, B. (1980). Measurement and experimentation in software engineering. Proceedings of the IEEE, 68(9), 1144-1157.

D'ASTOUS, P. (1997). Les aspects de l'échange de l'information dans un processus de Génie Logiciel, EPM/RT-97/06 , École Polytechnique.

D'ASTOUS, P., DETIENNE, F., ROBILLARD, P.N., VISSER, W. (1997). Technical Review Meetings: a Framework for Cognitive Analysis. Empirical Studies of Programmers: Seventh Workshop, ACM Press.

DETIENNE, F., SOLLOWAY, E. (1990). An Empirically-derived Control Structure for the Process of Program Understanding. International Journal of Man-Machine Studies, 33, 323-342.

FAGAN, M.E. (1976). Inspection Software Design and Code. IBM Systems Journal, 15,3 , 182-

211.

FAGEN, R.M., YOUNG, D.Y. (1978). Temporal patterns of behavior: Durations, intervals, latencies, and sequences. Quantitative Ethology, New York: Wiley.

FALZON, P., DARSES-DE MONTMOLLIN, F., BÉGUIN, P. (1996). Collective Design Processes, Second International Conference on the Design Cooperative Systems, June 12-14.

HARTWIG, F., DEARING, B.E. (1979). Exploratory data analysis. Sage University Paper Series on Quantitative Applications in the Social Sciences, Newbury Park, CA: Sage.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. (1993) IEEE Standards Collection for Software Engineering Std 1028, IEEE Publishing.

JOHNSON, P.M. (1996). Reengineering Inspection: The Future or Formal Technical Review. Technical Report at University of Hawaii (ICS-TR-95-24).

NAUR, P. (1960). Revised Report on the Algorithmic Language ALGOL 60. Communications of the ACM, 3, 299-314.

OLSON, G.M., HERBSLEB, J.D., RUETER, H.H. (1994). Characterizing the sequential structure of interaction behaviors through statistical and grammatical techniques. Human-Computer Interaction, 9, 427-472.

PENNINGTON, N. (1987). Comprehension Strategies in programming. Empirical Studies of Programmers: Second Workshop, 100-113.

PETRE, M. (1995). Why looking isn't always seeing: Readership skills and Graphical

Programming. Communication of the ACM, 38(6), 33-44.

PORTER A., SIY, H., MOCKUS, A., VOTTA, L. Understanding the Sources of Variation in Software Inspections. Technical Report at University of Maryland (UMCP-CSD:CS-TR-3762).

SANDERSON, P.M., FISHER, C. (1994). Exploratory Sequential Analysis: Foundations. Human-Computer Interaction, janvier, 251-317.

TUKEY, J.W. (1977). Exploratory data analysis. Reading, MA : Addison-Wesley.

VAN SCOY, R.L. (1992). Software Development Risk: Opportunity, Not Problem. Software Engineering Institute, Carnegie Mellon University.

VISSER, W. (1990). More or less following a plan during design: opportunistic deviation in specification. International Journal of Man-Machine Studies, 33, 247-278.

ÉCOLE POLYTECHNIQUE DE MONTREAL

3 9334 00205079 5

École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec)
H3C 3A7

